Kauno technologijos universitetas

Informatikos fakultetas

BitB atakos atpažinimo metodo sukūrimas ir tyrimas

Baigiamasis magistro projektas

Andrius Sapitavičius

Projekto autorius

doc. Rasa Brūzgienė

Vadovė

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

BitB atakos atpažinimo metodo sukūrimas ir tyrimas

Baigiamasis magistro projektas

Informacijos ir informacinių technologijų sauga (6211BX008)

Andrius Sapitavičius

Projekto autorius

doc. Rasa Brūzgienė

Vadovė

dr. Gedeiminas Činčikas

Recenzentas

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Andrius Sapitavičius

BitB atakos atpažinimo metodo sukūrimas ir tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Andrius Sapitavičius

Patvirtinta elektroniniu būdu

Sapitavičius, Andrius. BitB atakos atpažinimo metodo sukūrimas ir tyrimas. Magistro studijų baigiamasis projektas / vadovė doc. Rasa Brūzgienė; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos inžinerija.

Reikšminiai žodžiai: objektų atpažinimas, optinis simbolių atpažinimas, sukčiavimo tinklapiai.

Kaunas, 2025. 63 p.

Santrauka

Darbe nagrinėjama „BitB“ (naršyklė naršyklėje) atakos koncepcija ir sukuriamas metodas, leidžiantis realiu laiku atpažinti tokio tipo sukčiavimo bandymus vartotojo naršyklėje. Ataka pagrįsta tuo, kad kenkėjiškame tinklapyje imituojamas naršyklės iššokantysis langas, siekiant apgauti vartotoją ir išgauti jo prisijungimo duomenis. Tyrimo tikslas – sukurti veiksmingą aptikimo metodą, kuris atpažintų tokio tipo atakas bei jas užkardytų.

Darbe pasiūlytas metodas remiasi prisijungimo formos identifikavimu iš tinklapio vaizdo, siekiant nustatyti, ar prisijungimo forma nėra suklastota. Realizuotas sistemos prototipas susideda iš naršyklės įskiepio bei atskiros analizės programos. Sistema analizuoja tinklapio vaizdą realiuoju laiku, leidžia aptikti atakos bandymus, užblokuoti tinklapio siunčiamus duomenis ir perspėti vartotoją.

Eksperimentinis tyrimas parodė, kad sukurta sistema pasiekia aukštą atakų atpažinimo tikslumą ir veikia pakankamai greitai, kad būtų praktiškai naudojama realiose situacijose. Darbo rezultatai rodo, kad tokio tipo metodas gali būti veiksminga priemonė apsaugai nuo „BitB“ atakų.

Sapitavičius, Andrius. Development and Analysis of a BitB Attack Recognition Method. Master's Final Degree Project / supervisor Assoc. prof. Rasa Brūzgienė; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Informatics Engineering.

Keywords: object detection, optical character recognition, phishing websites.

Kaunas, 2025. 63 pages.

Summary

This paper explores the concept of a BitB (Browser in the Browser) attack and presents a method for real-time detection of such attacks in the user's browser. BitB attacks work by simulating a browser pop-up on a malicious website, tricking users into submitting login credentials. The aim of this project is to develop an effective detection method that identifies and prevents this type of attack.

The method proposed in this paper is based on the identification of a login form from a web page view, in order to determine whether the login form is fake. The prototype system implemented consists of a browser plug-in and a separate analysis program. The system analyses the web page image in real time, allowing it to detect attack attempts, block the data sent by the web page, and warn the user.

The evaluation of the system has shown that it achieves a high level of accuracy in detecting attacks and is fast enough to be applicable in real-world environments. The results of the work show that this type of approach can be an effective means of protection against BitB attacks.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Santrumpų ir terminų sąrašas

Terminai:

OCR (angl. Optical Character Recognition) – technologija, leidžianti kompiuteriui nuskaityti ir atpažinti tekstą iš nuotraukų.

HTML (angl. *HyperText Markup Language*) – teksto žymėjimo kalba, naudojama kuriant interneto tinklapius.

URL (angl. *Uniform Resource Locator*) – interneto adreso forma, naudojama pasiekti tam tikrą išteklių.

DOM (angl. *Document Object Model*) – struktūrinis interneto puslapio turinio modelis, sukuriamas iš HTML ar XML kodo.

API (angl. *Application Programming Interface*) – sąsaja tarp programų, leidžianti joms perduoti duomenis tarpusavyje bei apibrėžia, kaip viena programa gali prašyti duomenų ar paslaugų iš kitos.

Įvadas

„BitB“ (angl. *Browser in the Browser*) atakos tipas yra pakankamai naujas, ir išnaudoja plačias šiuolaikinių naršyklių galimybes. Ši ataka veikia tokiu principu, jog lankomo kenkėjiško tinklapio lange yra atvaizduojamas netikras naršyklės langas, bandant imituoti naršyklės iššokantįjį langą, bei suvilioti vartotoją įvesti prisijungimo duomenis prie patikimos svetainės. Vartotojas pagal tam tikrus požymius gali patikrinti, ar vaizduojamas naršyklės langas yra „tikras“ ir patikimas, tačiau nesitikint, jog tokia ataka gali būti įvykdyta, vartotojas dažnu atveju gali tapti auka.

Šis atakos tipas bene dažniausiai naudojamas prieš „Steam“ žaidimų platformos vartotojus, kenkėjiškiems tinklapiams bandant apsimesti el. sporto organizacijomis bei taip siekiant išvilioti didelę vertę turinčių paskyrų prisijungimo duomenis [1]. „BitB“ ataka taip pat buvo panaudota siekiant išvilioti „Microsoft 365“, „Google Workspace“ ir įvairių socialinių tinklų platformų prisijungimo duomenis [2].

Kadangi ši ataka naudojama įvairiuose sukčiavimui skirtuose tinklapiuose, jai gali būti užkirstas kelias naudojantis esamais metodais, tokiais kaip juodieji sąrašai. Tačiau tinklapiai į juoduosius sąrašus įtraukiami tik po kurio laiko – vidutiniškai 8 valandų – per kurį spėjama išvilioti duomenis iš didelio skaičiaus vartotojų [3]. Siekiant aptikti sukčiavimui skirtus tinklapius prieš tai, kai jie įtraukiami į juoduosius sąrašus, atakų bandymus reikia aptikti realiu laiku vartotojo įrenginyje.

Todėl reikalingas metodas, kuriuo naudojantis būtų galima vartotojo įrenginyje realiu laiku atpažinti „BitB“ atakos bandymus, bei aptikus tokį bandymą, jį užblokuoti bei perspėti vartotoją.

Darbo tikslas ir uždaviniai:

Šio darbo tikslas: Sukurti metodą, leidžiantį aptikti ir atpažinti „BitB“ ataką bei eksperimentiškai ištestuoti metodo veikimo patikimumą.

Išsikelti uždaviniai darbo tikslui pasiekti:

1. atlikti „BitB“ atakos požymių ir esamų jos aptikimo bei prevencijos būdų analizę;
2. sukurti ir suprojektuoti metodą, gebantį aptikti „BitB“ ataką pagal pagrindinius jos požymius ir prevenciškai ją sustabdyti;
3. praktiškai realizuoti sukurtą metodą sistemos naudotojo naršyklėje;
4. eksperimentiškai ištirti sukurtą metodą, atliekant jo kiekybinę ir kokybinę analizę bei įvertinti sukurto metodo veikimo patikimumą.

1. „BitB“ atakos principo bei esamų aptikimo būdų apžvalga

Šiame skyriuje analizuojamas „BitB“ atakos principas, esami sukčiavimo atakų aptikimo būdai bei kaip juos būtų galima pritaikyti „BitB“ atakos atpažinimui.

1.1. Susiję atakų tipai

1.1.1. „MitM“ ataka

„MitM“ (angl. *man-in-the-middle*) – atakos tipas, kai piktavalis perima komunikacijos kanalo tarp vartotojo ir paslaugų teikėjo, valdymą. Turėdamas prieigą prie perduodamų duomenų, piktavalis gali juos stebėti bei modifikuoti. Komunikacijos kanalo valdymas gali būti perimtas įvairiuose taškuose. Dažniausiai tai atliekama vietiniame tinkle, pavyzdžiui, jungiantis per neapsaugotą belaidžio interneto prieigos tašką [4]. Sėkmingai įvykdžius ataką, vartotojui beveik neįmanoma jos aptikti, kadangi nėra jokių išorinių šios atakos požymių [5]. Tačiau šiais laikais didžioji dauguma tinklapių naudoja HTTPS protokolą, kuris užkerta galimybę piktavaliams stebėti perduodamus duomenis, taip pat neleidžia nepastebėtai modifikuoti perduodamus duomenis. Ši ataka gali būti sėkmingai įvykdoma išskirtiniais atvejais, kai vartotojas ignoruoja naršyklės perspėjimus apie negaliojantį sertifikatą [4].

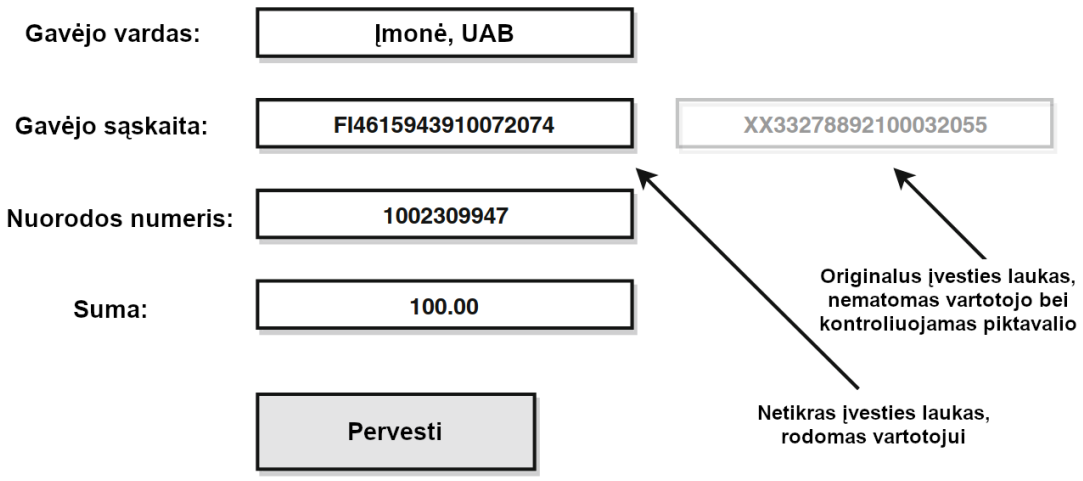
1.1.2. „MitB“ ataka

„MitB“ (angl. *man-in-the-browser*) atakos principas yra panašus į „MitM“ ataką. Piktavalis, vykdydamas „MitB“ ataką, panašiai kaip ir „MitM“ atakos metu, gali kontroliuoti duomenų srautą tarp vartotojo bei paslaugų teikėjo. Tačiau šios atakos skiriasi tuo, kurioje duomenų perdavimo kelio grandyje įsiterpia piktavalis. „MitB“ atakos metu, piktavalis kompromituoja vartotojo įrenginyje veikiančią interneto naršyklę. Tai piktavaliui leidžia be jokių apribojimų perimti bei modifikuoti siunčiamus duomenis [6].

Skirtingai nei „MitM“ atakos atveju, „HTTPS“ ar kitų saugių protokolų naudojimas apsisaugoti nepadeda, kadangi duomenų stebėjimas bei modifikavimas atliekamas prieš jų užšifravimą (arba po iššifravimo). Nuo šios atakos taip pat nepadeda apsisaugoti ir sudėtingos autentifikavimo procedūros, pavyzdžiui, dviejų faktorių autentifikacija [6].

Vartotojo interneto naršyklės kompromitavimas dažniausiai vykdomas į naršyklę įdiegiant įskiepį. Naršyklių įskiepiai turi galimybę naudojantis DOM (angl. *Document Object Model*) modifikuoti tinklapiuose atvaizduojamus duomenis. Kai kuriais atvejais, naršyklės įskiepis turi prieigą prie visų siunčiamų bei gaunamų „HTTP“ užklausų [7].

Kenkėjiškas įskiepis tokias plačias galimybes gali išnaudoti keliais būdais. Vienas iš būdų – perimti svarbius duomenis – prisijungimo, banko kortelių duomenis ir pan., bei juos persiųsti į piktavalio valdomą serverį. Kitas, sudėtingesnis „MitB“ atakos panaudojimo atvejis – vartotojui siunčiant užklausas, automatiškai modifikuoti užklausos turinį. Toks atakos pavyzdys pateiktas 1 pav.



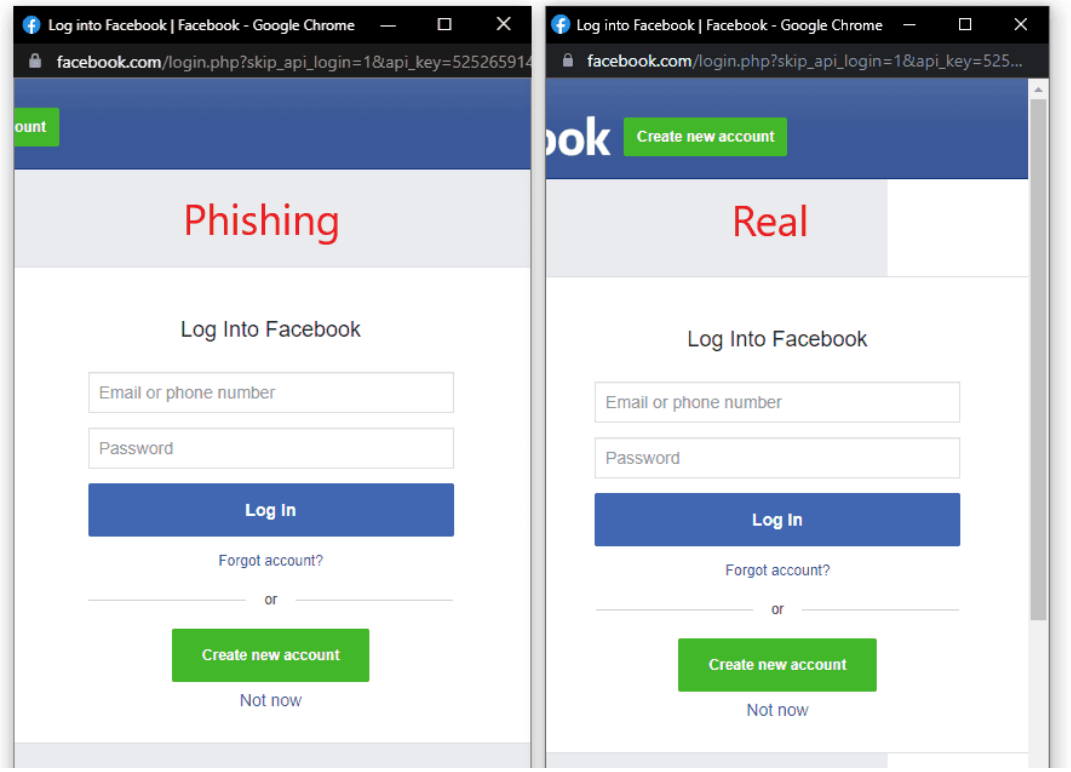
1 pav. „MitB“ atakos pavyzdys [7]

Toks būdas gali būti naudojamas, pavyzdžiui, atliekant pinigų pervedimo operacijas. Kenkėjiškas įskiepis gali modifikuoti sąskaitą, į kurią pervedami pinigai, taip pat pinigų sumą. Vartotojui įvedant duomenis, jam rodomas netikras įvesties laukas, ir patvirtinus pervedimą, bankui nusiunčiami piktavalio nurodyti duomenys [7].

1.2. „BitB“ atakos principas

Tai yra pakankamai naujas atakos tipas. 2022 metų kovo mėn. saugumo analitikas „mr.d0x“ šiam atakos tipui suteikė pavadinimą – „Browser in the Browser“ (naršyklė naršyklėje) [1].

Vienas iš pagrindinių požymių, pagal kurį vartotojas gali būti tikras, ar tinklapis, kuriame jis lankosi, yra saugus bei nekompromituotas – teisingas URL adresas bei galiojantis TLS (angl. *Transport Layer Security*) sertifikatas. Pagrindinė „BitB“ atakos idėja – tinklapyje atvaizduoti langą, kuris imituotų naršyklės iššokantįjį (angl. *pop-up*) langą. Tokiu būdu piktavalis gali netikrame naršyklės lange atvaizduoti netikrą URL adresą bei spynelės simbolį, simbolizuojantį galiojantį sertifikatą. Tokiu būdu vartotojas įtikinamas, jog atvaizduojamas puslapis yra tikrai tas, kurio URL adresas yra atvaizduojamas netikrame lange. 2 pav. pateiktas tikro bei sugeneruoto naršyklės iššokančiojo lango palyginimas.



2 pav. Tikro (dešinėje) bei sugeneruoto (kairėje) naršyklės langų palyginimas [8]

Iššokantieji langai yra dažnai naudojami vieno prisijungimo sistemos paslaugų tiekėjų. Bandant prisijungti prie tinklapio naudojant vieno prisijungimo sistemą, atidaromas naujas naršyklės langas. „BitB“ ataka daugeliu atveju ir yra naudojama siekiant išgauti prisijungimo duomenis iš populiarių vieno prisijungimo sprendimų tiekėjų, imituojant originalių prisijungimo formų išvaizdą [8].

1.3. „BitB“ atakos keliamos grėsmės

Piktavaliai, vykdydami „BitB“ ataką, gali įtikinti vartotoją, jog iššokančiame naršyklės lange vaizduojamas tinklapis yra tikras bei nesuklastotas. Tačiau norint įvykdyti šią ataką, vartotojas turi apsilankyti piktavalio kontroliuojamame tinklapyje. Šio tinklapio URL adreso suklastoti galimybės nėra, todėl vartotojas gali pakankamai nesunkiai atskirti, ar tinklapis yra suklastotas. Tačiau praktika rodo, jog tai netrukdo sėkmingai vykdyti atakas. Dažnu atveju, kenkėjiškas tinklapis imituoja populiarų tinklapį. Didelė dalis populiaraus tinklapio gali būti nuklonuota ir atlikti didelę dalį funkcijų, kurias atlieka originalus tinklapis. Kenkėjiško tinklapio URL adresas gali būti labai panašus į tinklapio, kurį bandoma imituoti. Dėl to vartotojas iš pirmo žvilgsnio gali neatskirti, jog lankosi kenkėjiškame tinklapyje. Kai kuriais atvejais kenkėjiška svetainė nebando apsimesti kita populiaria svetaine – ji gali būti unikali.

„BitB“ ataka gali būti sėkmingai įvykdyta net vartotojui apsaugojus paskyrą dviejų faktorių autentifikacija. Tai gali būti įgyvendinta tokiais žingsniais [9]:

1. vartotojas įrašo prisijungimo duomenis į netikrame naršyklės lange atvaizduotą prisijungimo formą;
2. prisijungimo duomenys į piktavalio valdomą serverį;
3. serveryje veikianti programa realiu laiku bando prijungti prie vartotojo paskyros;

4. esant įjungtam dviejų žingsnių autentifikavimo mechanizmui, vartotojas paprašomas įvesti kodą, skirtą antram autentifikacijos žingsniui;
5. vartotojui šį kodą įvedus, jį panaudoja piktavalis bei įgauna prieigą prie vartotojo paskyros.

Bene plačiausiai „BitB“ ataka pastaruoju metu yra naudojama prieš „Steam“ žaidimų platformos vartotojus. Vartotojai yra suviliojami apsilankyti kenkėjiškose svetainėse, kurios imituoja e. sporto organizacijos tinklapį. Šiame tinklapyje vartotojų prašoma prisijungti naudojant „Steam“ platformos paskyrą, norint naudotis įvairiomis funkcijomis. Pavyzdžiui, prisijungti prie komandos norint dalyvauti e. sporto turnyre, norint balsuoti už tam tikrą komandą, norint nusipirkti bilietus į e. sporto turnyrus ar kitus renginius. Reagavimo į kompiuterinio saugumo incidentus grupė CERT-GIB vien per 2022 metų liepos mėnesį identifikavo daugiau nei 150 apgaulingų bandymų apsimesti „Steam“ platforma [1].

1.4. „BitB“ atakos požymių bei aptikimo metodų analizė

Šiuo metu egzistuoja ne vienas metodas, kuriuo naudojantis galima identifikuoti kenkėjiškus, sukčiavimui bei duomenų išviliojimui skirtus tinklapius. „BitB“ ataka dažniausiai naudojama duomenų išviliojimui, todėl tinklapius, naudojančius šį atakos tipą galima identifikuoti esamais metodais. Tačiau egzistuojantys metodai turi įvairių trūkumų. Kadangi „BitB“ ataka yra specifinė ir turi išskirtinių požymių, tuo galima pasinaudoti siekiant tiksliau identifikuoti šią ataką, taip užkertant kelią galimiems duomenų išviliojimo atvejams.

1.4.1. „BitB“ atakos aptikimas rankiniu būdu

Vartotojas, norėdamas rankiniu būdu patikrinti, jog prieš jį nėra bandoma įvykdyti „BitB“ ataką, gali atkreipti dėmesį į keletą požymių bei atlikti keletą veiksmų.

Netikras iššokantysis langas gali turėti išvaizdos skirtumų, lyginant su tikru langu. Ypač reikia atkreipti dėmesį į patį naršyklės langą. Skirtingose interneto naršyklėse, taip pat skirtingose operacinėse sistemose naršyklės lango išvaizda skiriasi. Nors tinklalapiai turi tam tikrą informaciją apie vartotojo naršyklę bei naudojamą operacinę sistemą, realizuoti visus galimus lango išvaizdos variantus yra sudėtinga, todėl dažniausiai pasirenkama išvaizda, atitinkanti populiariausią interneto naršyklę bei operacinę sistemą – pavyzdžiui, „Google Chrome“ bei „Microsoft Windows 10“. Pastebėjus, jog iššokančiojo lango išvaizda skiriasi nuo įprastinės išvaizdos, galima įtarti, jog bandoma atlikti „BitB“ ataką [9].

Netikras iššokantysis langas taip pat gali turėti nepilną funkcionalumą. Pavyzdžiui, paspaudus ant spynos logotipo, nebus atidaromas langas su informacija apie tinklapio sertifikatą. Netikras langas taip pat gali neturėti dydžio keitimo funkcionalumo [9].

Kadangi netikras naršyklės langas yra sugeneruotas lankomame tinklapyje, jo negalima perkelti už pagrindinio tinklapio ribų. Dėl to, norint įsitikinti, jog langas yra tikras, jį galima pabandyti nutempti už pagrindinio naršyklės lango ribų [9].

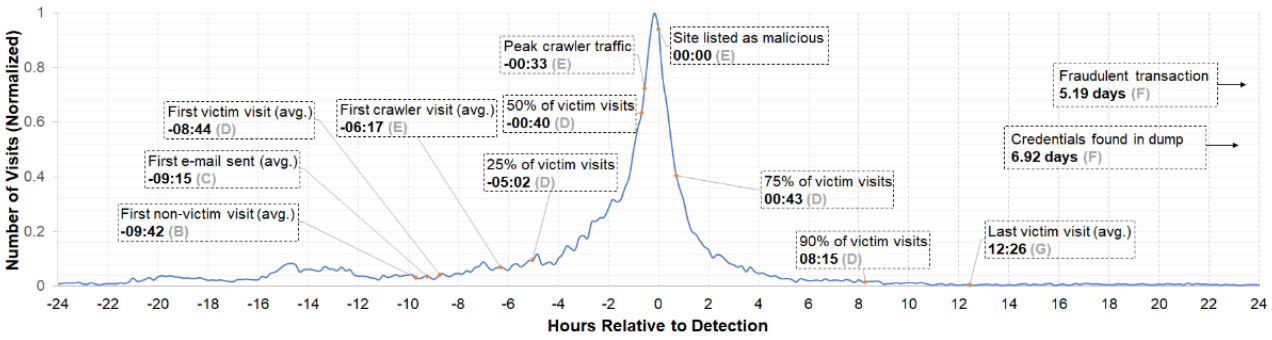
1.4.2. Sukčiavimo tinklapių aptikimas naudojant URL juoduosius sąrašus

Populiariausios interneto naršyklės bei paieškos varikliai, siekiant apsaugoti vartotojus nuo žalingos informacijos, naudoja tinklapių filtravimą pagal URL juoduosius sąrašus. Egzistuoja du pagrindiniai įrankiai – „Google Safe Browsing“ (GSB) bei „Phishtank“.

„GSB“ įrankis tinklapius į juodąjį sąrašą įtraukia automatiniu būdu, botams aptikus sukčiavimo atvejus, bandymus platinti kenkėjišką programinę įrangą. Taip pat yra galimybė ir pranešti apie kenkėjišką tinklapį rankiniu būdu [10].

Kitas įrankis, „Phishtank“ veikia visiškai kitu principu. Kenkėjiškų tinklapių aptikimas paremtas vartotojų pranešimais. Vartotojai gali pranešti apie naujus kenkėjiškus tinklapius, taip pat gali balsuoti už tinklapius, kurie jau įtraukti į sąrašą. Esant pakankamam kiekiui vartotojų, pranešusių apie kenkėjišką tinklapį, jis įtraukiamas į juodąjį sąrašą [11].

Filtravimas pagal URL adresus yra pagrindinis būdas, kaip interneto naršyklės apsaugo vartotojus nuo galimų sukčiavimo atakų, tame tarpe ir „BitB“ tipo atakų. Tačiau toks būdas turi tam tikrų trūkumų. Pagrindinis trūkumas – užtrunka laiko, kol tinklapis įtraukiamas į juodąjį sąrašą. 3 pav. pavaizduoti tyrimo rezultatai, per kiek laiko nuo sukčiavimo tinklapio sukūrimo bei pirmosios aukos praeina, iki kol tinklapis pažymimas kaip kenkėjiškas bei prarandamas vartotojų srautas. Vidutiniškai tai gali užtrukti apie 21 valandą [3]. Nors toks laiko tarpas yra pakankamai trumpas, jo užtenka jog būtų paveiktas nemažas žmonių kiekis. Naujų domenų registravimas bei TLS sertifikato gavimas yra nesudėtingas procesas, dėl to, piktavališkam tinklapiui atsidūrus juodajame sąraše, piktavaliai greitai perkelia tinklapį į kitą serverį ir domeną.



3 pav. Kenkėjiškų tinklapių gyvavimo laikas [3]

1.4.3. Sukčiavimo tinklapių atpažinimas pagal iš anksto numatytus kriterijus

Norint atpažinti sukčiavimo tinklapius, kol jie dar neįtraukti į juoduosius sąrašus, reikia analizuoti tinklapių turinį bei kitus parametrus kliento įrenginyje, arba nutolusiame serveryje, pagal kliento užklausą. Vienas iš būdų yra analizuoti tinklapių URL adresus bei išeities kodą pagal iš anksto numatytus kriterijus.

Straipsnyje „A lightweight and proactive rule-based incremental construction approach to detect phishing scam“ [12] siūlomi tokie kriterijai:

URL adreso tikrinimas:

- IP adreso naudojimas;
- šešioliktainių reikšmių naudojimas;
- neįprastai ilgas URL adresas;
- „@“ simbolio naudojimas;
- priešdėlio (angl. *prefix*) ar priesagos (angl. *suffix*) naudojimas;

- URL adreso sutrumpinimo paslaugų naudojimas;
- didelis subdomenų skaičius bei taškų naudojimas adrese.

Tinklapio turinio analizavimas:

- išjungtas dešinio pelės klavišo naudojimas;
- iššokančiųjų langų naudojimas;
- el. pašto funkcijų naudojimas;
- dvigubų pasviruųjų brūkšnelių naudojimas;
- „iframe“ HTML žymų naudojimas.

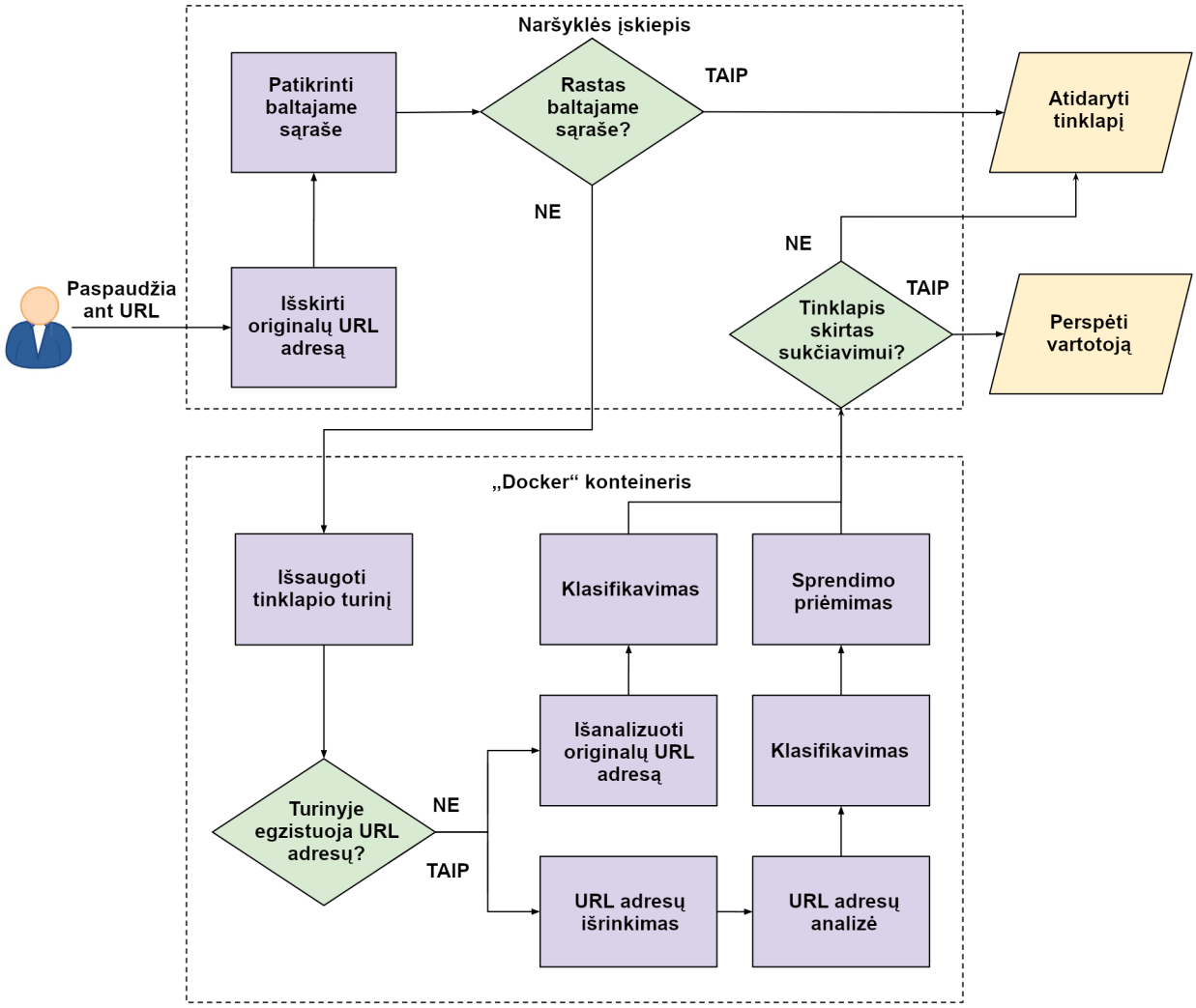
Kitų kriterijų analizavimas:

- TLS sertifikato patikimumas bei amžius;
- tinklapio populiarumo įvertinimas naudojant trečiųjų šalių servisus „Google Page rank“ bei „Alexa Ranking“.

Tinklapių atpažinimui pagal kriterijus naudojama keletas skirtingų metodų. Vienas paprasčiausių metodų – kiekvienam kriterijui suteikti tam tikrą svorį, kurį lemia šio kriterijaus vidutinis pasikartojimo dažnumas sukčiavimo tinklapiuose. Tinklapiui atitikus tam tikrus kriterijus, sudedami jų svoriai. Galutinei reikšmei viršijus iš anksto nustatytą ribą, tinklapis laikomas kaip skirtas sukčiavimui [12].

Kiti aptikimo metodai remiasi įvairiais klasifikavimo algoritmais. Straipsnyje [13] aprašytas metodas, kuriame tinklapių klasifikavimui panaudotas atsitiktinių medžių metodas (angl. *Random forest*). Iš pradžių surinkta tinklapių duomenų bazė, kurią sudaro apie pusę sukčiavimo tinklapių bei ne sukčiavimui skirtų tinklapių. Naudojant pasirinktus kriterijus, kiekvienam tinklapiui apskaičiuotos reikšmės, reiškiančios atitikimą konkrečiam kriterijui. Naudojantis šiais duomenimis, išmokytas mašininio mokymosi modelis.

2024 metais pasiūlytas hibridinis metodas „PhishingRTDS“, kuris leidžia efektyviau aptikti „BitB“ bei sutrumpintų URL adresų atakas nei kiti panašūs metodai. [14]. Šio metodo sprendimo modelis pateiktas 4 pav. Pirmame žingsnyje, lankomo tinklapio adresas yra patikrinamas baltajame sąraše. Jei adresas sąraše randamas, tinklapis atidaromas, kitu atveju vykdoma tolimesnė analizė. Tolimesniame žingsnyje atliekama URL adresų analizė. Skirtingai nei anksčiau minėtuose darbuose, kuriuose analizuojamas tik lankomo tinklapio URL adresas, šiame darbe analizuojami ir URL adresai, esantys pačiame tinklapio turinyje. Tai leidžia efektyviau atpažinti „BitB“ atakas, kurių vienas iš pagrindinių požymių yra URL adresas netikrame naršyklės lange. URL adresų klasifikavimas vykdomas naudojant mašininio mokymosi metodą, kuris buvo išmokytas naudojant tinklapių duomenų bazę, susidedančią iš 60 tūkst. sukčiavimui skirtų tinklapių bei 25 tūkst. atsitiktinių tinklapių. Darbo autoriai teigia, jog buvo pasiektas 0,92 tikslumo įvertis [14].



4 pav. „PhishingRTDS“ sprendimo modelis [14]

Tinklapių tikrinimas pagal minėtus kriterijus nereikalauja didelių skaičiavimo resursų, taip pat didelę jų dalį galima patikrinti vartotojo įrenginyje, neatliekant užklausų į išorinius serverius (išskyrus atvejus, kai naudojami hibridiniai metodai). Tačiau tokie aptikimo metodai turi ir trūkumų. Piktavaliai vis dažniau ima naudoti įvairius metodus, apsunkinančius tinklapių išeities kodo analizę. Pavyzdžiui, naudojami įrankiai „JavaScript“ kodo užtemdymui. Šiais įrankiais apdorotas kodas tampa sunkiai analizuojamas tiek rankiniu būdu, tiek automatiniais įrankiais. Taip pat naudojami įrankiai HTML bei CSS modifikavimui, kurie neatpažįstamai pakeičia struktūrą nepakeičiant funkcionalumo [15].

Bandant pritaikyti šį metodą „BitB“ atakos atpažinimui, kyla tos pačios problemos. Šiuo metu daugelis vykdomų „BitB“ atakų turi nesudėtingus indikatorius, pagal kuriuos jas būtų galima atpažinti. Pavyzdžiui, vienas pagrindinių indikatorių – „iframe“ HTML žymos naudojimas. Šis HTML funkcionalumas leidžia prisijungimo formą, saugomą atskirame HTML faile, nesunkiai atvaizduoti pagrindiniame tinklapio lange. Anksčiau nagrinėtas „PhishingRTDS“ metodas remiasi tuo, jog netikrame naršyklės lange turinys atvaizduojamas naudojant „iframe“ žymą [14]. Tačiau jau šiuo metu egzistuoja būdai, leidžiantys sugeneruoti netikrą naršyklės langą be „iframe“ žymių [16]. Kadangi netikro naršyklės lango turinys gali būti būti atsisiųstas dinamiškai, tik vartotojui atlikus tam tikrus veiksmus (paspaudus specifinį mygtuką), vien tik analizuojant atsisiųstą tinklapio

turinį pradinėje būsenoje gali būti neefektyvu. Kaip indikatorius taip pat galėtų būti „JavaScript“ kodas, leidžiantis tokius veiksmus kaip lango dydžio, pozicijos keitimas. Tačiau, kaip minėta, egzistuoja būdai, kurie leidžia tokius požymius padaryti sunkiai atpažįstamais.

1.4.4. Sukčiavimo tinklapių atpažinimas naudojant palyginimą

Kadangi sukčiavimo tinklapiai daugeliu atvejų imituoja populiarius tinklapius, jie yra vizualiai identiški arba labai panašūs. Aptikus, jog tinklapis yra panašus į kitą, populiarų, tinklapį, galima nesunkiai nuspėti, jog puslapis siekia apgauti vartotojus.

Puslapių panašumui palyginti naudojama keletas metodų. Daugelis jų nuolat renka informaciją apie vartotojo lankomus tinklapius ir išsaugo duomenis, kurie naudojami palyginti su vėliau lankomais tinklapiais [17].

Vienas iš tinklapių palyginimo būdų yra naudojant tinklapio objektų modelį. Dokumento objektų modelis (angl. *Document Object Model*), sutrumpintai DOM – hierarchinis, medžio struktūros dokumento modelis, reprezentuojantis dokumento struktūrą. DOM naudojamas HTML bei XML dokumentams. Lyginant skirtingų tinklapių DOM, galima pakankamai nesunkiai identifikuoti panašumus. Lyginimo algoritmas apskaičiuoja dokumentų objektų panašumus, ir esant pakankamai dideliam panašumo įvertinimui, tinklapiai laikomi panašiais [17].

Kitas būdas yra tinklapių lyginimas pagal išvaizdos kriterijus. Šis palyginimo būdas lygina įvairius parametrus, darančius tiesioginę įtaką tinklapio išvaizdai. Pirmiausiai, tinklapis padalijamas į atskirus blokus, tuomet surenkama informacija apie kiekvieną iš blokų. Tuomet šių blokų savybės palyginamos su kitais tinklapiais. Įvertinami tokie kriterijai [18]:

- tinklapyje pateiktas tekstas;
- fono spalva;
- kraštinių stilius, storis, spalva;
- šrifto stilius, dydis, spalva, teksto lygiavimas, stilius;
- paveikslų savybės (dydis, atributai, pozicija).

Kitas tinklapių vizualus tikrinimas remiasi ne informacija, surinkta iš tinklapio išeities kodo, tačiau iš tinklapio vaizdo kopijos. Vartotojui apsilankius tinklapyje, užfiksuojamas tinklapio vaizdas, ir naudojantis įvairiais algoritmais, ieškoma atitikmenų.

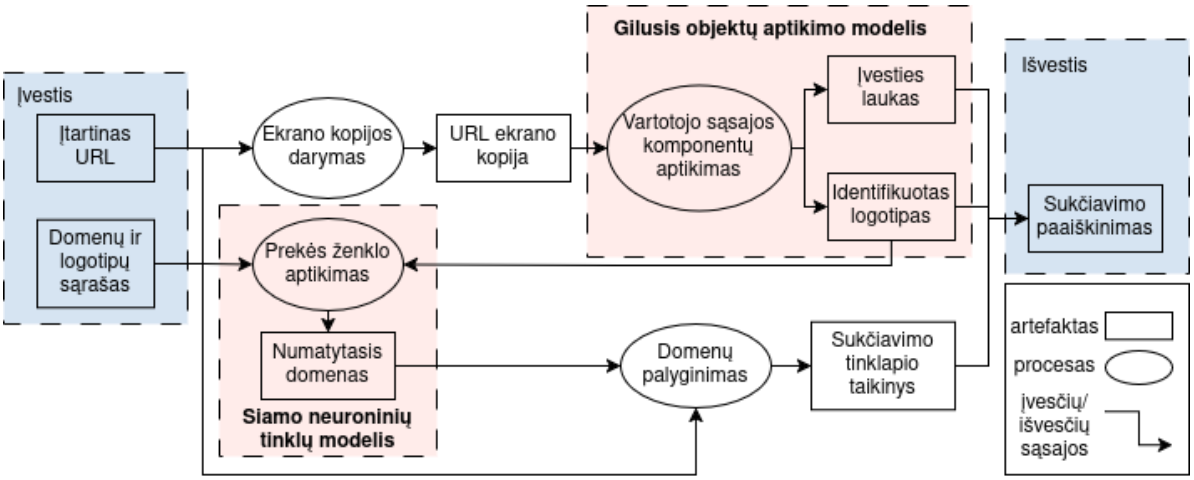
Straipsnyje [19] aprašytas metodas, kai iš tinklapio ekrano vaizdo, naudojant kelis skirtingus metodus, išgaunami deskriptoriai, pagal kuriuos ieškoma atitikmenų. Palyginimui naudoti penki skirtingi metodai. Panaudoti SDC (angl. *Scalable Color*) ir CLD (angl. *Color Layout*) deskriptoriai iš MPEG-7 standarto. SDC metodas sugeneruoja spalvų histogramą, apibūdinančią bendrą spalvų pasiskirstymą vaizde. CLD deskriptorius apibūdina erdvinį spalvų pasiskirstymą vaizde, padalinant vaizdą į 64 blokus ir apskaičiuojant vidutinę kiekvieno bloko spalvą. Kiti trys indikatoriai yra kompleksiški, įvertinantys keletą kriterijų – CEDD (angl. *Color Edge Directivity Descriptor*), FCTH (angl. *Fuzzy Color and Texture Histogram*) bei JCD (angl. *Joint Composite Descriptor*). Pasiūlytas metodas naudoja kliento – serverio architektūrą. Kliento įrenginyje veikianti programa išgauna tinklapio vaizdą, apskaičiuoja vaizdo deskriptorius, tuomet šiuos duomenis siunčia į serverį. Serverio duomenų bazėje saugomi tinklapių URL adresai bei juos atitinkantys deskriptoriai. Serveryje radus tinklapio atitikmenį, palyginami URL adresai – esant vizualiam panašumui, tačiau skiriantis URL adresams, galima daryti išvadą, jog tinklapis yra kito tinklapio kopija.

Toks aptikimo metodas, naudojant tinklapių vaizdų palyginimą, turi nemažų trūkumų. Šiuolaikinių tinklapių išvaizda yra pakankamai dinamiška. Neretai dalį ploto užima reklamos, kurių turinys nuolat kinta. Taip pat dažnai naudojamos nuolat besikeičiančios nuotraukos ar vaizdo įrašai [19]. Naudojantis vaizdų palyginimo metodu, tokių tinklapių lyginimas būtų labai netikslus. Kita vertus, svarbiausia sukčiavimo tinklapių dalis yra formos, į kurias prašoma įvesti prisijungimo ar kitus duomenis. Šių formų vaizdas yra labiau statinis, todėl toks vaizdų palyginimo metodas galėtų būti efektyvus. Dar vienas trūkumas yra toks, jog duomenų bazėje turi būti saugoma informacija apie didelį kiekį tinklapių. Taip pat ši informacija turi būti nuolat atnaujinama.

Kitas, pakankamai plačiai naudojamas metodas palyginti tinklapius, yra logotipų analizavimas. Beveik visais atvejais populiarūs tinklapiai, kuriuos bando imituoti piktavaliai, turi savo unikalius logotipus, reprezentuojančius konkrečią įmonę, prekės ženklą. Tarp visų elementų, logotipas yra vienas pagrindinių elementų, reprezentuojančių tinklapį [20]. Logotipai yra neatsiejama prekės ženklų dalis, ir daro didelę įtaką vartotojams, kaip jie vertina tinklapio patikimumą [21].

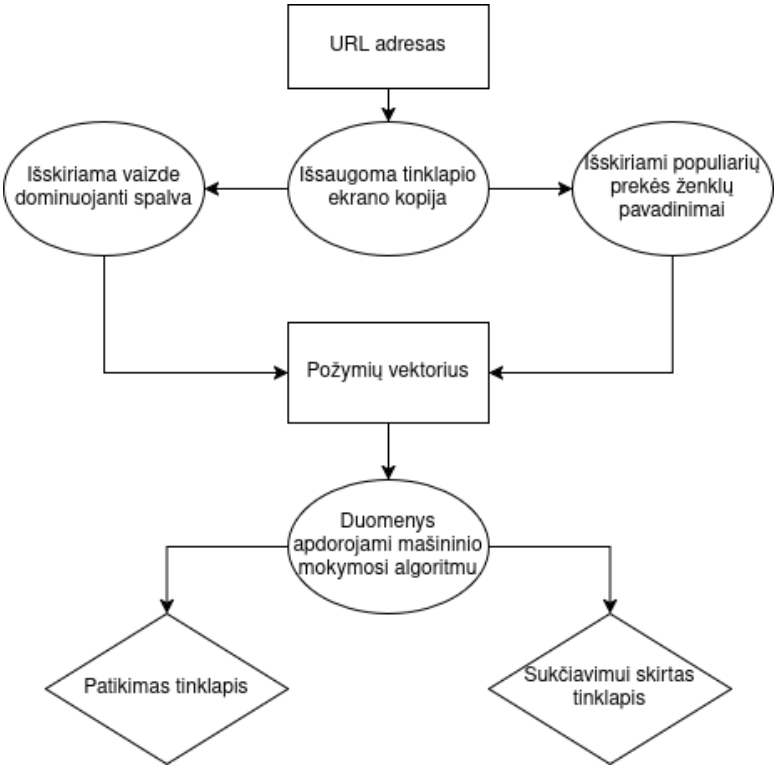
2020 metais pasiūlytas sprendimas „LogoSENSE“. Pagrindinė darbo idėja – naudojant orientuotų gradientų histogramų (angl. *Histogram of oriented gradients*) metodą, tinklapio vaizde surasti bei identifikuoti prekės ženklo logotipą. Darbo metu algoritmas buvo mokomas atpažinti 15 skirtingų prekės ženklų logotipus. Algoritmo mokymui buvo panaudotas pačių darbo autorių surinktas duomenų rinkinys, kuriame yra 1530 vaizdai skirti mokymui bei 1979 skirti tikslumui įvertinti. Mokymui skirti tinklapių vaizdai yra anotuoti – pateikiamos kiekviename vaizde esančių logotipų koordinatės bei logotipo pavadinimas. Išmokytas algoritmas pasiekė 93,5 % atpažinimo tikslumą [21].

2021 metais pasiūlytas metodas „Phishpedia“ [22]. Jame taikomas hibridinis modelis tinklapių prekės ženklui identifikuoti (žr. 5 pav.). Darbe panaudoti 2 skirtingi giliojo mokymosi modeliai. Pirmasis iš jų yra skirtas tinklapio vaizde identifikuoti įvesties formas bei logotipus. Pirmasis modelis nenustato, kokį prekės ženklą logotipas reprezentuoja, tačiau nustato tik jo poziciją vaizde. Antras modelis, naudojantis Siamo neuroninį tinklą, ieško logotipo atitikmens duomenų bazėje ir identifikuoja konkretų logotipą bei prekės ženklą, su kuriuo jis siejasi. Darbo metu taip pat buvo surinktas duomenų rinkinys, talpinantis informaciją apie 30 000 tinklapių – HTML turinį, ekrano vaizdus, kuriuose yra anotuoti logotipai ir nurodytas jų prekės ženklas. Metodo tikslumas buvo įvertintas bei palygintas su kitų metodų, tokių kaip „Phishzoo“ bei jau minėto „LogoSENSE“ tikslumu. Eksperimento rezultatai parodė, jog „Phishpedia“ pasiekė bendrą 98,2 % tikslumą, taip pat vidutinis analizavimo laikas buvo 0,19 sekundės. Tuo tarpu „LogoSENSE“ metodas pasiekė 26,9 % tikslumą ir vidutiniškai analizė truko 27,2 sekundės, testuojant tokiomis pačiomis sąlygomis. Visi kiti lyginti metodai taip pat nepasiekė „Phishpedia“ gauto tikslumo bei greitaveikos [22].



5 pav. „Phishpedia“ sprendimo modelis [22]

2023 metais pasiūlytas sprendimas „Phish-Sight“ (žr. 6 pav.). Šiame darbe naudojami du pagrindiniai požymiai tinklapio atitikmenų paieškai. Pirmas požymis – ekrano kopijoje dominuojančios spalvos. Darbe teigiama, jog piktavaliai vis plačiau naudoja įvairius kodo užtemdymo metodus, taip pat vengia kurti identiškas tinklapių kopijas, siekdami apsunkinti jų aptikimą. Nepaisant to, pagrindinės tinklapio dizaine naudojamos spalvos turi išlikti panašios, jog vartotojams tinklapis asocijuotųsi su tikruoju tinklapiu, kurį bandoma kopijuoti.



6 pav. „Phish-Sight“ architektūra [23]

Antras požymis, pagal kurį ieškoma tinklapio atitikmuo, yra populiarių prekių ženklų pavadinimų paieška tinklapio turinyje. Tekstui iš tinklapio išgauti yra naudojama OCR (angl. *Optical character*

recognition) technologija. OCR naudojamas dėl dviejų priežasčių – apeiti piktavalių naudojamus užtemdymo metodus, dėl kurių vartotojams pateikiamas tekstas nėra tiesiogiai aptinkamas HTML kodo išvestyje. Kita priežastis – dalis prekės ženklų naudoja ženklo pavadinimą pačiame logotipe. Naudojant OCR, galima identifikuoti dalį prekės ženklų iš jų logotipų. Išskyrus šiuos požymius, „Phish-Sight“ buvo išmokytas mašininio mokymosi modelis. Palyginti keli klasifikavimo algoritmai, geriausias rezultatas pasiektas naudojant atsitiktinių medžių algoritmą – pasiektas 99,13 % tikslumas [23].

1.5. „BitB“ atakos požymių apibendrinimas

Galima išskirti šiuos pagrindinius požymius, kurie sutinkami šiuo metu naudojamose „BitB“ atakose:

- vaizduojamas netikras naršyklės langas, kuriame yra matomas originalus suklastoto tinklapio adresas;
- dažniausiai imituojamos populiarių vieno prisijungimo paslaugas teikiančių tiekėjų prisijungimo formos;
- prisijungimo formų išvaizda vizualiai yra identiška arba nežymiai skiriasi nuo originalios.

1.6. Analizės apibendrinimas

1. „BitB“ atakos tipas yra pakankamai naujas, tačiau šiuo metu yra pakankamai plačiai ir sėkmingai naudojamas. Pagrindinis kriterijus, pagal kurį galima įsitikinti, jog lankomas tinklapis yra saugus ir tikras yra URL adresas bei galiojantis TLS sertifikatas. „BitB“ ataka šį požymį sėkmingai apeina, sugeneruojant netikrą naršyklės langą su iš požiūrio patikimu URL adresu bei galiojančiu sertifikatu.
2. Pagrindinis „BitB“ atakos taikinys – vartotojų prisijungimo duomenys prie įvairių paskyrų, teikiančių vieno prisijungimo paslaugas.
3. „BitB“ ataka tam tikrais atvejais leidžia apeiti dviejų žingsnių autentifikacijos mechanizmą.
4. Šiuo metu plačiausiai naudojamas būdas apsisaugojimui nuo sukčiavimo atakų, tame tarpe ir „BitB“ atakos – URL juodieji sąrašai. Tačiau jų pagrindinis trūkumas yra tai, jog praeina pakankamai didelis laiko tarpas nuo sukčiavimo puslapio sukūrimo iki jo įtraukimo į juoduosius sąrašus.
5. Egzistuoja ne vienas metodas nulinės dienos sukčiavimo atakų atpažinimui vartotojo įrenginyje. Vieni metodai remiasi tinklapio URL adreso analize, išeities kodo pagal konkrečius kriterijus analize. „BitB“ atakos tipas taip pat turi keletą išskirtinių požymių, kuriuos įmanoma aptikti analizuojant tinklapio išeities kodą. Tačiau piktavaliai naudoja įvairius metodus, skirtus apsunkinti išeities kodo analizę, pavyzdžiui, naudojant kodo užtemdymą.
6. Kiti atpažinimo metodai remiasi vizualiniu tinklapių palyginimu. Šių metodų tikslas – rasti sukčiavimo tinklapio atitikmenį tarp kitų populiarių tinklapių. Šiuolaikiniai tinklapiai dažnai naudoja įvairius dinaminius elementus, dėl kurių tinklapio išvaizda nuolat keičiasi, kas sumažina šio metodo efektyvumą. „BitB“ atakos atveju, pagrindinis vizualinis elementas, kurį reikia aptikti, yra netikras naršyklės langas. Todėl toks metodas gali būti naudojamas „BitB“ atakų atpažinimui.

2. „BitB“ atakos atpažinimo metodo projektas

Šiame skyriuje pateikiamas „BitB“ atakos atpažinimo metodo modelis, iškeliami reikalavimai metodui bei realizuojamam prototipui bei analizuojama, kokie metodai bei įrankiai geriausiai tiktų įgyvendinti siūlomą modelį.

2.1. Siūlomo sprendimo modelis

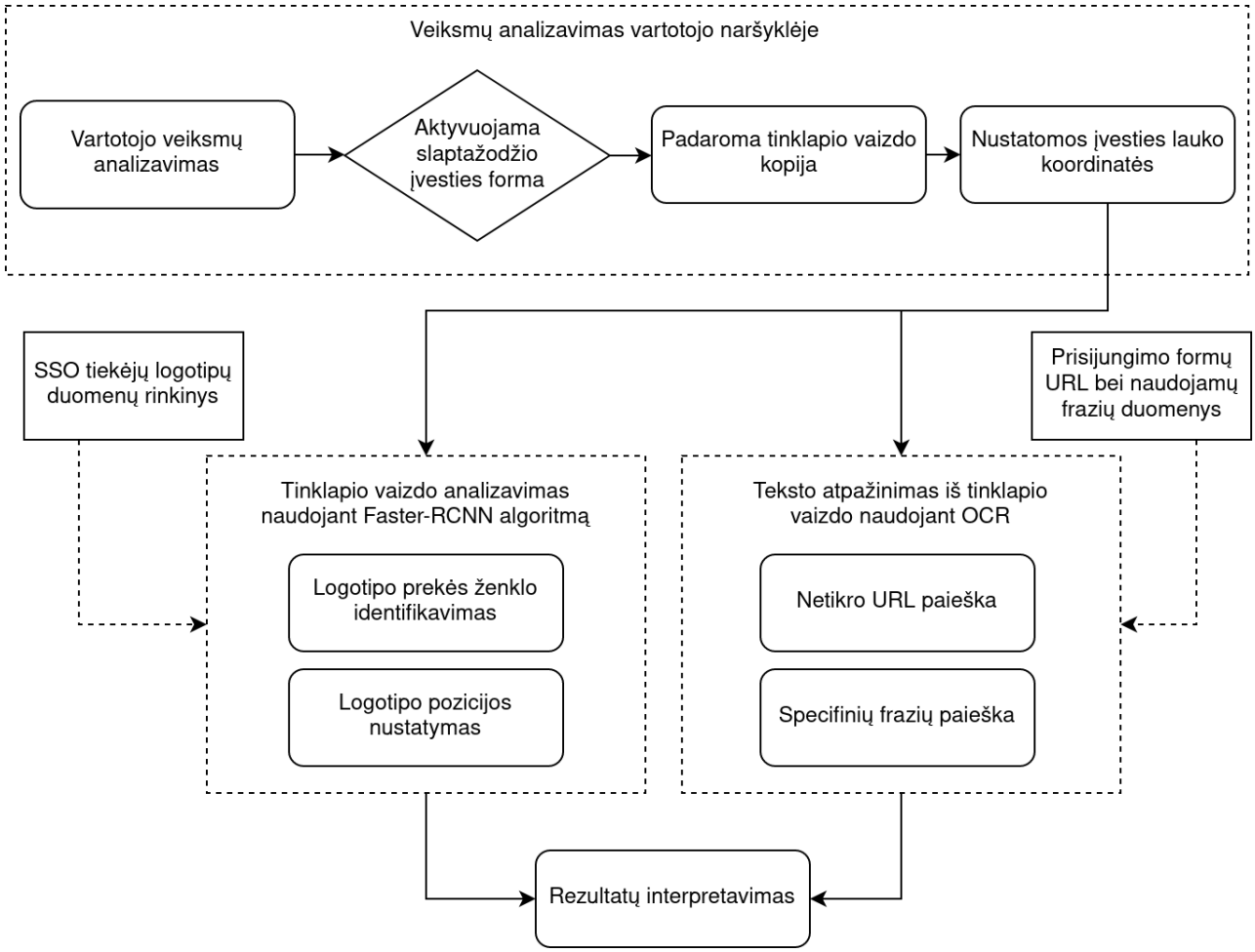
Analizės metu buvo nuspręsta, jog atakos aptikimas naudojant vizualų palyginimą geriausiai tiktų siekiant aptikti „BitB“ ataką. Siūlomas sprendimo modelis remiasi tuo, jog „BitB“ atakos dažniausiai naudojamos vieno prisijungimo paslaugų teikėjų prisijungimo formoms imituoti. Siūlomo sprendimo veikimo principas yra pagal įvairius požymius atpažinti konkrečią prisijungimo formą, kurią bandoma imituoti kenkėjiškame tinklapyje. Jeigu prisijungimo forma yra identifikuojama tinklapyje, kurio adresas nesutampa su originalios prisijungimo formos adresu, laikoma, jog vykdoma „BitB“ ataka.

Siūlomo sprendimo vizija pateikta 7 paveiksle. Analizavimo procesas prasideda vartotojo naršyklėje. Naršyklėje veikianti programa analizuoja, ar tinklapyje egzistuoja prisijungimo forma, skirta įvesti jautriems duomenims (prisijungimo slaptažodžiui). Aptikus, jog tokia forma egzistuoja, sekami vartotojo veiksmai. Vartotojui pažymėjus jautrių duomenų įvesties lauką, atliekami tolimesni veiksmai. Pirmiausiai padaroma tinklapio vaizdo kopija. Šiame vaizde išsaugoma tik ta tinklapio sritis, kuri tuo metu yra matoma vartotojui. Kitas veiksmas – nustatoma pažymėto įvesties lauko pozicija. Šie surinkti duomenys – tinklapio vaizdas bei įvesties lauko pozicija vaizde – perduodami kompiuteryje veikiančiai programai išanalizuoti.

Pirmas veiksmas, atliekamas su gautu tinklapio vaizdu, yra logotipų atpažinimas. Šiai užduočiai atlikti reikalingas objektų atpažinimo algoritmas, gebantis paveiksle identifikuoti konkretų logotipą. Modelis bus mokomas atpažinti populiariausių vieno prisijungimo sistemų tiekėjų logotipus. Kadangi tai yra objektų atpažinimo algoritmas, jis gebės atpažinti visus vaizde esančius logotipus bei jų pozicijas. Tai vėliau padės priimti galutinį sprendimą, ar logotipas yra prisijungimo formos dalis, ar ne.

Lygiagrečiai logotipų atpažinimui, bus vykdomas teksto atpažinimas iš tinklapio vaizdo naudojant OCR (angl. *Optical character recognition*) algoritmą. Tekstui atpažinti pasirinktas „Tesseract“ įrankis. Gautas tekstas yra naudojamas kaip papildomas požymis, siekiant identifikuoti prisijungimo formą.

Teksto atpažinimas iš vaizdo yra reikalingas dėl to, jog piktavaliai plačiai naudoja įvairius būdus, leidžiančius „paslėpti“ tekstą tinklapio HTML kodo išvestyje, ir padaryti jį neatpažįstamą įprastais metodais.



7 pav. Sprendimo vizija

Galiausiai, remiantis logotipų atpažinimo modelio rezultatais, OCR algoritmu atpažintu tekstu bei įvesties lauko pozicijos duomenimis, priimamas galutinis įvertinimas, ar ekrane yra rodoma netikra prisijungimo forma.

2.2. Keliami reikalavimai

Kuriamam „BitB“ atakos atpažinimo ir prevencijos metodui bei realizuotam sistemos prototipui keliami šie funkciniai reikalavimai:

- sistema turi gebėti atpažinti bent 5 skirtingų tipų prisijungimo formas;
- sistema prisijungimo formos identifikavimui turi panaudoti bent 3 skirtingus požymius;
- sistema turi prevenciškai užblokuoti visas tinklapio siunčiamas užklausas, jei atpažįstama „BitB“ ataka.

Keliami nefunkciniai reikalavimai:

- sistema negali naudoti daugiau nei 1,5 GB operatyvinės atminties bei daugiau nei 512 MB grafikos procesoriaus atminties;
- sistemos užklausos apdorojimo trukmė įprastomis sąlygomis negali viršyti 3 sekundžių;
- sistema turi veikti vartotojo įrenginyje, be poreikio komunikuoti su nutolusiais serveriais.

2.3. Slaptažodžio įvesties laukų analizavimas vartotojo naršyklėje

Norint išsiaiškinti, ar tinklapis siekia apgaulingai išgauti vartotojo duomenis, reikia nustatyti, ar tinklapyje egzistuoja duomenų įvesties forma. Siekiant tai išsiaiškinti, bus analizuojamas tinklapio išvesties kodas. Kadangi šiame darbe koncentruojamasi į prisijungimo formas, tinklapio išvesties kode bus ieškoma slaptažodžio įvesties lauko. HTML kalboje, slaptažodžių įvesties laukai žymimi *type="password"* atributu.

Identifikavus, jog tinklapyje egzistuoja toks elementas, laukiama, kol šis laukas yra aktyvuojamas vartotojo. Tai yra reikalinga dėl to, nes įvesties laukas tam tikru laiko momentu gali būti nematomas, pavyzdžiui, už tuo metu rodomo tinklapio fragmento ribų.

Nustačius, jog slaptažodžio įvesties formos laukas yra aktyvus, bus nustatomos šio lauko koordinatės tuo metu atvaizduojamo tinklapio srityje. Ši informacija vėliau bus naudojama lyginant reliatyvią įvesties lauko, logotipo bei surastų frazių poziciją, siekiant patvirtinti, jog elementų išdėstymas atitinka kriterijus. HTML elementų reliatyvią poziciją galima gauti naudojant „JavaScript“ metodus.

Galiausiai, išsaugomas tuo metu vartotojui pateikimo tinklapio fragmento vaizdas. Visos populiariausios interneto naršyklės turi API (angl. *Application programming interface*) prieigą, skirtą tinklapio vaizdo išsaugojimui.

2.4. Logotipų identifikavimas tinklapio vaizde

Norint identifikuoti konkrečius logotipus tinklapio vaizde, reikalingas objektų atpažinimo algoritmas. Nors egzistuoja daug algoritmų, šiuo metu geriausią tikslumą bei greitaveiką pasiekia dirbtinius neuroninius tinklus naudojantys algoritmai. Šiuo metu plačiausiai naudojami yra „Faster R-CNN“, „YOLO“ bei „SSD“.

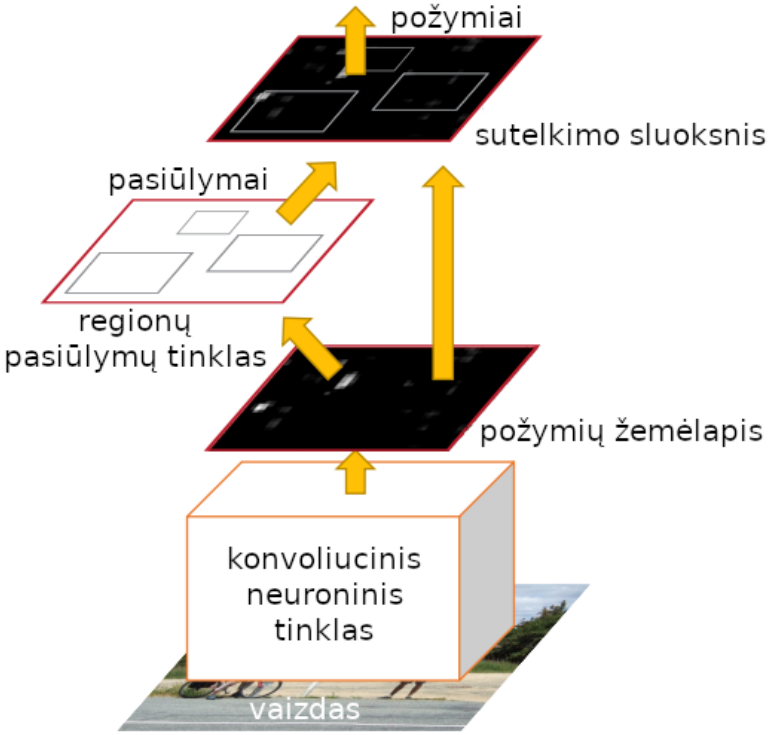
„Faster R-CNN“ algoritmas buvo pasiūlytas „Microsoft“ tyrėjų grupės 2015 metais. Juo siekiama išspręsti greitaveikos problemas, su kuriomis susidūrė praeitos šio algoritmo generacijos – „R-CNN“ bei „Fast R-RNN“. Norint suprasti šio algoritmo veikimo principą, kartu reikia panagrinėti ir praeitų generacijų veikimo principą [24].

Pirmoji algoritmo karta, „R-CNN“ veiksmus atlieka trimis skirtingais etapais. Pirmame etape, naudojant atrankinį paieškos algoritmą (angl. *selective search*), kuris atrenka 2000 regionų, kuriuose yra didžiausia tikimybė aptikti objektą. Sekančiame etape, rasti regionai yra apdorojami konvoliucinio neuroninio tinklo, kuris kiekvienam regionui sugeneruoja bruožų vektorių. Galiausiai, gauti bruožų vektoriai yra klasifikuojami tradiciniu SVM algoritmu [24].

Sekanti algoritmo karta, pavadinimu „Fast R-CNN“, buvo patobulinta siekiant pasiekti didesnę greitaveiką. Iš pradžių visa nuotrauka yra apdorojama neuroninio tinklo, kuris sugeneruoja bruožų vektorių matricą. Iš šių bruožų, nustatomi objektų regionai, bei bei apdorojami viena iteracija. Tai yra įmanoma, nes tie patys skaičiavimų rezultatai naudojami skirtingiems regionams analizuoti. Vaizdo klasifikavimas atliekamas paskutiniame neuroninio tinklo „softmax“ sluoksnyje, vietoj anksčiau naudoto SVM klasifikatoriaus [24].

Trečiojoje algoritmo kartoje, „Faster R-CNN“, buvo atsisakyta atrankinės paieškos objektų regionų paieškai. Vietoj to, objektų regionai randami naudojant regionų pasiūlymų tinklą (angl. *region*

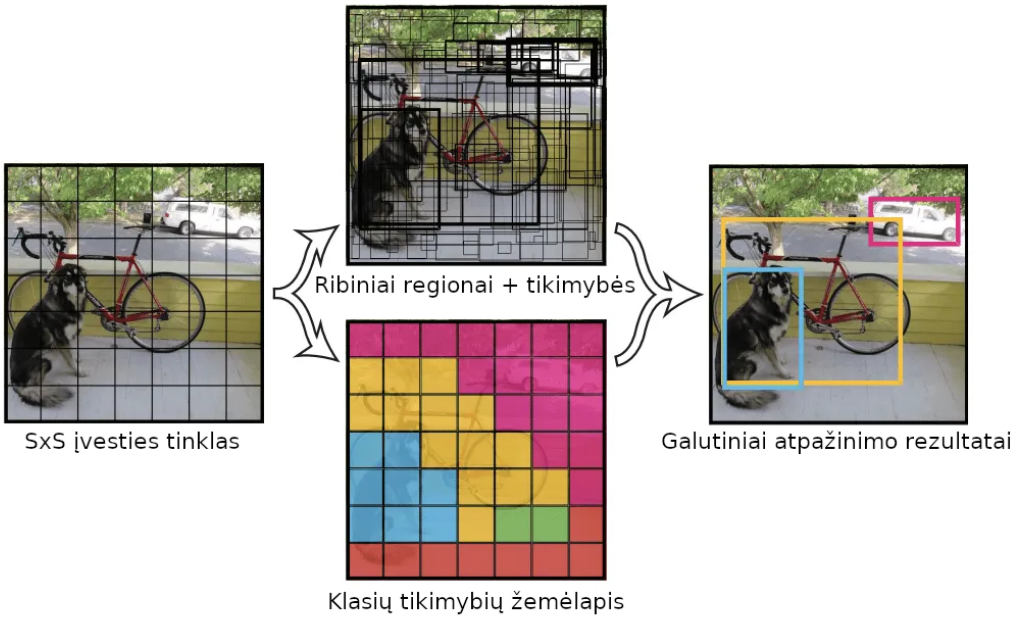
proposal network, RPN). Tai yra atskiras neuroninis tinklas, dedikuotas objektų atpažinimui. Lyginant su atrankine paieška, šis metodas yra žymiai greitesnis. 8 pav. pateikta šio algoritmo varianto principinė struktūra.



8 pav. „Faster R-CNN“ algoritmo struktūra [24]

Dažnu atveju yra naudojamas iš anksto išmokytas neuroninis tinklas, pavyzdžiui, „ResNet50“, ir iš naujo išmokomas galutinis neuroninio tinklo lygis. Tai leidžia išmokyti tinklą naudojant mažiau duomenų, bei naudojant gerokai mažiau skaičiavimo resursų. Tai yra įmanoma, nes neuroninio tinklo pirmieji sluoksniai įprastai skirti žemo lygio požymių nustatymui, tokių kaip primityvios geometrinės formos. Tolimesniuose sluoksniuose naudojami vis labiau abstraktūs bruožai. Kadangi Žemo lygio požymiai yra bendri visų tipų objektams, neuroninio tinklo sluoksnius galima perpanaudoti [24].

Kitas plačiai naudojamas objektų atpažinimo algoritmas yra „YOLO“. Algoritmo pavadinimas reiškia „pažvelgsi tik kartą“ (angl. *You Only Look Once*). Pasirodžius pirmajai algoritmo versijai jis išsiskyrė tuo, jog nenaudoja atskirų algoritmų ar modelių, nuspėjančių galimas objektų ribas ir vėliau bando klasifikuoti kiekvieną iš šių regionų. Vietoj to, tas pats neuroninis tinklas naudojamas objektų koordinačių aptikimui bei objektų klasifikavimui. 9 pav. pateiktas supaprastintas algoritmo veikimo principas. Iš pradžių nuotrauka yra padalinama į fiksuoto dydžio tinklą. Kiekvienam iš šių regionų yra paskaičiuojamos tikėtinos objekto koordinatės, bei tikimybė, jog objektas priklauso tam tikrai klasei. Galiausiai yra atmetami objektai, kurie dubliuojasi, arba turi nepakankamą tikimybę [26].



9 pav. „YOLO“ algoritmo veikimo principas [26]

Nuo pirmosios algoritmo versijos, buvo išleista daug atnaujinimų bei patobulinimų, kurie tiek didino aptikimo tikslumą, ypač smulkių objektų, tiek gerino aptikimo bei mokymo greitį. Šiuo metu naujausia algoritmo versija yra „YOLO v8“ [26]. Ši algoritmo versija turi paruoštus 5 skirtingus modelių variantus. Pagrindinis skirtumas tarp jų yra parametrų skaičius. Didesnis parametrų skaičius lemia geresnį tikslumą, tačiau tuo pačiu metu reikalauja didesnių skaičiavimų resursų. 1 lentelėje pateiktas skirtingų modelių palyginimas – parametrų skaičius, tikslumas naudojant „COCO“ duomenų rinkinį bei vidutinė analizės trukmė atpažinimui naudojant CPU.

1 lentelė. „YOLO v8“ modelių palyginimas [27]

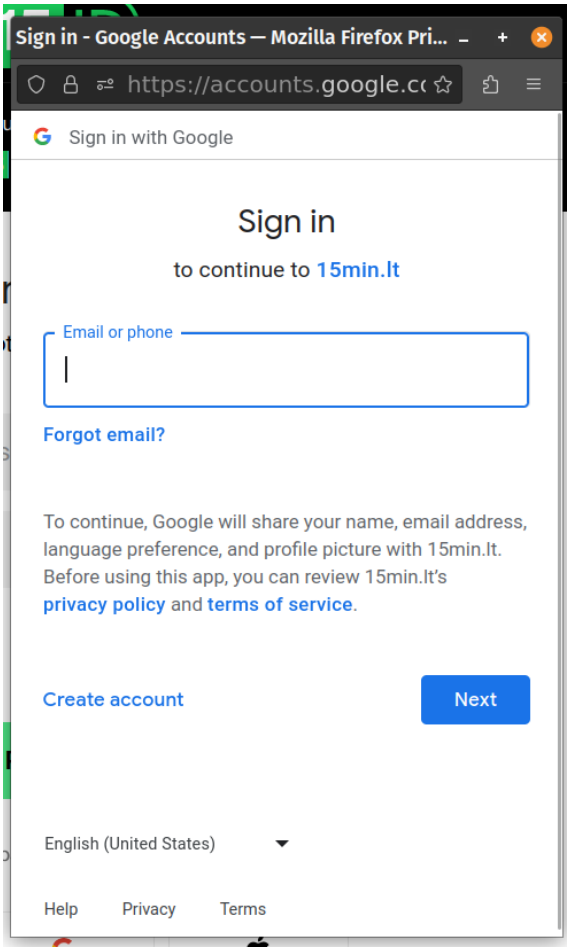
Modelis	**mAP50-95 įvertis**	**Vid. atpažinimo trukmė naudojant CPU, ms**	**Parametrų skaičius, mln.**
yolov8n	37,3	80,4	3,2
yolov8s	44,9	128,4	11,2
yolov8m	50,2	234,7	25,9
yolov8l	52,9	375,2	43,7
yolov8x	53,9	479,1	68,2

Dar vienas plačiai naudojamas objektų aptikimo algoritmas yra „SSD“. Analogiškai kaip ir „YOLO“, objektų atpažinimas atliekamas per vieną iteraciją. Iš pradžių iš vaizdo yra išskiriami požymiai, naudojant įprastą konvoliucinį neuroninį tinklą, tokį kaip „VGG-16“. Toliau, naudojami keli, progresyviai mažėjančios raiškos konvoliuciniai sluoksniai, gebantys aptikti skirtingų dydžių objektus. Ši skirtingų dydžių sluoksnių architektūra išsiskiria iš kitų objektų atpažinimo algoritmų, ir dėl jos algoritmas geba tiksliau aptikti skirtingų dydžių objektus. Kiekvienas iš šių sluoksnių sugeneruoja galimų objektų regionų sąrašą, bei apskaičiuoja, kokia tikimybė jog konkrečios klasės

objektas egzistuoja šiame regione. Toliau yra atmetami besidubliuojantys regionai, bei tie, turintys per mažą objekto egzistavimo tikimybę [28].

Lyginant šiuos tris algoritmus, „YOLO“ bei „SSD“ pasižymi gerokai didesne greitaveika – dėl to puikiai tinka panaudojimo atvejams kai reikia atpažinimo realiu laiku. Tačiau „Faster R-CNN“ algoritmas yra pranašesnis tuo, jog pasiekia didesnį tikslumą. Įprasto dydžio objektų atpažinimo tikslumas skiriasi minimaliai, tačiau mažų objektų atpažinime „Faster R-CNN“ pasiekia pastebimai didesnį tikslumą [29].

Išanalizavus populiariausių vieno prisijungimo sistemų tiekėjų prisijungimo formas pastebėta, jog kai kuriose iš jų logotipų dydis yra palyginus mažas. Pavyzdžiui, „Google“ prisijungimo formoje logotipas yra tik 16×16 pikselių dydžio (žr. 10 pav.).



10 pav. „Google“ prisijungimo forma

Šiuo atveju kuriamai sistemai nėra ypač svarbu, jog algoritmas pasiektų kuo didesnę greitaveiką. Kadangi vaizdo analizavimas prasidės vartotojui pradėjus įvedinėti duomenis į prisijungimo formą, analizavimas galėtų užtrukti iki 3 sekundžių, vartotojui nepatiriant nepatogumų. Dėl šių priežasčių logotipų atpažinimui buvo pasirinktas „Faster R-CNN“ algoritmas.

2.4.1. Duomenų rinkiniai

Logotipų atpažinimo modelio išmokymui reikalingas anotuotas duomenų rinkinys. Egzistuoja nemažai logotipų duomenų rinkinių, tačiau daugelį jų sudaro didelis klasių kiekis su nedideliu kiekiu vaizdų kiekvienai klasei. Kadangi šiame darbe koncentruojamasi į populiariausių SSO tiekėjų logotipų atpažinimą, iš esamų duomenų rinkinių bus išsirenkami tik reikiamų klasių logotipai.

Bus naudojami šie duomenų rinkiniai:

- LogoDet-3K [30];
- FlickrLogos-47 [31];
- QMUL-OpenLogo [32];
- LogoSENSE [21].

Iš duomenų rinkinių, bus išrinkti šių prekės ženklų logotipai:

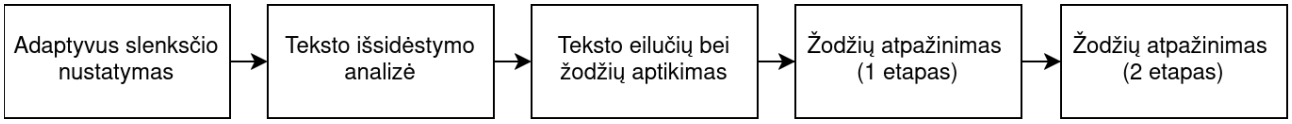
- „Google“;
- „Facebook“;
- „Microsoft“;
- „PayPal“;
- „Steam“.

Šie prekės ženklai pasirinkti dėl to, jog egzistuoja populiariausiuose logotipų duomenų rinkiniuose, bei turėtų padengti daugelį „BitB“ atakose naudojamų vieno prisijungimo tiekėjų formų [33].

2.5. Teksto atpažinimas naudojant OCR procesą

Siekiant apeiti piktavalių dažnai taikomus turinio užtemdymo metodus, tinklapyje pateikiamo teksto nuskaitymui bus naudojamas OCR procesas, skirtas teksto atpažinimui iš nuotraukų.

Šiai užduočiai atlikti pasirinktas „Tesseract OCR“ įrankis. Jis yra atviro kodo, turi galimybę pasirinkti iš keleto skirtingų atpažinimo metodų, bei suteikia API lengvai integracijai į kitas programas. 11 pav. pateikta principinė įrankio veikimo schema. Pirmuoju etapu vykdoma pradinis nuotraukos apdorojimas – nustatoma slenksčio riba, pagal kurią vaizdas konvertuojamas į juodai baltą. Sekančiame etape nustatoma teksto orientacija vaizde, nustatomi regionai kuriuose yra tekstas. 3 žingsnyje aptinkamos teksto eilutės, bei atskiriami individualūs žodžiai pagal tarpus tarp jų. Galiausiai vykdomas žodžių atpažinimas dviem etapais, naudojant KNN klasifikatorių [34].



11 pav. „Tesseract OCR“ įrankio veikimo schema [34]

Nuskaičius tinklapyje pateikiamą tekstą, jis bus analizuojamas, ieškant konkrečių teksto fragmentų. Vienas iš pagrindinių „BitB“ atakos požymių yra tas, jog vaizduojamame netikrame naršyklės lange yra rodomas URL adresas, atitinkantis tikros prisijungimo formos URL. Pavyzdžiui, „Google“ prisijungimo forma pateikiama adresu „https://accounts.google.com“, „Facebook“ –

„https://www.facebook.com/login.php“ ir pan. Tekste radus tokį URL, tai būtų vienas iš požymių, jog yra vaizduojama netikra prisijungimo forma.

Kitas požymis, pagal kurį galima identifikuoti prisijungimo formas, yra konkrečių teksto fragmentų naudojimas. Pavyzdžiui, „Facebook“ prisijungimo formoje naudojamas tekstas „*Login to use your Facebook account with* <...>“, „Google“ prisijungimo formoje – „*Sign in with Google*“, „*To continue, Google will share your name, email address* <...>“ ir pan.

Konkrečių prisijungimo formų URL adresai, teksto fragmentai bus atrinkti rankiniu būdu bei saugomi duomenų bazėje.

Atliekant teksto nuskaitymą naudojant OCR negalima tikėtis 100 % tikslumo. Taip pat įmanoma, jog piktavalių sukurtoje formoje tekstas bus modifikuotas – bus pakeisti keli žodžiai. Todėl identiškų atitikmenų paieška dažnu atveju gali būti neefektyvi. Dėl šios priežasties, atliekant teksto palyginimą, reikia įvertinti teksto panašumą.

Teksto panašumui palyginti bus naudojamas kosinusinio panašumo (angl. *cosine similarity*) rodiklis. Šis rodiklis yra apskaičiuojamas kiekvieną žodį konvertuojant į jį reprezentuojantį vektorių. Atitinkamų tekstų vektorių reikšmės yra susumuojamos. Tuomet yra apskaičiuojama kosinuso reikšmė naudojant kampą tarp šių dviejų vektorių. Gautas rezultatas yra teksto panašumo įvertinimas. Reikšmė lygi 1 reiškia identišką tekstą.

Riba, pagal kurią bus įvertinama, ar rasti teksto fragmentai yra pakankamai panašūs, bus nustatoma eksperimentiniu būdu, atsižvelgiant į OCR nuskaitytame tekste pasitaikančių klaidų skaičių.

2.6. Galutinis rezultatų interpretavimas

Galutiniam rezultatui priimti, ar tinklapyje yra vaizduojamas netikras naršyklės langas su prisijungimo forma, imituojančia vieno prisijungimo sistemos tiekėjų formą, turimi šie duomenys:

- slaptažodžio įvedimo lauko egzistavimas bei jo pozicija;
- tinklapyje vaizduojami logotipai bei jų pozicijos;
- tinklapyje pateikti prisijungimo formų URL adresai bei jų pozicijos;
- tinklapyje aptikti teksto fragmentai iš konkrečių prisijungimo formų, bei šių fragmentų pozicija.

Norint užtikrintai teigti, jog tinklapis bando imituoti populiaraus vieno prisijungimo sistemos tiekėjo prisijungimo formą, turėtų būti išpildyti visi kriterijai – surastas konkretaus tiekėjo logotipas, rasta to pačio tiekėjo prisijungimo formos nuoroda bei rasti formoje naudojami teksto fragmentai. Tačiau iškėlus tokias sąlygas, galima numanyti, jog aptikimo tikslumas būtų nepakankamas.

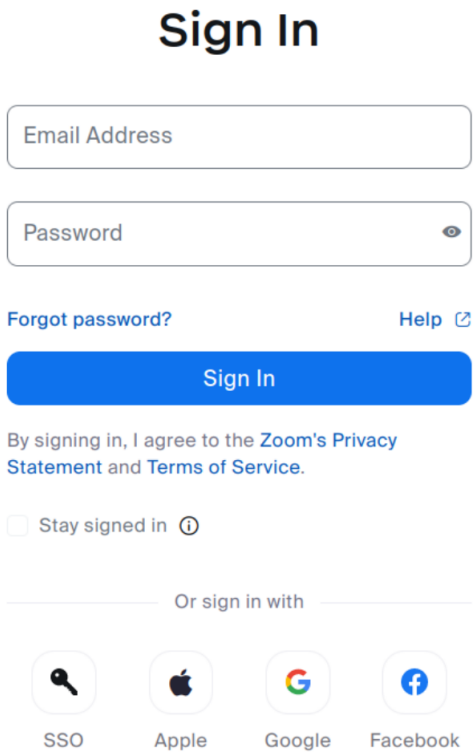
Visų pirma, piktavalių sukurta prisijungimo forma nebūtinai bus identiška tikrai prisijungimo formai. Formoje nebūtinai bus pateiktas identiškas tekstas, nebūtinai pateiktas logotipas ar URL adresas. Taip pat logotipų atpažinimas bei OCR rezultatas nebus 100 % tikslūs. Dėl to reikia apsibrėžti, kokie aptikti požymiai yra pakankami, jog būtų galima teigti, kad tinklapio vaizde yra pateikta netikra prisijungimo forma.

Pirmas dalykas, ką reikia nustatyti, yra tai, ar aptiktų elementų reliatyvi pozicija atitinka realioje prisijungimo formoje esančių elementų išdėstymą. Kaip pavyzdį, galima paanalizuoti „Google“

prisijungimo formą (žr. 10 pav.). Iššokančiajame prisijungimo lange elementai yra išsidėstę tokia tvarka, pradedant nuo viršaus:

- tinklapio antraštė („Sign in – Google Accounts“);
- tinklapio adresas („https://accounts.google.com…“);
- prisijungimo formos logotipas;
- teksto fragmentai („Sign įn with Google“, „Sign in to continue to …“);
- įvesties formos laukas;
- teksto fragmentai („To continue, Google will share your name, email address, <...>“);

Elementų reliatyvios pozicijos įvertinimas yra svarbus aspektas, kadangi to neįvertinus, galima tikėtis daug klaidingų teigiamų rezultatų. Tai ypač svarbu vertinant aptiktų logotipų poziciją. Didelė dalis tinklapių leidžia vartotojams prisijungti naudojant vieno prisijungimo sistemą. Dėl to prie įprastų prisijungimo formų, kurios nėra sukurtos siekiant imituoti kito paslaugų tiekėjo prisijungimo formą, yra dažnai pateikti mygtukai su logotipais (žr. 12 pav.). Šie mygtukai dažniausiai yra pateikiami žemiau prisijungimo formos įvesties laukų, kai tuo tarpu logotipai, identifikuojantys pačią prisijungimo formą, dažniausiai pateikiami viršutinėje dalyje, aukščiau įvesties laukų.



12 pav. Prisijungimo formos pavyzdys su logotipais

Žinoma, kiekvienos formos dizainas yra individualus, tad nors šie požymiai gali būti taikomi daugeliui formų, individualūs atvejai gali skirtis. Dėl to yra poreikis saugoti informaciją apie kiekvienos formos reliatyvų elementų išdėstymą.

Vertinant išsidėstymą, bus atsižvelgiama tik į y koordinatę, kadangi keičiant lango dydį, elementai keičia savo poziciją x koordinatės atžvilgiu. Taip pat bus vertinama ne konkreti koordinatės reikšmė, bet reliatyvi pozicija – aukščiau arba žemiau konkretaus elemento. Dėl to apie kiekvieną elementą bus saugoma skaitinė reikšmė, reikšianti lygį. Reikšmė 0 atitinka žemiausią elementą, 1 –

aukščiau esantį elementą ir t.t. Keli elementai gali turėti tokį patį lygį – tokiu atveju jų tarpusavio pozicija nebus vertinama.

Šie duomenys apie konkrečias prisijungimo formas, kurių atitikmenų bus ieškoma, bus surinkti rankiniu būdu. Kadangi darbe koncentruojamasi į populiariausius vieno prisijungimo sistemų tiekėjus, reikiamų surinkti duomenų kiekis nebus didelis. Tačiau visada yra galimybė šį procesą automatizuoti.

Apie kiekvieną prisijungimo formą bus saugojami tokie duomenys:

- pavadinimas (atitinka logotipo pavadinimą „Faster R-CNN“ modelyje);
- tinklapio adresas (naudojamas netikro adreso aptikimui);
- formoje naudojamų frazių sąrašas;
- reliatyvi kiekvieno objekto *y* koordinatės pozicija.

Gavus rezultatus iš logotipų aptikimų algoritmo bei iš OCR įrankio, jie bus lyginami su duomenų bazėje saugomais duomenimis, ieškant atitikmenų. Galutinis sprendimas, jog tinklapyje yra pateikta netikra prisijungimo forma, bus priimtas jeigu bus tenkinamos tokios sąlygos:

- aptikti ne mažiau kaip 3 požymiai, priklausantys tai pačiai prisijungimo formai – URL adresas, logotipas bei bent vienas teksto fragmentas;
- atpažintų objektų išsidėstymas tinklapyje atitinka duomenų bazėje saugomą išsidėstymą.

2.7. Apibendrinimas

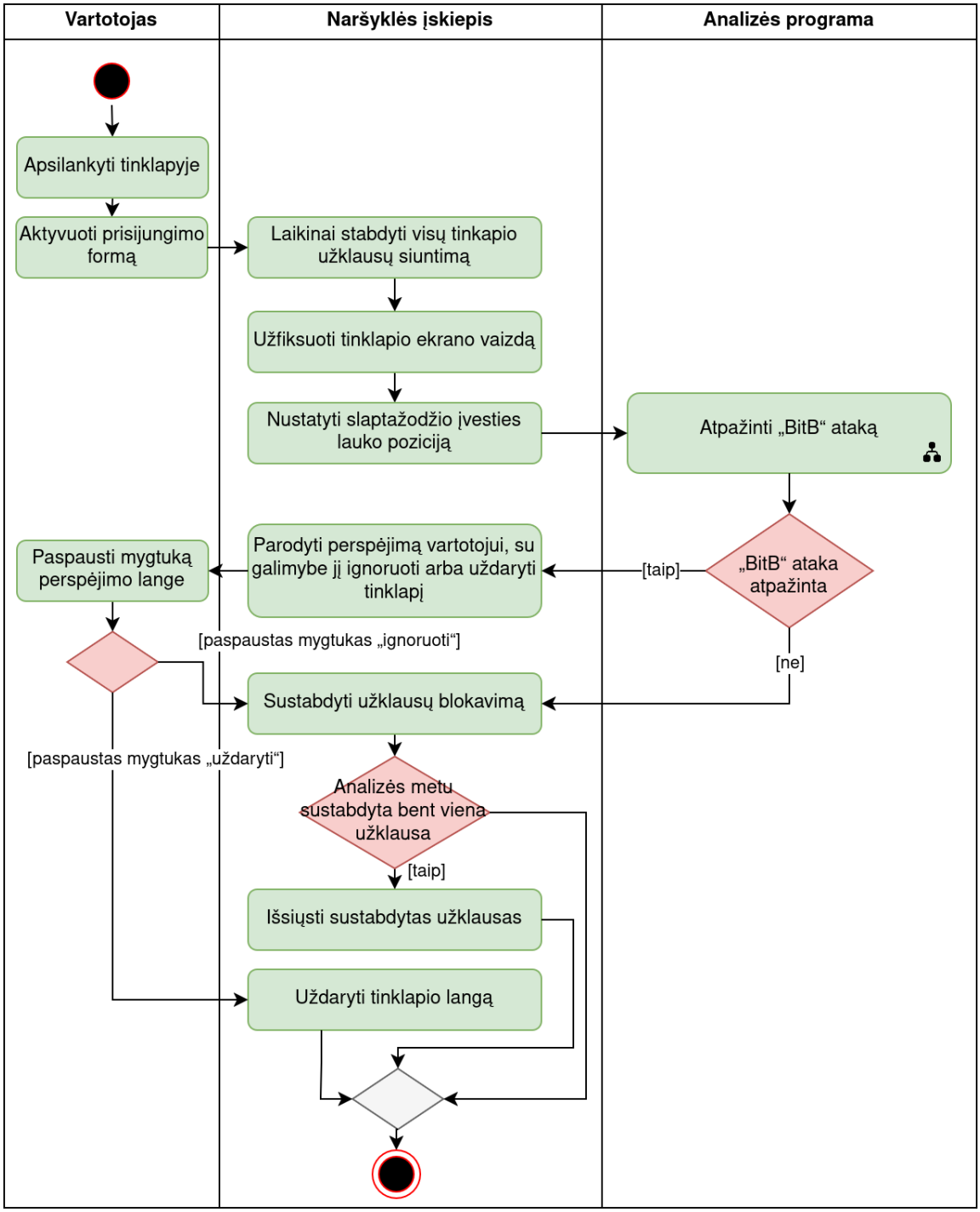
Pasiūlytas atakos atpažinimo metodas remiasi netikros prisijungimo formos identifikavimu pagal vizualius požymius. Kaip požymiai naudojami logotipai, teksto fragmentai ir URL adresai, rasti tinklapio vaizde. Palyginus skirtingus objektų atpažinimo vaizduose algoritmus, kaip tinkamiausias pasirinktas „Faster R-CNN“. Siūloma sistema veikia vartotojo įrenginyje bei nereikalauja papildomų vartotojo veiksmų, jog būtų inicijuotas galimos atakos atpažinimas, bei aptikus galimą ataką, blokuoja tinklapio siunčiamas užklausas, jog įvesti prisijungimo duomenys nebūtų išsiųsti.

3. „BitB“ atakos atpažinimo metodo realizacija

Šiame skyriuje pateikiama „BitB“ atakos atpažinimo metodo realizacija, paremta 2 skyriuje aprašytu projektu.

3.1. „BitB“ atakos atpažinimo sistemos modelis

13 pav. pateikta pagrindinė sistemos veikimo veiklos diagrama. Sistemą sudaro dvi pagrindinės dalys – interneto naršyklės įskiepis, bei analizės programa, kuri veikia kaip atskiras procesas, bei su interneto naršyklės įskiepiu komunikuoja per specifinę naršyklės sąsają.



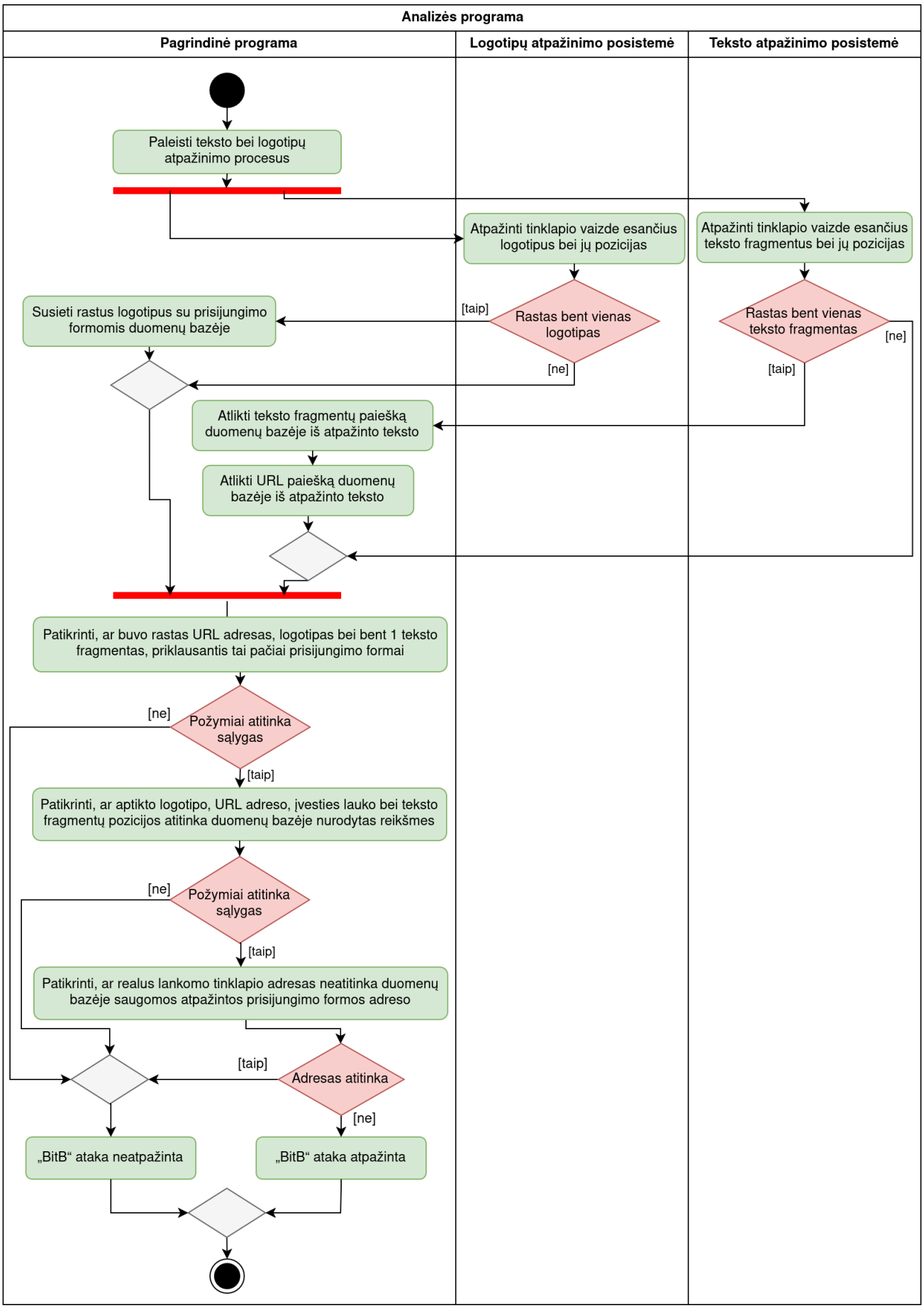
13 pav. „BitB“ atakos atpažinimo sistemos veiklos diagrama

Vartotojui naršant internete, naršyklės įskiepis stebi vartotojo veiksmus, bei laukia, kol tinklapyje pasirodys prisijungimo forma, skirta įvesti jautriems duomenims (prisijungimo slaptažodžiui). Aptikus, jog tokia forma egzistuoja, sekami vartotojo veiksmai. Vartotojui pažymėjus jautrių duomenų įvesties lauką, pradedamas galimos atakos prevencijos, požymių rinkimo bei analizės procesas.

Visų pirma, konkrečiam naršyklės langui, kuriame atidarytas analizuojamas tinklapis, yra laikinai blokuojamos visos išeinančios užklausos. Tai daroma todėl, jog jautrūs duomenys nebūtų išsiųsti į serverį iki tol, kol nebus baigtas analizės procesas. Toliau naršyklės įskiepis užfiksuoja tinklapio ekrano kopiją, nustato slaptažodžio įvesties lauko poziciją bei perduoda šiuos duomenis analizės programai. Šioje programoje atliekamas atakos atpažinimo procesas, bei gražinamas atsakymas, ar ataka buvo atpažinta (atpažinimo procesas detalizuotas atskira diagrama). Jei ataka nebuvo atpažinta, išjungiamas užklausų blokavimas, bei išsiunčiamos užklausos, kurios buvo užblokuotos atliekant analizę. Kitu atveju, jei ataka buvo atpažinta, vartotojui yra parodomas dialogo langas su perspėjimu apie atpažintą ataką. Vartotojas šiuo atveju turi du pasirinkimus – ignoruoti perspėjimą spaudžiant mygtuką „ignoruoti“, arba uždaryti įtartiną tinklapį paspaudus mygtuką „uždaryti“. Vartotojui pasirinkus perspėjimą ignoruoti, veiksmų seka analogiška lyg ataka nebūtų atpažinta.

14 pav. pateikta detalizuota „BitB“ atakos atpažinimo proceso veiklos diagrama. Iš pradžių lygiagrečiai paleidžiami du procesai. Pirmasis iš jų atpažįsta tinklapio vaizde esančius logotipus bei jų pozicijas. Antras procesas atpažįsta tinklapio vaizde esantį tekstą bei jo poziciją. Vaizde surasti logotipai yra susiejami su duomenų bazėje aprašytomis prisijungimo formomis. Iš tinklapio vaizdo atpažintame tekste yra ieškoma dviejų tipų požymių. Pirmasis požymis – URL adresas, kuris „BitB“ atakos atveju yra vaizduojamas netikrame naršyklės lange. Duomenų bazėje aprašytas kiekvienos prisijungimo formos unikalus adresas, ir iš aptikto teksto bandoma surasti šių adresų atitikmenis duomenų bazėje, susiejant juos su konkrečia prisijungimo forma. Antras požymis – teksto fragmentai. Duomenų bazėje aprašyti konkrečioje prisijungimo formoje naudojami teksto fragmentai, ir tinklapio tekste bandoma surasti jų atitikmenis, susiejant su konkrečia prisijungimo forma.

Šiame etape yra surinkti visi požymiai, kuriuos interpretuojant galima atpažinti „BitB“ ataką – tinklapio tekste rastų adresų sąrašas, logotipų sąrašas, teksto fragmentų sąrašas, bei visų šių elementų pozicija vaizde bei kokiai prisijungimo formai jie priklauso. Taip pat žinoma įvesties lauko pozicija vaizde. Jeigu tarp požymių nebuvo rastas bent vienas logotipas, URL adresas bei teksto fragmentas, priklausantys tai pačiai prisijungimo formai, laikoma, jog požymių nepakanka ir „BitB“ ataka nėra vykdoma. Jei buvo rasta bent po vieną iš šių požymių, toliau atliekamas tikrinimas, ar šių elementų pozicijos atitinka duomenų bazėje aprašytą konkrečios prisijungimo formos elementų išsidėstymą. Jeigu buvo rasta daugiau nei po vieną tokį patį požymį konkrečiame tinklapyje (pavyzdžiui, atpažinti keli vienodi logotipai), laikoma, jog išsidėstymas yra teisingas, bei bent vienas iš šių požymių yra teisingoje pozicijoje.



14 pav. „BitB“ atakos atpažinimo proceso veiklos diagrama

Jeigu elementų išsidėstymas teisingas, galiausiai atliekamas patikrinimas, ar realus tinklapio adresas nesutampa su atpažintos prisijungimo formos adresu. Jei nesutampa – laikoma, jog buvo atpažinta „BitB“ ataka.

3.2. Logotipų atpažinimo posistemė

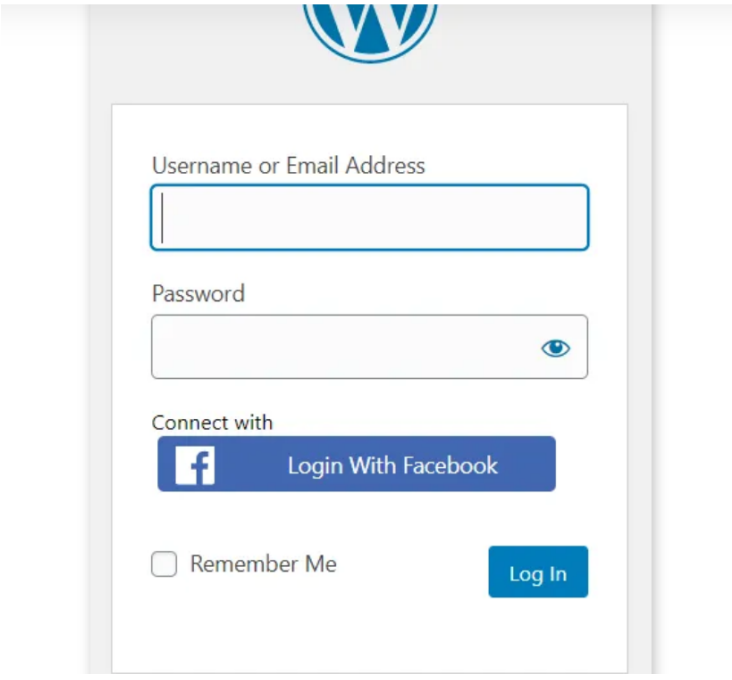
Norint atpažinti tinklapyje vaizduojamą prisijungimo formą, reikalinga identifikuoti vieną iš pagrindinių požymių – logotipą. Tam bus sudarytas duomenų rinkinys bei išmokytas objektų atpažinimo algoritmas.

3.2.1. Duomenų rinkinio sudarymas

Projekto metu buvo numatyta reikalingų logotipų duomenų rinkinį susidaryti iš jau esamų duomenų rinkinių, atrenkant paveikslus su aktualiomis duomenų klasėmis. Buvo atsisiųsti bei peržiūrėti šie duomenų rinkiniai: „*LogoDet-3K*“, „*FlickrLogos-47*“, „*QMUL-OpenLogo*“, „*LogoSENSE*“.

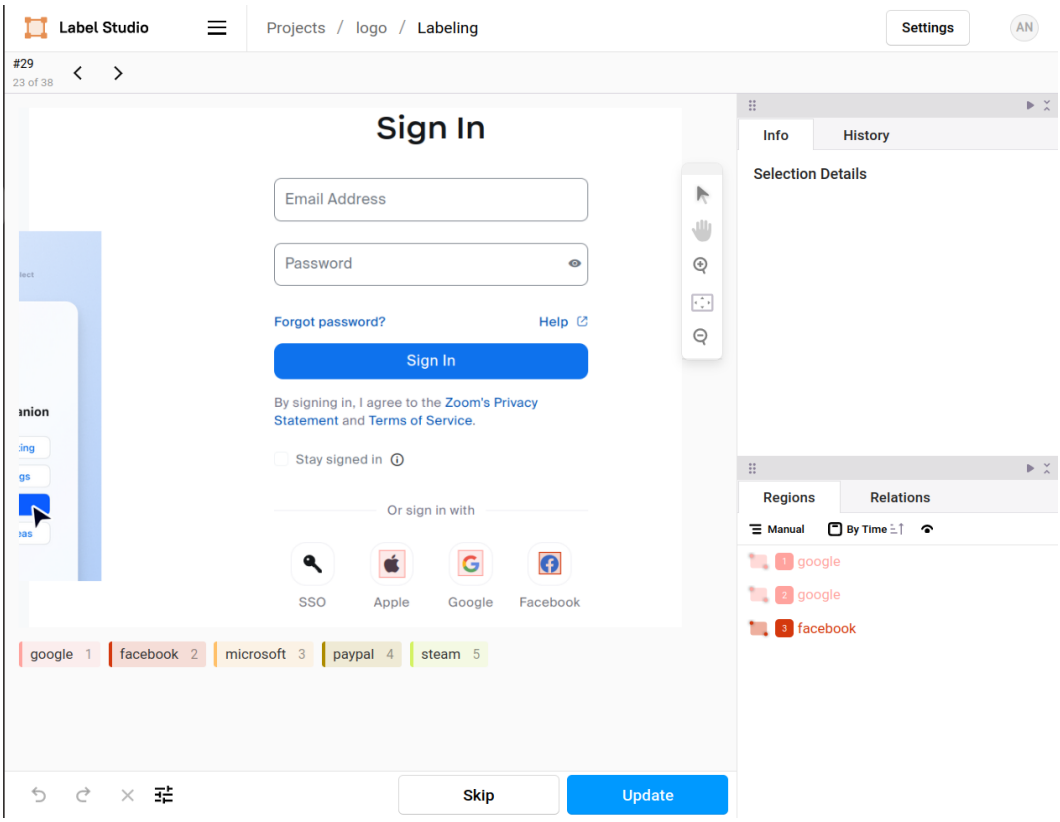
Peržiūrėjus esamus duomenų rinkinius pastebėta, jog juose esantys logotipai yra iš 2012 – 2018 metų laikotarpio. Peržiūrėjus šiuo metu naudojamus logotipus nustatyta, jog dabartinė logotipų išvaizda visiškai neatitinka duomenų rinkiniuose esančių logotipų. Dėl šios priežasties buvo nuspręsta sudaryti naują duomenų rinkinį, kuriame būtų naudojami naujausi, šiuo metu prisijungimo formose naudojami logotipai.

Kadangi logotipų atpažinimui pasirinktas „Faster R-CNN“ objektų atpažinimo algoritmas, jo išmokymui reikalingas anotuotas duomenų rinkinys. Naudojamas algoritmas sugeba ne tik nustatyti logotipo klasę, bet ir nustatyti kokioje konkrečioje zonoje jis yra išskiriant iš paveikslėlio fono. Dėl to yra svarbu, jog duomenų rinkinyje paveikslėliuose esantis foninis vaizdas tinkamai reprezentuotų realiomis sąlygomis įprastai esantį foną. Dėl šių priežasčių, duomenų rinkinys buvo sudarytas iš šiuo metu veikiančių tinklapių, kuriuose naudojami analizuojamų prisijungimo formų logotipai. 15 pav. pateiktas pavyzdys iš sudaryto duomenų rinkinio, kuriame naudojamas „Facebook“ logotipas.



15 pav. Duomenų rinkinio pavyzdys, kuriame naudojamas „Facebook“ logotipas

Paveikslų anotavimui buvo pasirinktas nemokamas, atviro kodo įrankis „Label Studio“. Įrankyje pridėtos 5 anotacijos klasės – *google, facebook, microsoft, paypal, steam*. Importavus paveikslėlių failus, buvo pažymėti visi aktualūs logotipai. 16 pav. pateikta anotavimui skirto įrankio vartotojo sąsaja bei anotuoto paveikslo pavyzdys.



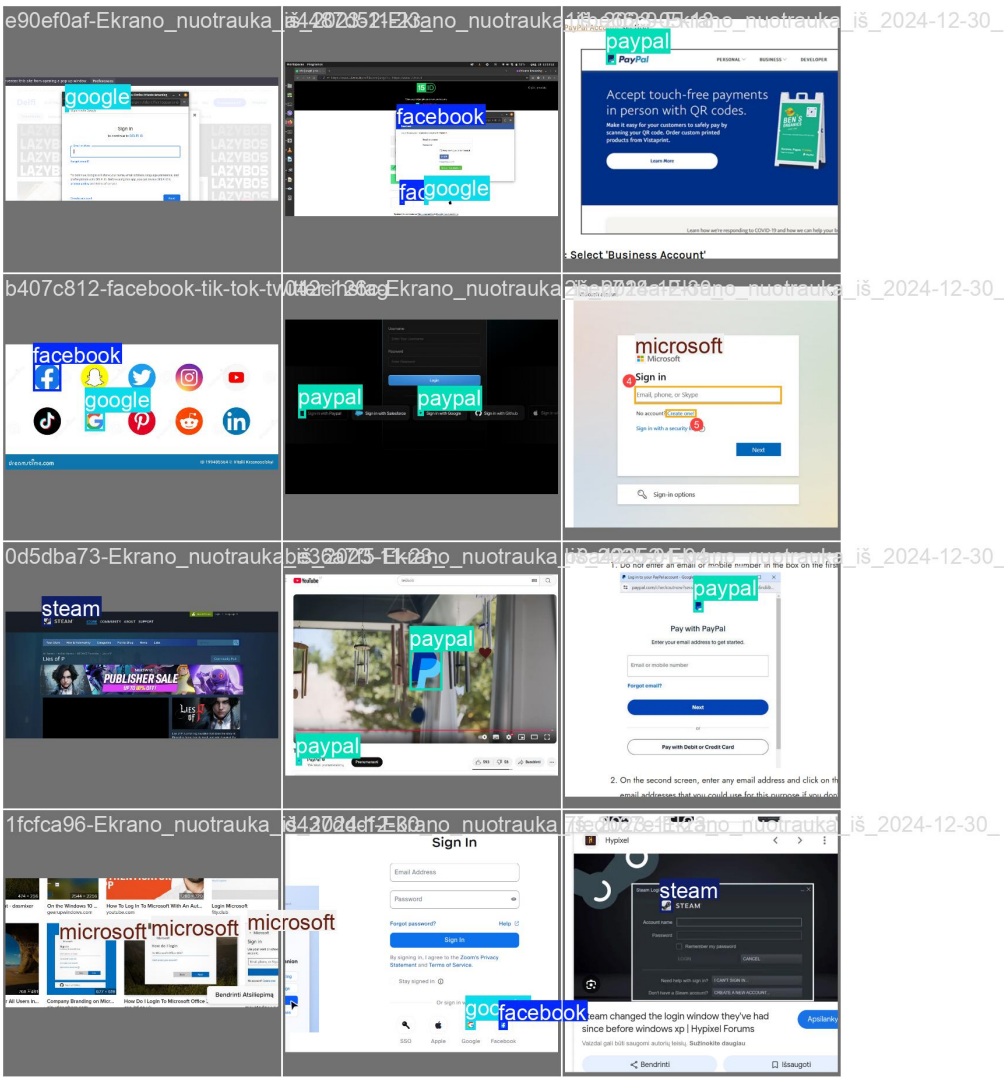
16 pav. „Label Studio“ įrankio vartotojo sąsaja

Duomenų rinkinys sudarytas iš 152 paveikslų. Kiekvienos klasės anotacijų skaičius, bei bendras visų kasių anotacijų skaičius pateiktas 2 lentelėje.

2 lentelė. Anotacijų skaičius sudarytame duomenų rinkinyje

Klasės pavadinimas	**Anotacijų skaičius**
google	64
facebook	63
microsoft	55
paypal	49
steam	41
Viso	**272**

17 paveiksle pateikta duomenų rinkinio pavyzdys, kuriame matomi paveikslai bei jų anotacijos su klasių pavadinimais.



17 pav. Sudaryto duomenų rinkinio pavyzdys atlikus anotaciją

Paveikslai kartu su anotacijomis buvo eksportuoti „COCO“ formatu.

3.2.2. Paveikslų apdorojimas prieš modelio mokymą

„Faster R-CNN“ algoritmas yra realizuotas bent keliose populiariausiose mašininio mokymosi karkasuose, tokiuose kaip „TensorFlow“ bei „PyTorch“. Šiam darbui pasirinktas „PyTorch“ įrankis dėl paprastesnio naudojimo.

Kaip pagrindas, paimtas pavyzdinis „PyTorch“ projektas, kuriame realizuota duomenų importavimo bei paveikslų apdorojimo logika.

Neuroninis tinklas kaip įvestį priima iš anksto nustatyto dydžio paveikslus. Todėl visi paveikslai, neatitinkantys numatyto dydžio, yra konvertuojami. Pagal egzistuojančią logiką, paveikslai konvertuojami, neišlaikant jų kraštinių santykio. Kadangi darbo metu sudarytame logotipų duomenų rinkinyje yra įvairios raiškos bei kraštinių santykių paveikslų, konvertavimo metu jie iškraipomi.

Mokymo metu pastebėta, jog atliekant tokį konvertavimą, gaunami rezultatai yra prasti. Po 50 epochų, vidutinis atpažinimo tikslumas (angl. *mean average precision*, mAP), naudojant 0,95

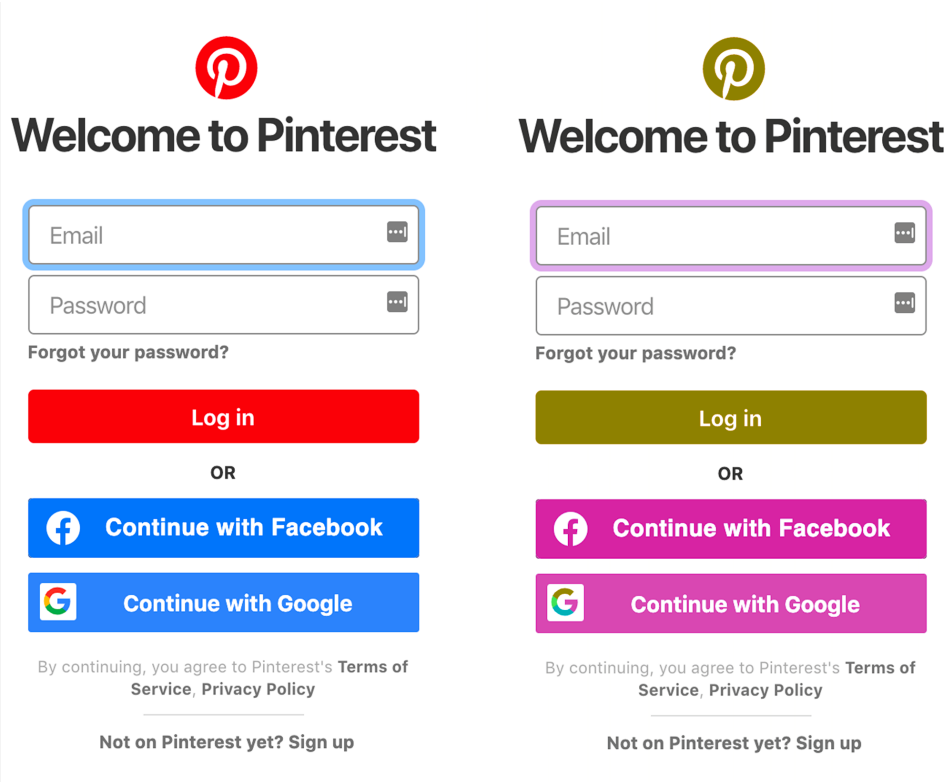
sankirtos ribą, nesiekė 30 %. Kadangi šiame darbe analizuojamos ne realaus pasaulio nuotraukos, o grafiniai elementai, jie turėtų išlaikyti kraštinių santykį.

Dėl to logika buvo modifikuota, jog keičiant paveikslų mastelį, būtų išlaikomas kraštinių santykis. Neužpildytas paveikslo plotas yra užpildomas baltu fonu. Tuo pačiu buvo modifikuotas ir paveikslų anotacijų koordinačių konvertavimas, jog jos atitiktų objektų koordinates pakeitus paveikslo mastelį.

Paveikslėlių paruošimui taip pat panaudotos įvairios transformacijos, kurių funkcionalumas realizuotas „PyTorch“ įrankio „transforms.py“ bibliotekoje. Transformacijos iš vieno paveikslėlių sugeneruoja kelis, pritaikant įvairius iškraipymus, siekiant pagerinti atpažinimo tikslumą įvairiomis sąlygomis. Buvo pasirinkta naudoti šiuos iškraipymus:

- vaizdo suliejimas;
- atsitiktinis kontrasto pakeitimas;
- spalvų iškraipymas;
- rūko efektas.

18 pav. pateiktas spalvų iškraipymo transformacijos pavyzdys.



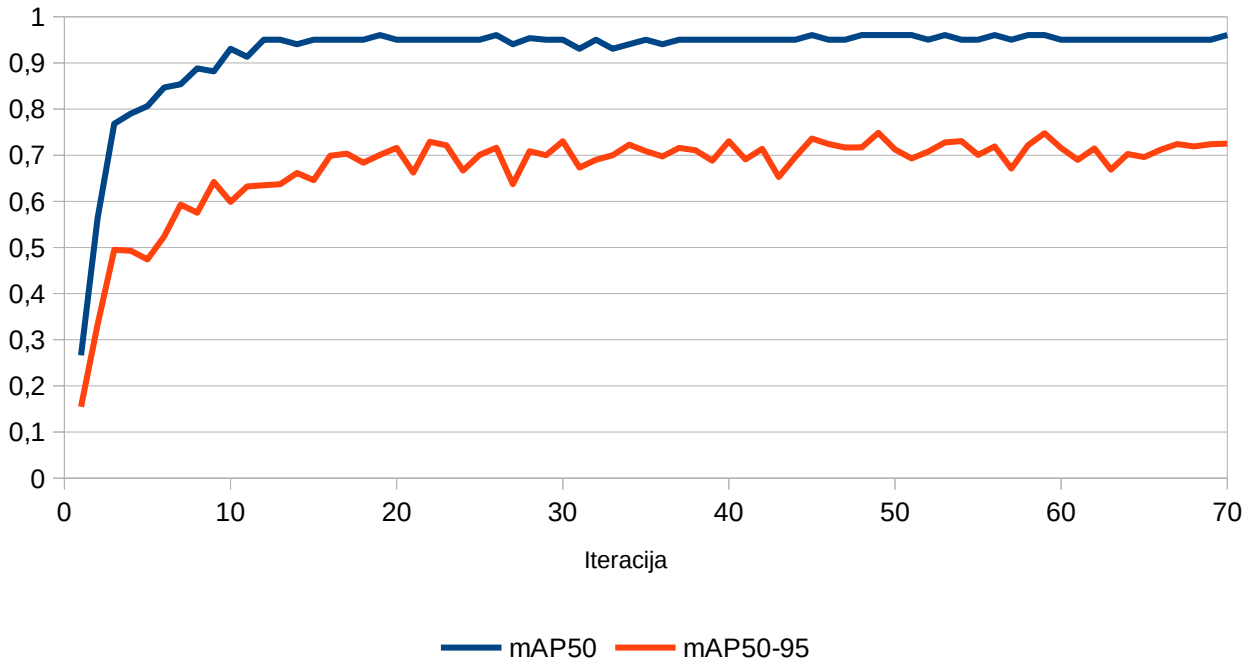
18 pav. Paveikslo transformacijos pavyzdys: kairėje pusėje originalus paveikslas, dešinėje - modifikuotas

3.2.3. Modelio mokymas

Anotuotas duomenų rinkinys buvo padalintas į dvi dalis 4:1 proporcija, atitinkamai modelio mokymui bei tikslumo įvertinimui. Modelio mokymas buvo vykdytas 70 iteracijų. Stebint mAP įverčio reikšmę pastebėta, jog maždaug ties 60 iteracija, modelis pasiekė didžiausią tikslumą, ir

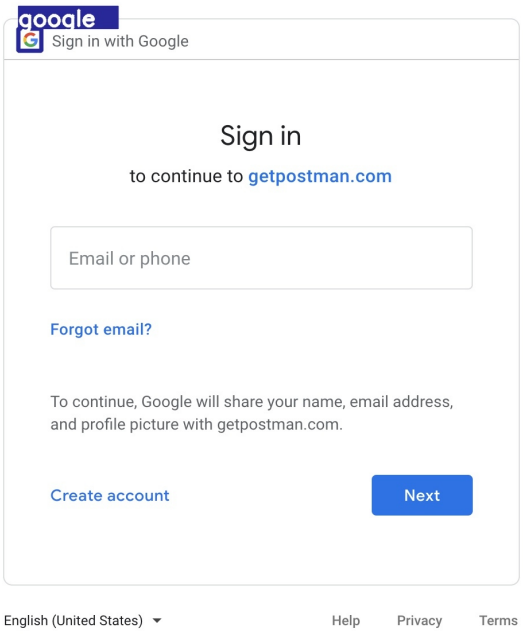
toliau nebedidėjo. Mokymas truko 18 valandų, skaičiavimams naudojant „Intel Core i5-8150U“ procesorių.

19 pav. pateiktas grafikas, vaizduojantis pasiektą atpažinimo tikslumą po kiekvienos iteracijos. Pasiekta 0,96 mAP (angl. *mean average precision*) įverčio reikšmė, naudojant 0,5 sankirtos ribą.



19 pav. Modelio tikslumas (mAP) prie 0,5 ir 0,95 sankirtos ribų po kiekvienos mokymo iteracijos

Išmokyto modelio veikimas buvo patikrintas analizuojant paveikslėlius, kurių nebuvo mokymui bei įvertinimui skirtuose duomenų rinkiniuose. 20 pav. pateikti išmokyto modelio atpažinti ir pažymėti logotipai.



20 pav. Išmokyto modelio veikimo pavyzdys

3.3. Prisijungimo formų požymių duomenų bazė

Norint identifikuoti konkrečią prisijungimo formą, reikia iš anksto identifikuoti jos požymius. Apie kiekvieną prisijungimo formą buvo surinkti tokie požymiai:

- logotipo klasė, atitinkanti klasę logotipų atpažinimo posistemėje bei jo pozicija;
- URL adresas bei jo pozicija;
- įvesties lauko pozicija;
- teksto fragmentai bei jų pozicijos.

Tolimesniame kodo fragmente pateikti surinkti požymiai apie visas analizuojamas prisijungimo formas:

```
formData['google'] = {"logo": [{"val": "google", "pos": 1}],
   "url": {"val":"https://accounts.google.com", "pos": 0},
   "input": {"val": "", "pos": 3},
   "text": [
      {"val": "Sign in with Google", "pos": 2},
      {"val": "To continue, Google will share your name, email address,", "pos": 4},
      {"val": "Before using this app, you can review,", "pos": 5}]}
formData['facebook'] = {"logo": [{"val": "facebook", "pos": 1}],
   "url": {"val":"https://www.facebook.com/login.php", "pos": 0},
   "input": {"val": "", "pos": 3},
   "text": [
      {"val": "Log in to use your Facebook account with", "pos": 2},
      {"val": "Keep me logged in to,", "pos": 4}]}
formData['microsoft'] = {"logo": [{"val": "microsoft", "pos": 1}],
   "url": {"val":"https://login.live.com/", "pos": 0},
   "input": {"val": "", "pos": 4},
   "text": [
      {"val": "Microsoft", "pos": 2},
      {"val": "Enter password", "pos": 3},
      {"val": "Because you're accessing sensitive info, you need to verify", "pos": 4} ]}
formData['steam'] = {"logo": [{"val": "steam", "pos": 1}],
   "url": {"val":"https://steamcommunity.com/", "pos": 0},
   "input": {"val": "", "pos": 3},
   "text": [
      {"val": "STORE COMMUNITY ABOUT SUPPORT", "pos": 2},
      {"val": "Use the Steam Mobile App to", "pos": 4}]}
formData['paypal'] = {"logo": [{"val": "paypal", "pos": 1}],
   "url": {"val":"https://paypal.com/signin", "pos": 0},
   "input": {"val": "", "pos": 3},
   "text": [
      {"val": "Email or mobile number", "pos": 2},
      {"val": "ContactUs Privacy Legal Policy Updates Worldwide", "pos": 3}]}
```

3.4. Teksto atpažinimo posistemė

Kadangi vaizdų atpažinimo sistema veikia „PyTorch“ karkaso pagrindu, kuris naudoja „Python3“ programavimo kalbą, dėl paprastesnės integracijos, kitos sistemos dalys bus realizuojamos „Python3“ programavimo kalba. Skyrelyje 2.4 teksto atpažinimui iš tinklapio nuotraukos buvo pasirinktas „Tesseract“ įrankis, integracijai su „Python3“ bus naudojama biblioteka „pytesseract“.

Atpažintas tekstas bus naudojamas URL adreso paieškai vaizde bei teksto fragmentų, būdingų konkrečiai prisijungimo formai, paieškai. Siekiant patikrinti, ar visų aptiktų elementų išsidėstymas yra būdingas konkrečiai prisijungimo formai, taip pat reikia žinoti kiekvieno teksto fragmento koordinates.

Naudojant „pytesseract“ bibliotekos funkcijas, gaunamas žodžių masyvas bei kiekvieno žodžio koordinatės. Jog būtų galima atlikti teksto fragmentų paiešką, žodžiai yra sugrupuojami pagal eilutės numerį (*y* koordinatę).

Kadangi teksto atpažinimas gali būti ne 100 % tikslus, pavyzdžiui, tam tikri grafiniai elementai gali būti interpretuojami kaip simboliai, taip pat suklastotos prisijungimo formos gal naudoti šiek tiek skirtingą tekstą, teksto paieškai naudojamas teksto panašumo įvertinimas. Tam panaudotas kosinusinio panašumo (angl. *cosine similarity*) rodiklis. Eksperimentiniu būdu buvo pasirinkta, jog 80 % panašumas yra pakankamas, jog būtų rasta atitiktis, tuo pačiu metu neatpažįstant nesusijusių teksto fragmentų.

3.5. Naršyklės įskiepis

Kuriamoje sistemoje naršyklėje veikiantis įskiepis turi realizuoti tokį funkcionalumą:

- sekti vartotojo veiksmus naršyklėje, bei inicijuoti tolimesnius veiksmus kai vartotojas pradeda įvedinėti tekstą į slaptažodžio įvesties formą;
- padaryti tinklapio vaizdo kopiją;
- nustatyti įvesties lauko koordinates;
- perduoti surinktus duomenis kompiuteryje veikiančioje programoje;
- atpažinus suklastotą prisijungimo formą, parodyti perspėjimą vartotojui.

Vartotojo veiksmų sekimas atliekamas į kiekvieną lankomą tinklapį, įskaitant ir jame esančius „iframe“ objektus įterpiant „*Javascript*“ kodą. Jame realizuota vartotojo pelės paspaudimų sekimo logika. Užregistravus paspaudimą, patikrinama, ar paspausta buvo ant įvesties lauko, kurio tipas „password“. Jei taip, inicijuojami sekantys veiksmai.

Tinklapio vaizdo kopija gaunama naudojant *tabs.captureVisibleTab()* naršyklės API metodą, kuris gražina šiuo metu aktyvaus naršyklės lango vaizdą.

Įvesties lauko pozicija tinklapyje nustatoma naudojant *getBoundingClientRect()* metodą, gražinančią DOM (angl. *Document Object Model*) objekto poziciją bei dydį vartotojui rodomo tinklapio dalyje. Testavimo metu pastebėta, jog metodo gražinamos reikšmės nepriklauso nuo nustatyto tinklapio mastelio, tai yra, jei mastelis nėra lygus 100 %, gražinamos neteisingos reikšmės. Dėl to pozicija perskaičiuojama pagal tinklapio mastelį, gaunamą iš *window.devicePixelRatio* savybės. Taip pat pastebėta, jog įvesties laukui esant „iframe“ DOM elemente, gaunama pozicija reliatyvi šiame „iframe“ lange vaizduojamam tinklapiui, o ne

pagrindiniam langui. Dėl to pridėta papildoma logika, jog būtų nustatyta „iframe“ elemento pozicija pagrindiniame lange.

3.6. Apibendrinimas

Sistema realizuota naudojant „Python“ programavimo kalbą analizės programai bei „JavaScript“ naršyklės įskiepiui. Peržiūrėjus esamus logotipų duomenų rinkinius pastebėta, jog juose pateikti pasenę logotipai, kurie esamose prisijungimo formose nebenaudojami. Dėl to buvo sudarytas bei rankiniu būdu anotuotas naujas duomenų rinkinys, susidedantis iš 5 klasių, 152 paveikslų bei 272 anotacijų. Išmokius „Faster R-CNN“ algoritmą, gautas 0,96 mAP-50 tikslumo įvertis. Rankiniu būdu sudaryta duomenų bazė, sauganti prisijungimo formų požymius – URL adresą, logotipo klasę, teksto fragmentus, įvesties lauką bei šių elementų išsidėstymą.

4. „BitB“ atakos atpažinimo metodo tyrimas

4.1. Tyrimo aplinka

Sukurto metodo, kuris realizuotas prototipu tyrimui buvo panaudotas asmeninis kompiuteris, kurio parametrai pateikti 3 lentelėje.

3 lentelė. Tyrimui naudojamo kompiuterio parametrai

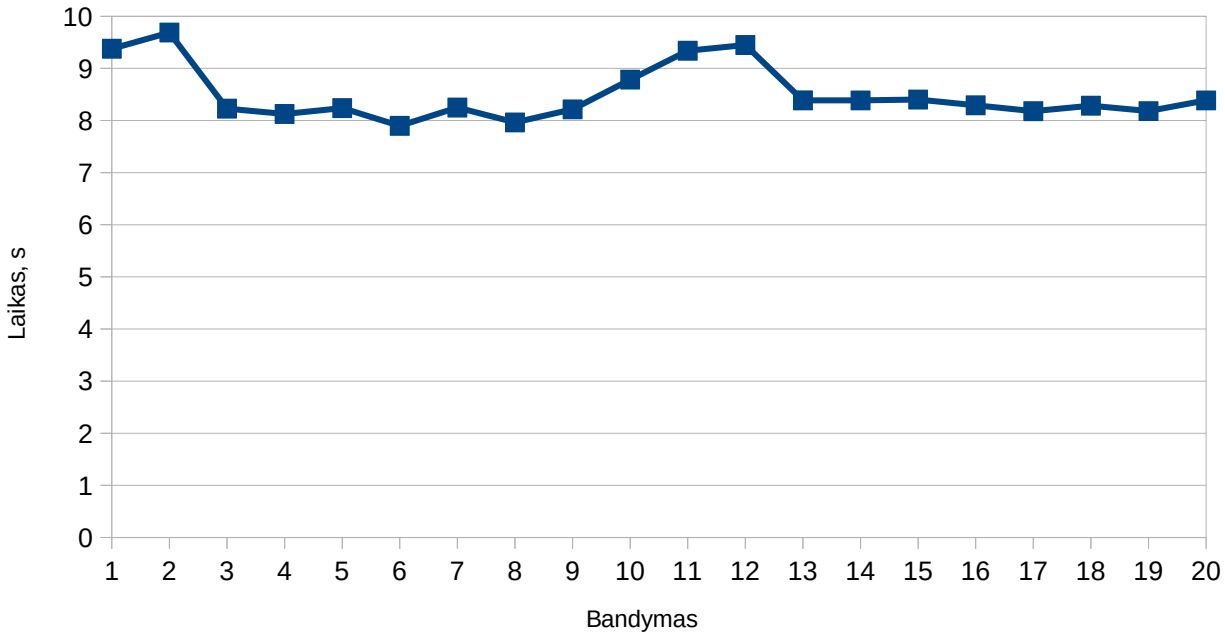
Parametras	**Reikšmė**
Operacinė sistema	Ubuntu 24.04.1 LTS
Centrinis procesorius	Intel(R) Core(TM) i5-4460 @ 3.20GHz, 4 branduolių
Grafinis procesorius	Nvidia GeForce GTX 980 Ti, 6 GB VRAM
Operatyvinė atmintis	16 GB DDR3
Ekrano raiška	1920x1080

4.2. Sistemos greitaveikos tyrimas

Sistemos greitaveikos tyrimas buvo atliekamas rankiniu būdu, lankantis „BitB“ ataką imituojančiuose tinklapiuose bei matuojant laiką tarp slaptažodžio įvedimo lauko aktyvavimo, iki rezultatų gavimo.

Kiekvieno testavimo atveju buvo atliekama po 20 matavimų, bei paskaičiuojamas vidutinis laikas, sugaištas tinklapio analizei. Taip pat užfiksuotas operatyvinės bei grafinės atminties naudojimas analizės metu bei fone, neatliekant jokių veiksmų.

21 paveiksle pateiktas grafikas, vaizduojantis tinklapio analizės laiką, logotipų atpažinimui naudojant „Faster R-CNN“ algoritmą, kai skaičiavimai vykdomi centrinio procesoriaus (CPU).



21 pav. Užklausų vykdymo laikas naudojant „Faster R-CNN“ veikiantį ant CPU

4 lentelėje pateiktos vidutinės analizės atlikimo reikšmės bei vidutinės operatyvinės bei grafikos procesoriaus atminties naudojimo reikšmės.

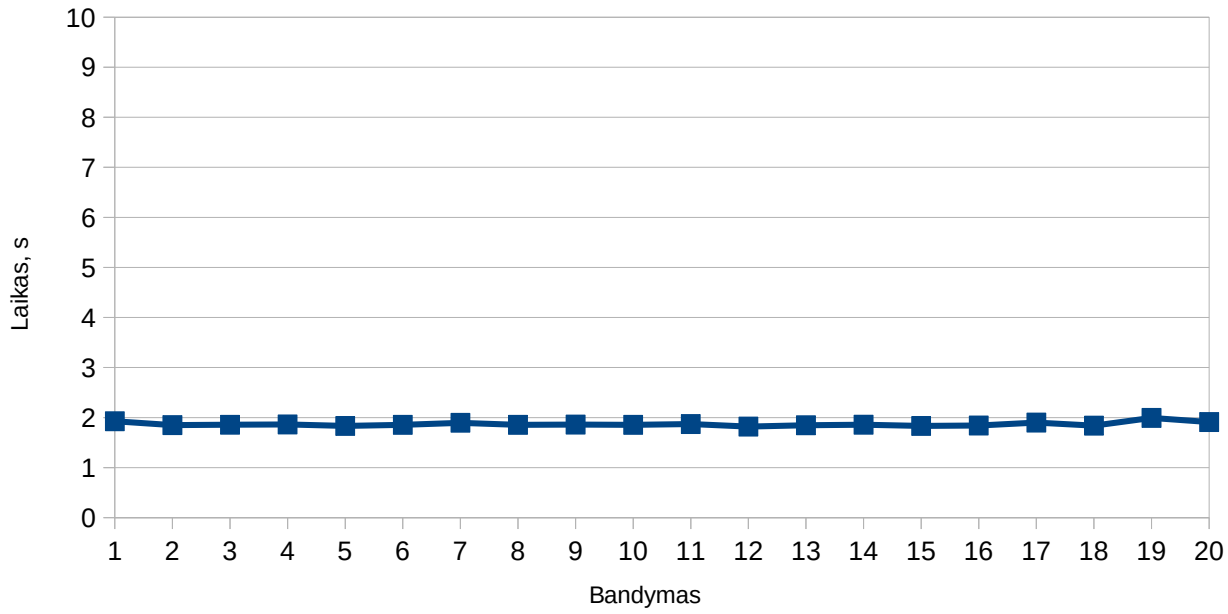
4 lentelė. Resursų naudojimas naudojant „Faster R-CNN“ algoritmą, naudojant centrinį procesorių

Vidutinis laikas, reikalingas atpažinimui	7,45 sekundės
Operatyvinės atminties naudojimas analizės metu	1598 MB
Operatyvinės atminties naudojimas fone	768 MB
Grafikos procesoriaus atminties naudojimas analizės metu	-
Grafikos procesoriaus atminties naudojimas fone	-

Gauti testavimo rezultatai rodo, jog vidutiniškai tinklapio analizė užtrunka daugiau nei 7 sekundes, o operatyvinės atminties naudojimas viršija 1,5 GB. Šios reikšmės neatitinka išsikeltų nefunkcinių sistemos reikalavimų. Sistema pastebimai sulėtina prisijungimo prie tinklapių procesą, dėl to vartotojai patiria nepatogumų. Pasigilinus į atskirų sistemos komponentų resursų naudojimą bei veikimo laiką pastebėtą, jog analizės proceso greitaveikai didžiausią įtaką daro logotipų atpažinimo posistemė.

Siekiant padidinti logotipų atpažinimo posistemės greitaveiką, yra galimybė paspartinti skaičiavimus naudojant grafinį akceleratorių. Šiuo metu naudojamas „Pytorch“ karkasas palaiko tris akceleratorių tipus: „CUDA“, „MTIA“ bei „XPU“ [35]. „CUDA“ akceleratorius yra palaikomas „Nvidia“ gamintojo grafikos procesorių, „XPU“ palaikomas „Intel“ grafikos procesorių, „MTIA“ palaikomas „Meta“ kompanijos sukurto akceleratoriaus. Tyrimui naudojamas kompiuteris turi „Nvidia“ gamintojo grafikos procesorių, kuris palaiko „CUDA“ akceleraciją. Sistema buvo sukonfigūruota, jog akceleracija būtų naudojama, bei atliktas pakartotinis testavimas.

22 pav. pateiktas grafikas, vaizduojantis tinklapio analizės laiką, logotipų atpažinimui naudojant „Faster R-CNN“ algoritmą logotipų įjungus „CUDA“ akceleratorių.



22 pav. Užklausų vykdymo laikas naudojant „Faster R-CNN“ su „CUDA“ akceleratoriumi

5 lentelėje pateiktos vidutinės analizės atlikimo reikšmės bei vidutinės operatyvinės bei grafikos procesoriaus atminties naudojimo reikšmės.

5 lentelė. Resursų naudojimas naudojant „Faster R-CNN“ algoritmą, naudojant grafinį procesorių

Vidutinis laikas, reikalingas atpažinimui	1,73 sekundės
Operatyvinės atminties naudojimas analizės metu	1598 MB
Operatyvinės atminties naudojimas fone	813 MB
Grafikos procesoriaus atminties naudojimas analizės metu	1544 MB
Grafikos procesoriaus atminties naudojimas fone	1544 MB

Gauti rezultatai rodo, jog vidutinis analizės laikas naudojant „CUDA“ akceleratorių sumažėjo daugiau nei 4 kartus, ir atitinka išsikeltą nefunkcinį reikalavimą. Nepaisant to, šiek tiek padidėjo operatyvinės atminties naudojimas, kuris ir taip neatitiko išsikeltų reikalavimų, bei smarkiai išaugo grafikos procesoriaus atminties naudojimas. Sistema nuolat naudoja ~1,5 GB grafikos procesoriaus atminties – tai yra apie 25 % visos testavimui naudojamame kompiuteryje esančios atminties.

Kita problema su akceleratoriais yra ta, jog jie prieinami tik dalyje sistemų. Jeigu sistemoje nėra prieinamas nei vienas palaikomas akceleratorius, analizės laikas išauga tiek, jog pradeda sudaryti nepatogumų sistemos naudotojui. Įvertinus šias problemas, buvo nuspręsta rinktis kitą algoritmą logotipų atpažinimui.

4.3. Logotipų atpažinimo algoritmo keitimas

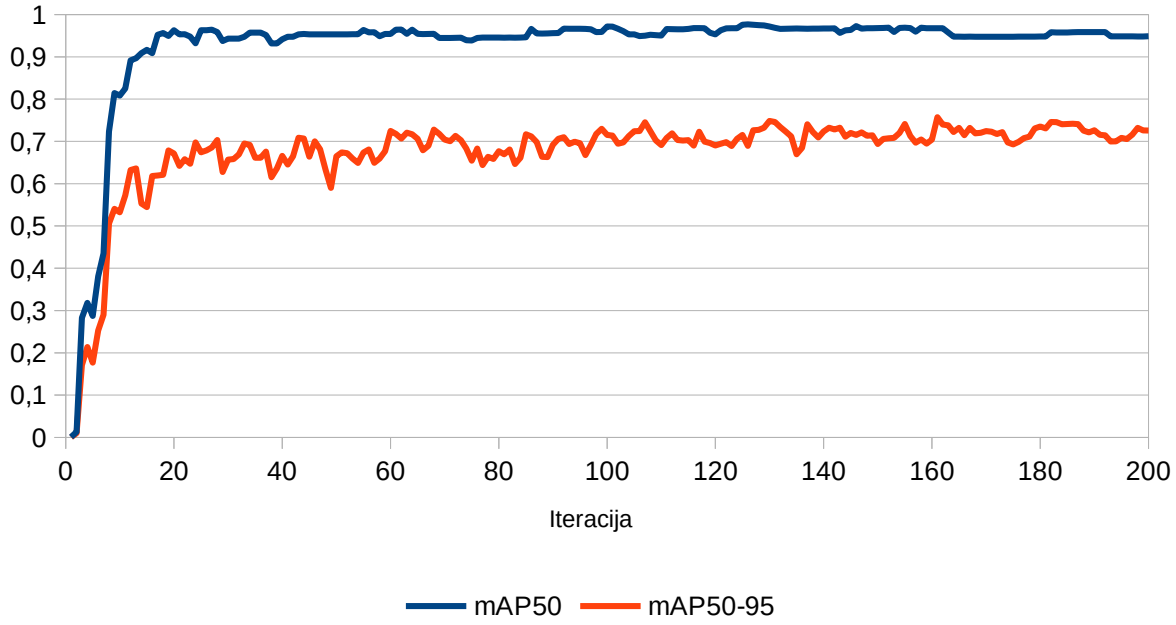
Siekiant išspręsti sistemos greitaveikos problemas, buvo nuspręsta keisti logotipų atpažinimo posistemėje naudojamą algoritmą į „YOLO v8“ kadangi iš analizės dalyje nagrinėtų algoritmų jis suteikia geriausią tikslumo/greitaveikos santykį.

„YOLO v8“ buvo išmokytas naudojant tą patį sudarytą duomenų rinkinį, kuris buvo naudojamas „Faster R-CNN“ algoritmo mokymui.

Kadangi „YOLO v8“ mokymo trukmė gerokai trumpesnė nei „Faster R-CNN“, buvo nuspręsta pasinaudoti programa, gebančia automatiškai rasti optimalius modelio hiperparametrus. Buvo bandoma optimizuoti tiek įprastus mašininio mokymosi algoritmų hiperparametrus – apsimokymo greitį (angl. *learning rate*), partijos dydį (angl. *batch size*), tiek paveikslų transformacijų parametrus – pasukimo, šviesumo, spalvų gamos ir kt. Optimizuojant hiperparametrus, modelis buvo mokomas 200 kartų, kiekvieną kartą po 30 iteracijų.

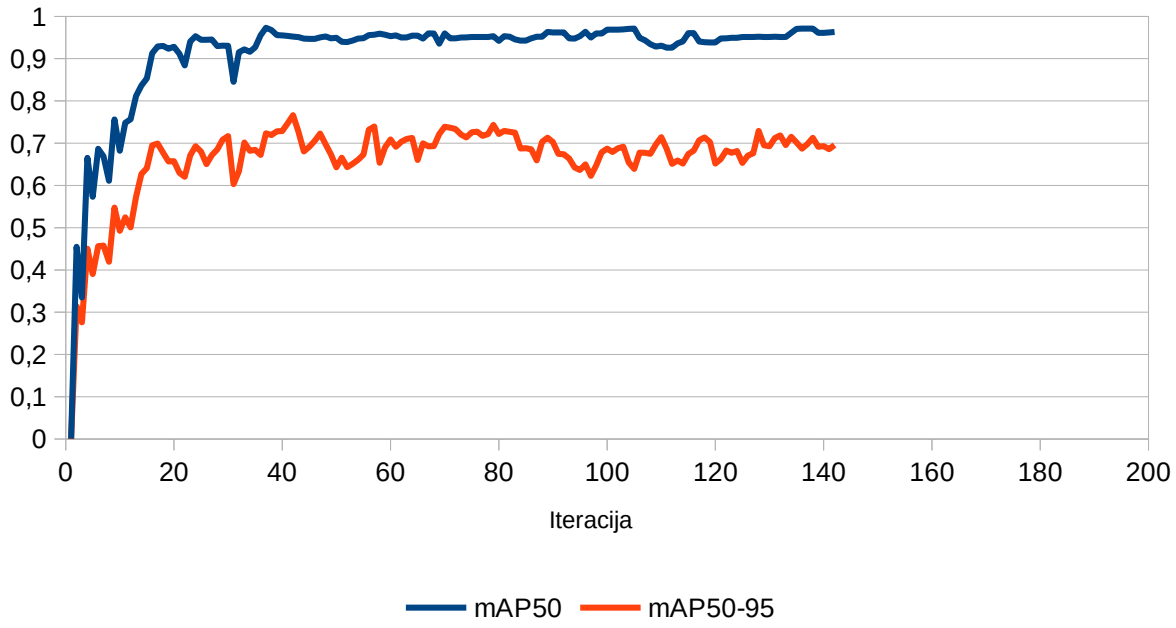
Gavus optimalių hiperparametrų reikšmes, buvo išmokyti trys „YOLO v8“ modelio variantai – „yolov8n“, „yolov8s“ ir „yolov8m“. 23-25 paveiksluose pateikti grafikai, vaizduojantys visų modelio variantų mAP įverčių priklausomybę nuo mokymo iteracijos. „yolov8s“ modelis buvo mokomas tik 140 iteracijų, kadangi mokymas buvo automatiškai nutrauktas, nepasiekus geresnių rezultatų per paskutines 100 mokymo iteracijų.

„yolov8n“ modelis geriausią mAP50 įvertį pasiekė 127 iteracijos metu (žr. 23 pav.).



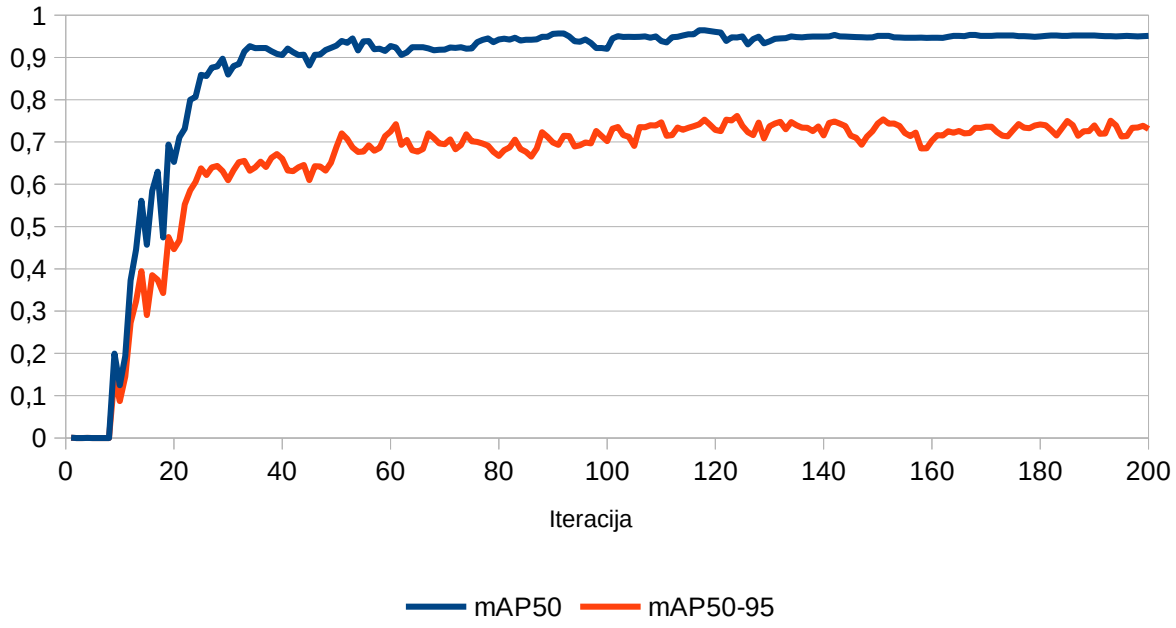
23 pav. „yolov8n“ modelio mAP įvertis kiekvienos iteracijos metu

„yolov8s“ modelis geriausią mAP50 įvertį pasiekė 38 iteracijos metu (žr. 24 pav.).



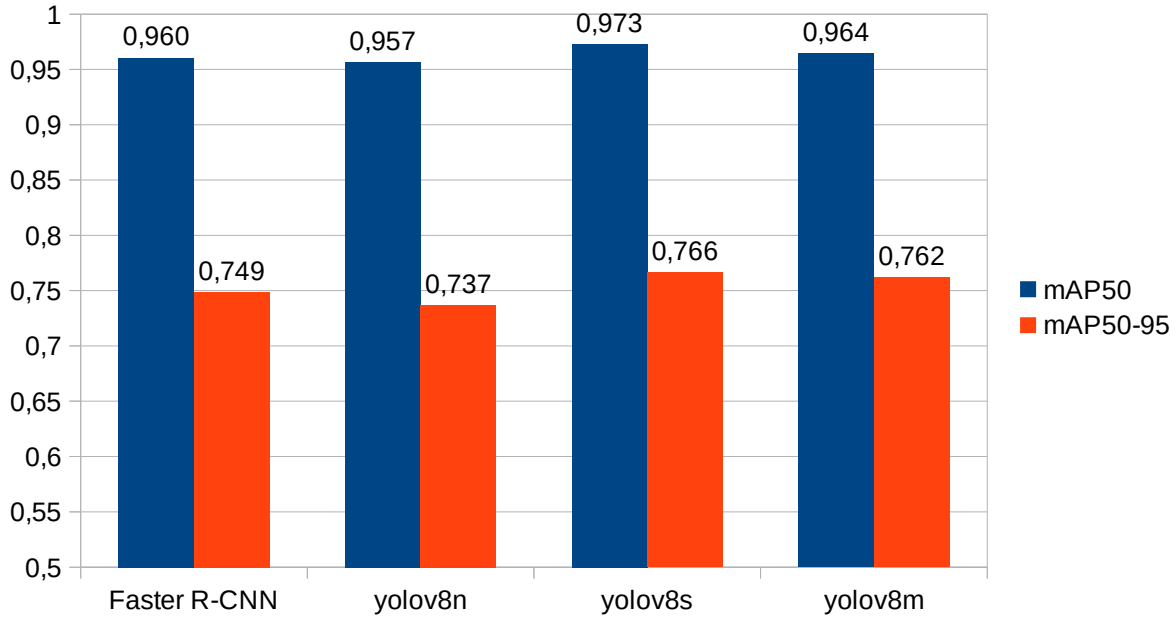
24 pav. „yolov8s“ modelio mAP įvertis kiekvienos iteracijos metu

„yolov8m“ modelis geriausią mAP50 įvertį pasiekė 118 iteracijos metu (žr. 25 pav.).



25 pav. „yolov8m“ modelio mAP įvertis kiekvienos iteracijos metu

Gauti rezultatai rodo, jog visų „YOLO v8“ modelio variantų tikslumas yra panašus. 26 paveiksle pateiktas apibendrintas visų modelių tikslumo įverčių grafikas.



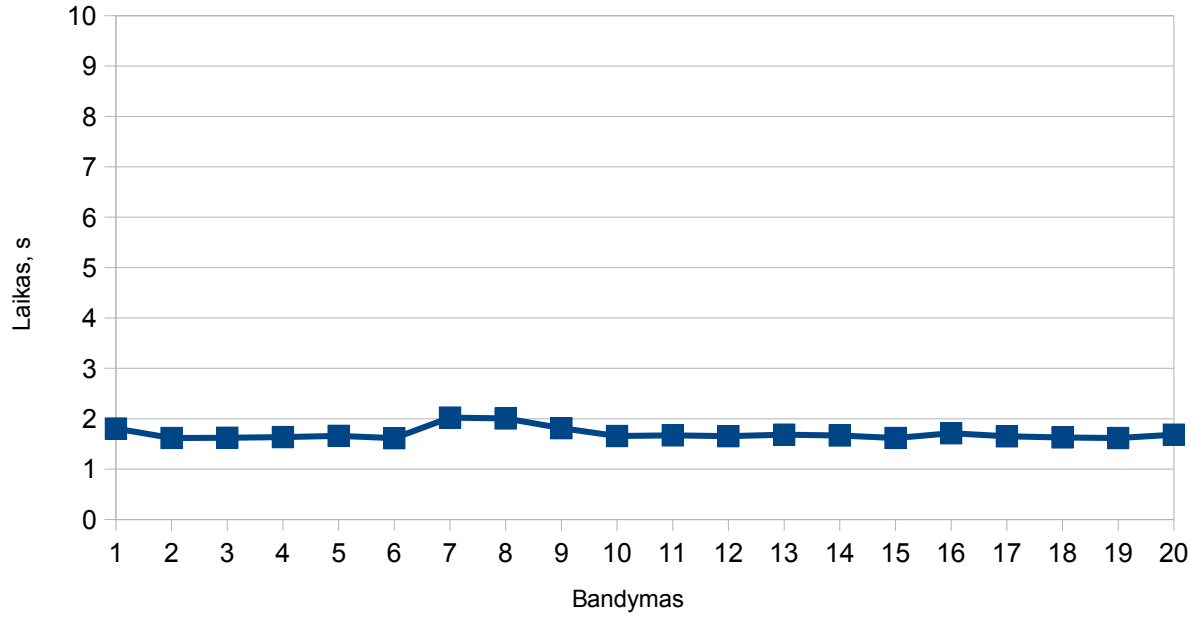
26 pav. Visų logotipų atpažinimo modelių mAP įverčių grafikas

„yolov8s“ bei „yolov8m“ tikslumo įverčiai viršijo „Faster R-CNN“ įverčius, „yolov8n“ nežymiai atsiliko. Nors „yolov8m“ modelis turi didesnį skaičių parametrų bei teoriškai turėtų pasiekti geresnį tikslumą, gautas tikslumas mažesnis lyginant su „yolov8s“. Tą galėjo lemti naudota mažesnė „batch“ hiperparametro vertė. Mažesnė vertė buvo naudota dėl ribotų techninių galimybių – nepakankamo grafikos procesoriaus atminties kiekio. Dėl geriausių parodytų tikslumo įverčių, buvo nuspręsta naudoti „yolov8s“ paremtą modelį.

4.4. Greitaveikos tyrimas logotipų atpažinimui naudojant „YOLO“ algoritmą

Tyrimas buvo pakartotas analogiškomis sąlygomis kaip 4.2 skyrelyje.

27 paveiksle pateiktas grafikas, vaizduojantis tinklapio analizės laiką, logotipų atpažinimui naudojant „YOLO“ algoritmą, kai skaičiavimai vykdomi centrinio procesoriaus (CPU).



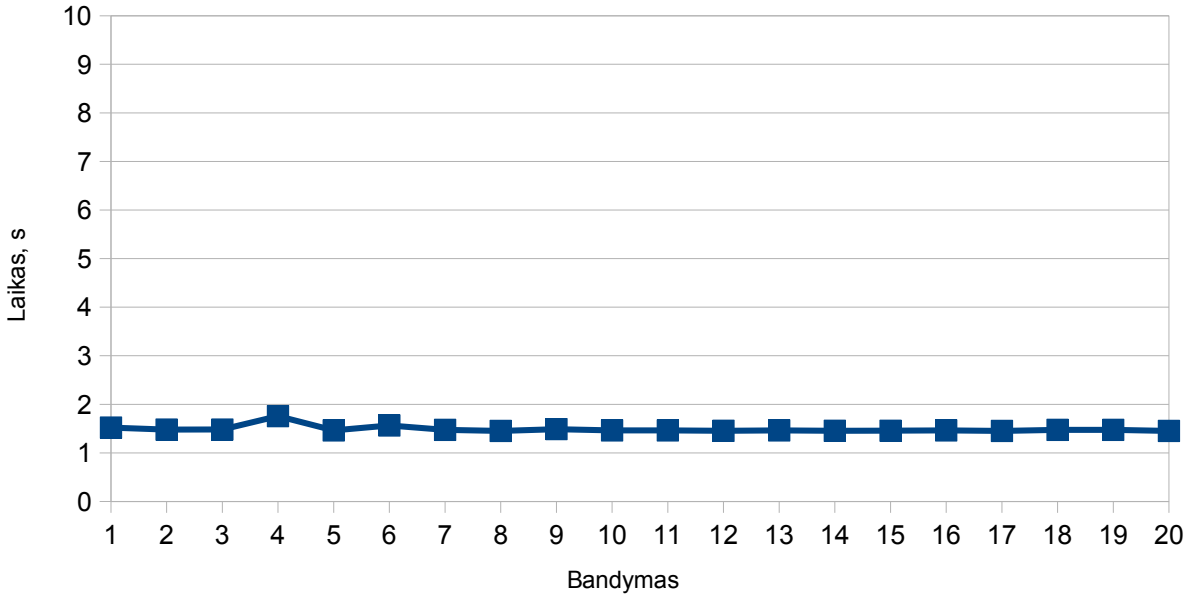
27 pav. Užklausų vykdymo laikas naudojant „YOLO“ veikiantį ant CPU

6 lentelėje pateiktos vidutinės analizės atlikimo reikšmės bei vidutinės operatyvinės bei grafikos procesoriaus atminties naudojimo reikšmės.

6 lentelė. Resursų naudojimas naudojant „YOLO“ algoritmą, naudojant centrinį procesorių

Vidutinis laikas, reikalingas atpažinimui	1,75 sekundės
Operatyvinės atminties naudojimas analizės metu	1061 MB
Operatyvinės atminties naudojimas fone	552 MB
Grafikos procesoriaus atminties naudojimas analizės metu	0 MB

28 paveiksle pateiktas grafikas, vaizduojantis tinklapio analizės laiką, logotipų atpažinimui naudojant „YOLO“ algoritmą, kai skaičiavimai vykdomi grafinio procesoriaus (GPU).



28 pav. Užklausų vykdymo laikas naudojant „YOLO“ veikiantį ant GPU

7 lentelėje pateiktos vidutinės analizės atlikimo reikšmės bei vidutinės operatyvinės bei grafikos procesoriaus atminties naudojimo reikšmės.

7 lentelė. Resursų naudojimas naudojant „YOLO“ algoritmą, naudojant grafinį procesorių

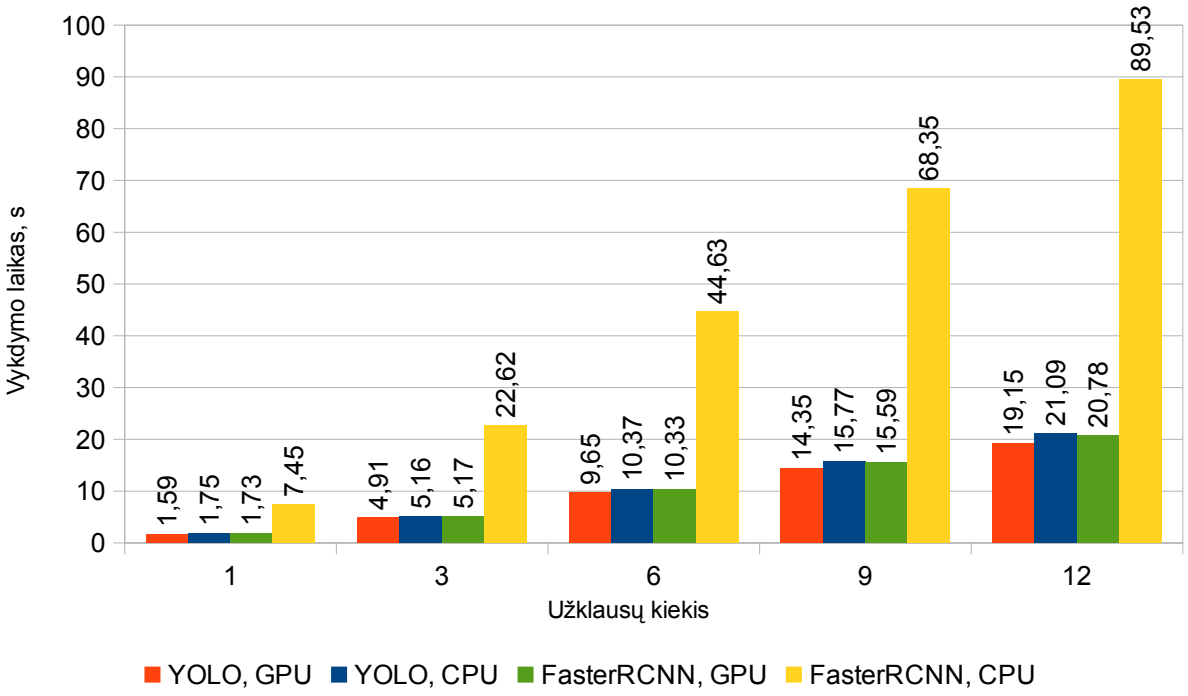
Vidutinis laikas, reikalingas atpažinimui	1,59 sekundės
Operatyvinės atminties naudojimas analizės metu	1427 MB
Operatyvinės atminties naudojimas fone	708 MB
Grafikos procesoriaus atminties naudojimas analizės metu	390 MB
Grafikos procesoriaus atminties naudojimas fone	390 MB

Rezultatai rodo, jog vidutinis laikas, reikalingas atpažinimui tenkina reikalavimuose išsikeltą 3 sekundžių ribą abiem atvejais – tiek logotipų atpažinimo posistemei skaičiavimus atliekant centriniame procesoriuje, tiek naudojant grafinį akceleratorių. Operatyvinės bei grafikos procesoriaus atminties naudojimas taip pat tenkina išsikeltus reikalavimus – atitinkamai 1,5 GB bei 512 MB. Naudojant akceleratorių, vidutiniškai atpažinimo laikas sutrumpėja tik 0,16 sekundės, tačiau operatyvinės bei grafikos procesoriaus naudojimas padidėja atitinkamai ~400 MB bei ~700 MB. Dėl to šiuo atveju grafinio akceleratoriaus naudojimas turi daugiau trūkumų nei suteikia naudos.

4.5. Greitaveikos tyrimas esant kelioms užklausoms vienu metu

Kadangi suprojektuota sistema bei jos prototipas veikia vartotojo įrenginyje, bei aptarnauja tik vieną vartotoją, vienu metu dažniausiai yra poreikis apdoroti tik vieną užklausą. Tačiau gali pasitaikyti atvejų, kai nebaigus analizuoti vieno tinklapio, atidaromi nauji tinklapiai kuriuos reikia išanalizuoti. Dėl to buvo atliktas greitaveikos tyrimas, siekiant išsiaiškinti, kokia yra analizės laiko priklausomybė nuo užklausų kiekio. 29 pav. pateikti tyrimo rezultatai, kuriuose pateikiamos laiko

reikšmės naudojant skirtingus algoritmus logotipų atpažinimui, naudojant ir nenaudojant akceleratorių.



29 pav. Analizės vykdymo laiko priklausomybė nuo užklausų kiekio

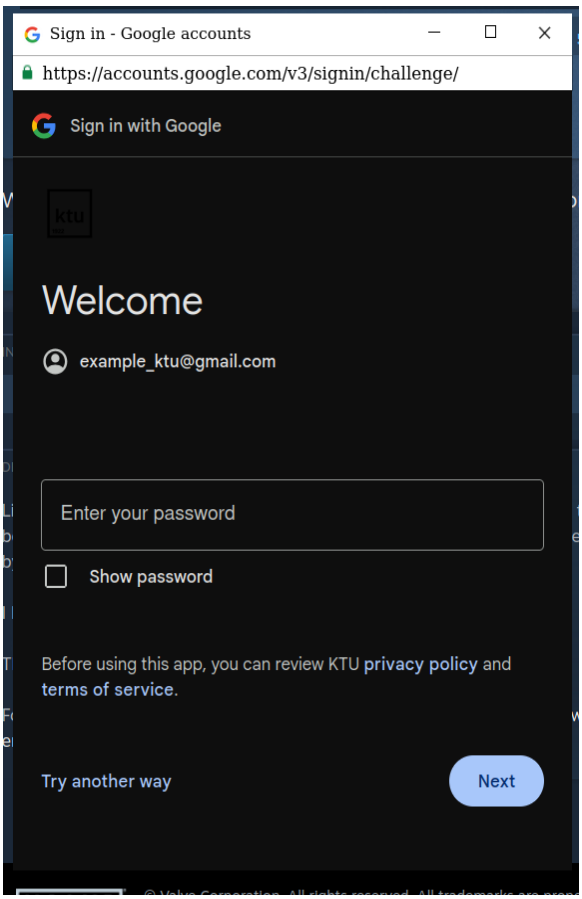
Rezultatai rodo, jog sulaukus 3 užklausų vienu metu, analizės laikas pailgėja iki ~5 sekundžių, ir sistema neatitinka reikalavimuose išsikeltos 3 sekundžių ribos. Tačiau tokios sąlygos turėtų pasitaikyti tik išskirtiniais atvejais, todėl tokį sistemos veikimą galima laikyti priimtinu.

4.6. Pasiruošimas sistemos kokybiniam tyrimui atlikti

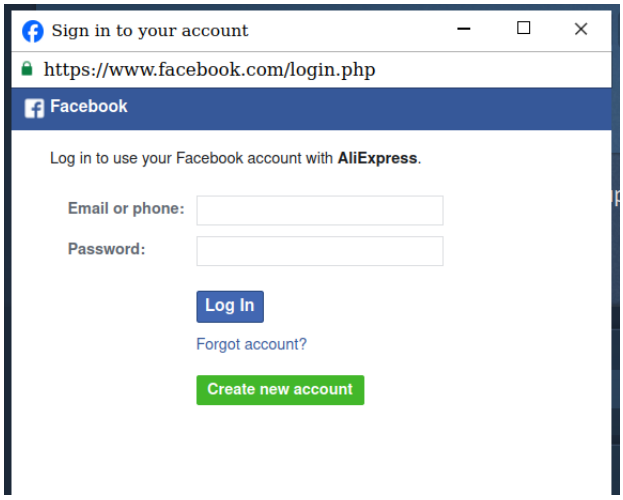
Siekiant tinkamai įvertinti sukurtos sistemos veikimo tikslumą, reikalinga prieiga prie realių tinklapių, kurie naudojami „BitB“ atakų vykdymui. Tokių tinklapių paieškai buvo pasinaudota „Phishtank“ sistema, kuriame vartotojai gali pranešti apie sukčiavimui naudojamus tinklapius, bei peržiūrėti kitų vartotojų pranešimus. Išfiltravus duomenis pagal aktualius prekės ženklus, buvo rasta keletas „BitB“ atakos pavyzdžių. Tačiau buvo susidurta su problema, jog šių tinklapių gyvavimo laikas yra trumpas, kas apsunkino galimybes tinkamai atlikti tyrimą. Kita problema – rasti pavyzdžiai imituoja tik „Steam“ bei „Google“ prisijungimo formas, kitų nagrinėjamų prisijungimo formų pavyzdžių rasti nepavyko.

Dėl šios priežasties buvo nuspręsta sukurti kontroliuojamą tyrimo aplinką, kuri nepriklausytų nuo išorinių veiksnių. Šiam tikslui pasiekti buvo išsisaugotas realus „BitB“ atakai naudojamas tinklapis, atkurtas užtemdytas išeities kodas, jog jį būtų galima modifikuoti pagal poreikius. Naudojant šį pavyzdį, sukurtos 5 kopijos, kuriose netikrame iššokančiame lange vaizduojamos skirtingos, tyrime nagrinėjamos prisijungimo formos – „Google“, „Facebook“, „Microsoft“, „Paypal“ bei „Steam“.

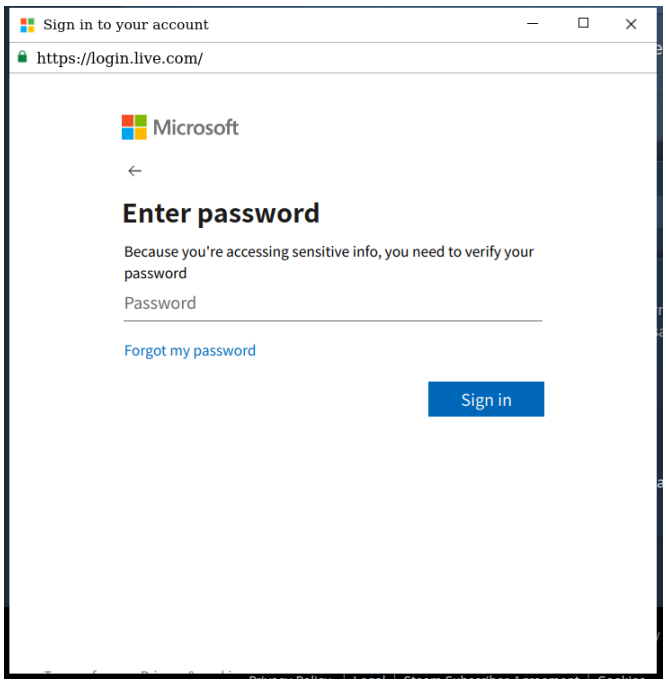
30-34 pav. pateikti sudarytų netikrų iššokančiųjų langų vaizdai.



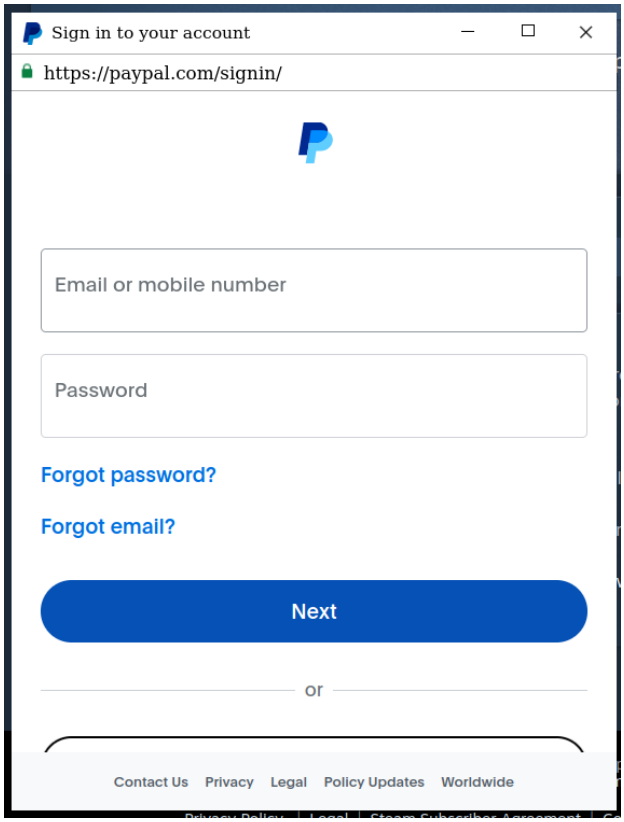
30 pav. Sugeneruotas „Google“ prisijungimo langas



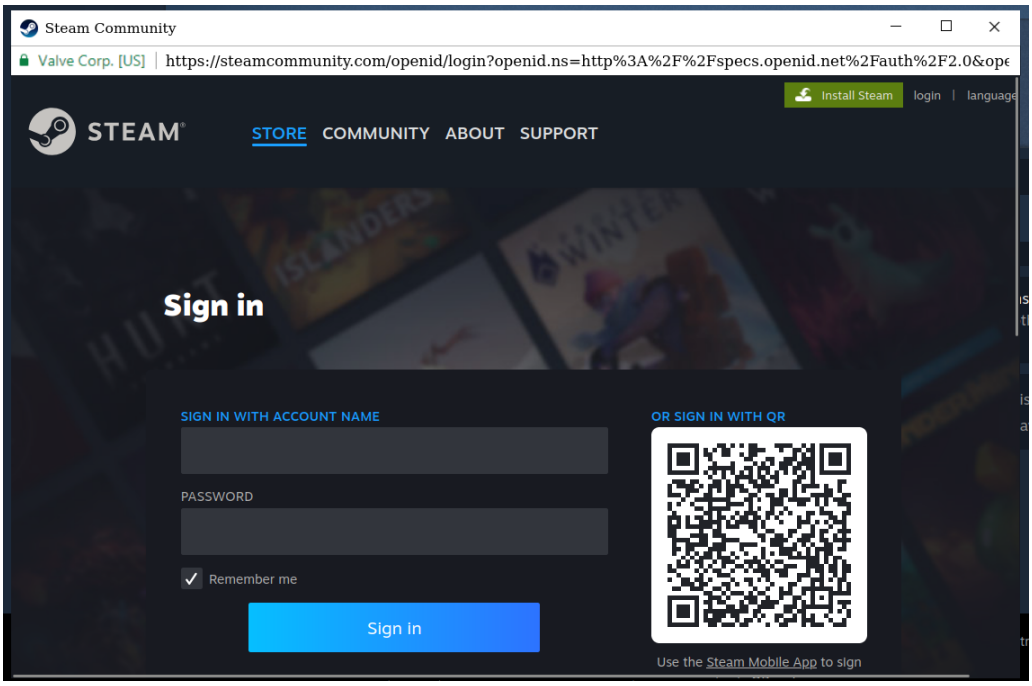
31 pav. Sugeneruotas „Facebook“ prisijungimo langas



32 pav. Sugeneruotas „Microsoft“ prisijungimo langas



33 pav. Sugeneruotas „Paypal“ prisijungimo langas



34 pav. Sugeneruotas „Steam“ prisijungimo langas

4.7. Sistemos kokybinis tyrimas

Sistemos prototipas buvo testuojamas rankiniu būdu, išbandant atsitiktiniuose tinklapiuose, kuriuose nėra imituojamos prisijungimo formos, bei kenkėjiškuose tinklapiuose, kuriuose naudojama „BitB“ tipo ataka.

Atliekant eksperimentus su 4.6 skyrelyje paruoštais „BitB“ atakos pavyzdžiais buvo pastebėta, jog logotipų bei teksto atpažinimo tikslumas priklauso nuo kelių faktorių. Pirmas faktorius – netikro naršyklės lango pozicija ekrane. Logotipų atpažinimo užtikrintumo (angl. *confidence*) reikšmė keičiasi nuo aplink esančių objektų, nors pats logotipas išlieka identiškas. Tuo pat metu, teksto atpažinimo tikslumas taip pat skiriasi nuo elementų išsidėstymo ekrane. Dėl šių priežasčių kiekvienos prisijungimo formos atpažinimo tikslumas buvo tikrintas 3 kartus, netikro naršyklės lango poziciją keičiant į skirtingas, iš anksto apibrėžtas vietas. 8 lentelėje pateikti skirtingi lango pozicijų variantai.

8 lentelė. Netikro lango pozicijų variantai

Pozicija	**Y pozicija, px**	**X pozicija, px**
1	100	100
2	150	500
3	200	1000

Kitas faktorius, nuo kurio priklauso teksto bei logotipo atpažinimo tikslumas yra spalvų schema. Populiariausios interneto naršyklės leidžia pasirinkti išvaizdos temą – šviesią arba tamsią. Nuo šios parinkties skiriasi ne tik pačios naršyklės grafinių elementų išvaizda, bet gali skirtis iš pačių tinklapių turinio išvaizdą, jei tai buvo numatyta tinklapio kūrėjo. Taip pat plačiai naudojami naršyklės įskiepiai, kurie gali keisti tinklapio spalvų schemą nepaisant to, ar tinklapio kūrėjai yra

aprašę skirtingas spalvų schemas. Kadangi darant eksperimentus buvo pastebėta, jog spalvų schema daro įtaką tiek logotipų atpažinimo tikslumui, tiek teksto atpažinimo tikslumui, testavimas buvo atliktas tiek su tamsia, tiek su šviesia spalvų schemomis. Iš viso, sudėjus skirtingas lango pozicijų bei spalvų schemų kombinacijas, buvo gauta bei ištestuota 30 variantų.

Siekiant ištestuoti, ar metodas neidentifikuoja įprastų prisijungimo formų kaip „BitB“ atakos, buvo išbandyti 43 atsitiktiniai populiarūs tinklapiai, turintys prisijungimo formas. Tinklapiai buvo pasirinkti pagal „DuckDuckGo“ paieškos variklio rezultatus panaudojus užklausą „login“. Tinklapių sąrašas yra pateiktas 1 priede.

Metodo veikimo įvertinimui naudojami įprasti kokybiniai rodikliai – tikslumas (angl. *accuracy*), preciziškumas (angl. *precision*), jautrumas (ang. *recall*) (žr. 1-3 formules).

$$Tikslumas = \frac{TP + TN}{TP + TN + FP + FN}; \tag{1}$$

$$Preciziškumas = \frac{TP}{TP + FP}; \tag{2}$$

$$Jautrumas = \frac{TP}{TP + FN}; \tag{3}$$

čia TP (angl. *true positive*) reiškia teisingai teigiami, TN (angl. *true negative*) – teisingai neigiami, FP (angl. *false positive*) – neteisingai teigiami, o FN (angl. *false negative*) – neteisingai neigiami atpažinimo rezultatai.

Testavimo rezultatai pateikti 9 lentelėje.

9 lentelė. Testavimo rezultatai

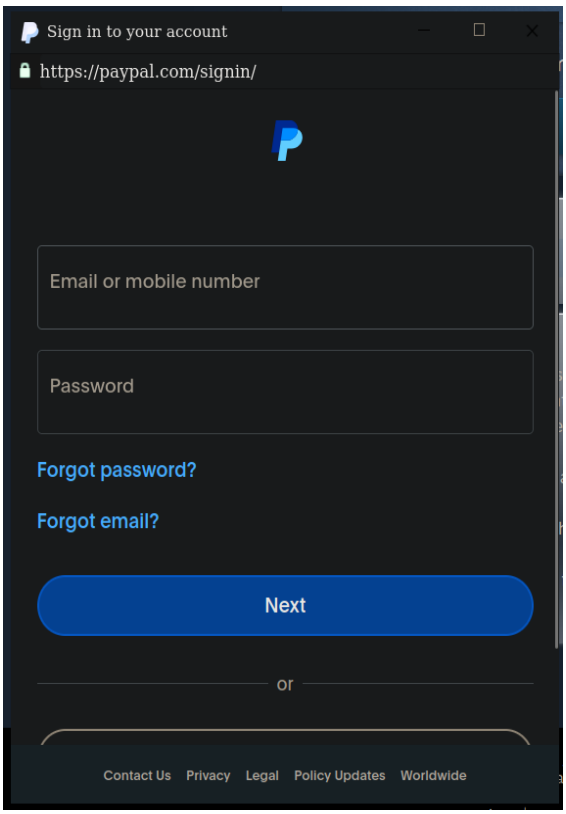
Teisingai teigiami (TP)	**Teisingai neigiami (TN)**	**Neteisingai teigiami (FP)**	**Neteisingai neigiami (FN)**
28	43	0	2

Iš šių rezultatų paskaičiuoti kokybiniai rodikliai pateikti 10 lentelėje.

10 lentelė. Kokybiniai metodo rodikliai

Tikslumas	**Preciziškumas**	**Jautrumas**
0,959	0,966	0,933

Rezultatai rodo, jog iš 73 bandymų, 3 buvo identifikuoti neteisingai. Du iš šių atvejų yra neteisingai neigiami atvejai, kai naudojant tamsią tinklapio temą, dvejose iš trijų lango pozicijų nebuvo identifikuotas „Paypal“ prisijungimo formoje esantis logotipas. 35 pav. pateiktas šios prisijungimo formos vaizdas.



35 pav. Netikra „Paypal“ prisijungimo forma naudojant tamsią temą

Peržiūrėjus duomenų rinkinį buvo pastebėta, jog jame buvo tik viena nuotrauka, kurioje „Paypal“ logotipas patektas tamsiame fone. Galima numanyti, jog dėl šios priežasties logotipų atpažinimo modelis nepakankamai gerai išmoko atpažinti logotipą tokiomis sąlygomis. Siekiant pagerinti logotipų atpažinimo tikslumą, reikalingas didesnis duomenų rinkinys, kuriame būtų naudojami įvairesni foniniai vaizdai. Kadangi rankiniu būdu tokį duomenų rinkinį sudaryti yra sudėtinga, bei procesas užima daug laiko, optimaliausia būtų panaudoti įrankius, skirtus sintetinių duomenų rinkinių generavimui.

Rezultatai rodo, jog vizualiai panašias prisijungimo formas sistema geba identifikuoti daugeliu atvejų. Vizualiai nepanašios formos, kuriuose nenaudojami nei logotipai, nei tekstas, naudojami originalioje formoje, nėra identifikuojami.

Iš bandytų tinklapių, kuriuose naudojamos prisijungimo formos nėra suklastotos, nei viename nebuvo klaidingai identifikuota prisijungimo forma.

4.8. Tyrimo išvados

1. Atlikus pirminį prototipo greitaveikos tyrimą buvo nustatyta, jog sistema neatitinka užsibrėžto reikalavimo, jog analizės laikas neviršytų 3 sekundžių. Nustačius, jog logotipų atpažinimo procesas trunka ilgiausiai, buvo nuspręsta pakeisti logotipų atpažinimui naudojamą algoritmą į efektyvesnį.
2. Kaip alternatyva „Faster R-CNN“ algoritmui, buvo pasirinktas „YOLO v8“. Palyginus skirtingus modelio variantus, kaip optimaliausias pasirinktas „yolov8s“. Palyginus tikslumo įverčius, gautas nežymiai geresnis tikslumas naudojant „YOLO v8“, bei gerokai didesnė

greitaveika. Gautas vidutinis atpažinimo laikas tenkina užsibrėžtus reikalavimus – siekia apie 1,75 sekundės.

3. Peržiūrėjus esamas sukčiavimo tinklapių duomenų bazes buvo rasta nedaug „BitB“ atakos pavyzdžių. Dėl to buvo sukurti 5 nauji pavyzdžiai, remiantis realiu „BitB“ ataką naudojančiu tinklapiu. Testavimas atliktas naudojant sukurtus „BitB“ atakos pavyzdžius bei testuojant atsitiktinius tinklapius. Atlikus testavimą, gautas 0,959 tikslumo įvertis. Nustatyta, jog sistemos prototipas ne visada tinkamai atpažįsta tinklapio vaizde esančius logotipus specifinėmis sąlygomis. Siekiant padidinti logotipų atpažinimo tikslumą, tuo pačiu ir visos sistemos tikslumą, reikalingas kokybiškesnis duomenų rinkinys logotipų atpažinimo algoritmo mokymui.

Išvados

1. Išanalizavus „BitB“ atakos požymius buvo nustatyta, jog ši ataka dažniausiai naudojama imituoti prisijungimo formoms, naudojamoms vieno prisijungimo paslaugų teikėjų. Esami atakos prevencijos metodai, tokie kaip tinklapių blokavimas naudojant juoduosius sąrašus nėra pakankamai efektyvus, kadangi blokavimas yra uždelstas, dėl to reikalingas metodas, gebantis ataką atpažinti realiu laiku. Tinkamiausias būdas „BitB“ atakai atpažinti pasirinktas palyginimas pagal vizualius požymius, dėl didesnio atsparumo piktavalių naudojamiems metodams, kuriais siekiama išvengti atpažinimo.
2. Pasiūlytas atakos atpažinimo metodas remiasi netikros prisijungimo formos identifikavimu pagal vizualius požymius. Metodas remiasi tuo, jog netikros prisijungimo formos vizualiai yra panašios į originalias prisijungimo formas, jog vartotojui nekiltų įtarimų jog tinklapis yra suklastotas. Siūloma sistema veikia vartotojo įrenginyje bei nereikalauja papildomų vartotojo veiksmų, jog būtų inicijuotas galimos atakos atpažinimas, bei aptikus galimą ataką, blokuoja tinklapio siunčiamas užklausas, jog įvesti prisijungimo duomenys nebūtų išsiųsti.
3. Realizuotas sistemos prototipas, susidedantis iš dviejų dalių – naršyklės įskiepio bei atskiros, vartotojo įrenginyje veikiančios programos. Peržiūrėjus esamus logotipų duomenų rinkinius pastebėta, jog juose pateikti pasenę, esamose prisijungimo formose nebenaudojami logotipai, dėl to buvo sudarytas bei rankiniu būdu anotuotas naujas duomenų rinkinys. Rankiniu būdu sudaryta duomenų bazė, sauganti prisijungimo formų požymius – URL adresą, logotipo klasę, teksto fragmentus, įvesties lauką bei šių elementų išsidėstymą.
4. Atliktas pirminis sistemos greitaveikos tyrimas parodė, jog tinklapio analizės laikas yra didesnis nei tikėtasi, bei netenkina išsikeltų nefunkcinių reikalavimų. Išsiaiškinus, jog didžiąją dalį resursų išnaudoja logotipų atpažinimo posistemė, buvo nuspręsta pakeisti naudojamą algoritmą į „YOLO v8“. Išmokius naują algoritmą bei pakartojus greitaveikos tyrimą nustatyta, jog bendras analizės laikas sutrumpėjo daugiau nei 4 kartus, o logotipų atpažinimo tikslumas nežymiai pagerėjo.
5. Atlikus kokybinį testavimą nustatyta, jog sistema geba atpažinti „BitB“ atakas 95,9 % tikslumu. Kadangi atliktas testavimas yra pakankamai nedidelės apimties, galima numanyti, jog naudojant sistemą realiomis sąlygomis atpažinimo tikslumas gali skirtis. Peržiūrėjus atvejus, kai „BitB“ ataka nebuvo teisingai atpažinta, nustatyta, jog pagrindinė problema yra neatpažinti logotipai esant specifinėms sąlygoms. Siekiant pagerinti logotipų atpažinimo tikslumą, reikalingas geresnis duomenų rinkinys, kurio sudarymui optimaliausia būtų naudoti automatizuotus įrankius.

Literatūros sąrašas

1. *Secure Blink: Steam account stolen involving new Browser-in-the-Browser attacks* [interaktyvus]. 2022 [žiūrėta 202-12-27]. Prieiga per: https://www.secureblink.com/cyber-security-news/steam-account-stolen-involving-new-browser-in-the-browser-attacks
2. BLANCAFLOR, Eric B., et al. Advanced Phishing Techniques: Analyzing Adversary-in-the-Middle and Browser-in-the-Browser Attacks in Modern Cybersecurity. *Cybernetics and Information Technologies*, 2025, 25.1. Prieiga per: doi: https://doi.org/10.2478/cait-2025-0004
3. OEST, Adam, et al. Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In: *29th {USENIX} Security Symposium ({USENIX} Security 20)*. 2020. Prieiga per: https://www.researchgate.net/publication/342782170_Sunrise_to_Sunset_Analyzing_the_End-to-end_Life_Cycle_and_E_ectiveness_of_Phishing_Attacks_at_Scale
4. CALLEGATI, Franco; CERRONI, Walter; RAMILLI, Marco. Man-in-the-Middle Attack to the HTTPS Protocol. *IEEE Security & Privacy*, 2009, 7.1: 78-81. Prieiga per: doi: https://doi.org/10.1109/MSP.2009.12
5. AKWUKWUMA, Veronica VN; EGWALI, Annie O. E-commerce: online attacks and protective mechanisms. *Asian Journal of Information Technology*, 2008, 7.9: 394-402. Prieiga per: https://www.makhillpublications.co/view-article/1682-3915/ajit.2008.394.402
6. UTAKRIT, Nattakant. Review of browser extensions, a man-in-the-browser phishing techniques targeting bank customers. 2009. Prieiga per: https://www.researchgate.net/publication/49279643_Review_of_Browser_Extensions_a_Man-in-the-Browser_Phishing_Techniques_Targeting_Bank_Customers
7. RAUTI, Sampsa. Man-in-the-browser attack: a case study on malicious browser extensions. In: *International Symposium on Security in Computing and Communication*. Springer, Singapore, 2020. p. 60-71. Prieiga per: doi: https://doi.org/10.1007/978-981-15-4825-3_5
8. *Browser In The Browser (BITB) Attack* [interaktyvus]. 2022 [žiūrėta 2022-12-28]. Prieiga per: https://mrd0x.com/browser-in-the-browser-phishing-attack
9. *Group-IB: Letting off steam* [interaktyvus]. 2022 [žiūrėta 2022-12-27]. Prieiga per: https://blog.group-ib.com/steam
10. SANDHU, Priyam Kaur; SINGLA, Sanjam. Google Safe Browsing—Web Security. *IJCSET*, 2015, 5: 283-287. Prieiga per: https://www.ijcset.net/docs/Volumes/volume5issue7/ijcset2015050717.pdf
11. BELL, Simon; KOMISARCZUK, Peter. An analysis of phishing blacklists: Google safe browsing, openphish, and phishtank. In: *Proceedings of the Australasian Computer Science Week Multiconference*. 2020. p. 1-11. Prieiga per: doi: https://doi.org/10.1145/3373017.3373020
12. SATHEESHKUMAR, M.; SRINIVASAGAN, K. G.; UNNIKRISHNAN, G. A lightweight and proactive rule-based incremental construction approach to detect phishing scam. *Information Technology and Management*, 2022, 1-28. Prieiga per: doi: https://doi.org/10.1007/s10799-021-00351-7
13. JAIN, Ankit Kumar; GUPTA, Brij B. Towards detection of phishing websites on client-side using machine learning based approach. *Telecommunication Systems*, 2018, 68.4: 687-700. Prieiga per: doi: https://doi.org/10.1007/s11235-017-0414-0

14. ASIRI, Sultan, et al. PhishingRTDS: A real-time detection system for phishing attacks using a Deep Learning model. *Computers & Security*, 2024, 141: 103843. Prieiga per: doi: https://doi.org/10.1016/j.cose.2024.103843
15. Proofpoint: Hiding in Plain Sight - Obfuscation Techniques in Phishing Attacks. 2016 [žiūrėta 2022-12-29]. Prieiga per: https://www.proofpoint.com/sites/default/files/proofpoint-obfuscation-techniques-phishing-attacks-threat-insight-en-v1.pdf
16. *Kitplot: Frameless-Bitb - A New Approach To Browser In The Browser (BITB) Without The Use Of Iframes* [interaktyvus]. 2024 [žiūrėta 2025-05-01]. Prieiga per: https://www.kitploit.com/2024/04/frameless-bitb-new-approach-to-browser.html
17. Jain, A. K., & Gupta, B. B. (2017). Phishing detection: analysis of visual similarity based approaches. *Security and Communication Networks*, *2017*. Prieiga per: doi: https://doi.org/10.1155/2017/5421046
18. Liu, W., Deng, X., Huang, G., & Fu, A. Y. (2006). An antiphishing strategy based on visual similarity assessment. *IEEE Internet Computing*, *10*(2), 58-65. Prieiga per: doi: https://doi.org/10.1109/MIC.2006.23
19. MAURER, Max-Emanuel; HERZNER, Dennis. Using visual website similarity for phishing detection and reporting. In: *CHI'12 extended abstracts on human factors in computing systems*. 2012. p. 1625-1630. Prieiga per: doi: https://doi.org/10.1145/2212776.2223683
20. CHIEW, Kang Leng, et al. Utilisation of website logo for phishing detection. Computers & Security, 2015, 54: 16-26. Prieiga per: doi: https://doi.org/10.1016/j.cose.2015.07.006
21. BOZKIR, Ahmet Selman; AYDOS, Murat. LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition. *Computers & Security*, 2020, 95: 101855. Prieiga per: doi: https://doi.org/10.1016/j.cose.2020.101855
22. LIN, Yun, et al. Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages. In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021. p. 3793-3810. Prieiga per: https://www.researchgate.net/publication/349684348_Phishpedia_A_Hybrid_Deep_Learning_Based_Approach_to_Visually_Identify_Phishing_Webpages
23. PANDEY, Pankaj; MISHRA, Nishchol. Phish-Sight: a new approach for phishing detection using dominant colors on web pages and machine learning. *International Journal of Information Security*, 2023, 1-11. Prieiga per: doi: https://doi.org/10.1007/s10207-023-00672-4
24. REN, Shaoqing, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 2015, 28. Prieiga per: doi: https://doi.org/10.1109/TPAMI.2016.2577031
25. REDMON, Joseph, et al. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p. 779-788. Prieiga per: doi: https://doi.org/10.48550/arXiv.1506.02640
26. HUSSAIN, Muhammad. Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo. *IEEE access*, 2024, 12: 42816-42833. Prieiga per: doi: https://doi.org/10.1109/ACCESS.2024.3378568
27. *Ultralytics: Explore Ultralytics YOLOv8* [interaktyvus]. 2023 [žiūrėta 2023-12-28]. Prieiga per: https://docs.ultralytics.com/models/yolov8/
28. LIU, Wei, et al. Ssd: Single shot multibox detector. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I*

14. Springer International Publishing, 2016. p. 21-37. Prieiga per: doi: https://doi.org/10.1007/978-3-319-46448-0_2

29. BHAVYA SREE, B.; YASHWANTH BHARADWAJ, V.; NEELIMA, N. An Inter-Comparative Survey on State-of-the-Art Detectors—R-CNN, YOLO, and SSD. In: *Intelligent Manufacturing and Energy Sustainability: Proceedings of ICIMES 2020*. Springer Singapore, 2021. p. 475-483. Prieiga per: doi: https://doi.org/10.1007/978-981-33-4443-3_46
30. WANG, Jing, et al. Logodet-3k: A large-scale image dataset for logo detection. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2022, 18.1: 1-19. Prieiga per: doi: https://doi.org/10.1145/3466780
31. ROMBERG, Stefan, et al. Scalable logo recognition in real-world images. In: *Proceedings of the 1st ACM international conference on multimedia retrieval*. 2011. p. 1-8. Prieiga per: doi: https://doi.org/10.1145/1991996.1992021
32. SU, Hang; ZHU, Xiatian; GONG, Shaogang. Open logo detection challenge. *arXiv preprint arXiv:1807.01964*, 2018. Prieiga per: doi: https://doi.org/10.48550/arXiv.1807.01964
33. *Okta: What Is Social Login and Is It Worth Implementing?* [interaktyvus]. 2020 [žiūrėta 2023-01-14]. Prieiga per: https://www.okta.com/blog/2020/08/social-login/
34. AKHIL, S. An overview of tesseract OCR engine. In: A seminar report. Department of Computer Science and Engineering National Institute of Technology, Calicut Monsoon. 2016. Prieiga per: doi: https://doi.org/10.1109/ICDAR.2007.4376991
35. *PyTorch: PyTorch documentation* [interaktyvus]. 2024 [žiūrėta 2024-01-12]. Prieiga per: https://pytorch.org/docs/stable/torch.html

Priedai

1 priedas. Testavimui naudotų tinklapių adresų sąrašas

https://zoom.us/signin#/login
https://login.ktu.lt/
https://www.instagram.com/accounts/login/
https://www.roblox.com/
https://login.yahoo.com/account/create
https://www.netflix.com/lt/login
https://secure.login.gov/
https://www.americanexpress.com/en-us/account/login
https://www.citi.com/login
https://login.apus.edu/padsts/login
https://olui2.fs.ml.com/login/signin.aspx
https://login.aol.com/account/create
https://id.spectrum.net/login
https://accounts.snapchat.com/accounts/v2/login
https://app.n26.com/login
https://login.k12.com/
https://login.flvs.net/
https://login.salesforce.com/?locale=us
https://www.hudl.com/Authentication
https://account.box.com/login?redirect_url=%2F
https://app.hubspot.com/login/
https://login.squarespace.com/api/1/login/oauth/provider
https://create.kahoot.it/auth/login
https://www.disneyplus.com/identity/login
https://login3.id.hp.com/login3/password
https://www.amazon.com/ap/signin
https://www.starbucks.com/account/signin
https://www.ups.com/lasso/login
https://secure.ssa.gov/RIL/SiView.action
https://www.frostbank.com/mf/
https://connect.secure.wellsfargo.com/auth/login/present?origin=cob
https://www.dropbox.com/login
https://accounts.spotify.com/en/login
https://access.paylocity.com/
https://loglogin.lastpass.com/?ac=1
https://mail.hostinger.com/
https://store.steampowered.com/login/

https://www.facebook.com/
https://accounts.google.com/v3/signin
https://steamcommunity.com/login/home/?goto=
https://www.paypal.com/signin
https://login.live.com/
https://www.linkedin.com/authwall