



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Jūratė Bosevičiūtė

**ELEKTRONINIŲ DOKUMENTŲ ŽENKLINIMO IR JŲ NUTEKINIMO PREVENCIJOS
METODŲ TYRIMAS**

**RESEARCH ON ELECTRONIC DOCUMENTS MARKING AND LEAKAGE
PREVENTION METHODS**

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų saugos studijų programa, valstybinis kodas 621E14007

Technologijos mokslų sritis

Informatikos inžinerijos studijų kryptis

Vilnius, 2017

Vilniaus Gedimino technikos universitetas

Fundamentaliųjų mokslų fakultetas

Informacinių sistemų katedra

ISBN ISSN

Egz. sk.

Data-....-....

Informacijos ir informacinių technologijų saugos studijų programos baigiamasis magistro darbas

Pavadinimas **Elektroninių dokumentų ženklavimo ir jų nutekėjimo prevencijos metodų tyrimas**

Autorius **Jūratė Bosevičiūtė**

Vadovas doc. dr. **Nikolaj Goranin**

Kalba



lietuvių



užsienio

Anotacija

Kiekvienais metais daugėja įsilaužimų į organizacijų sistemas, duomenų praradimo ar informacijos nutekėjimo incidentų. Todėl labai svarbu skirti didelį dėmesį svarbių dokumentų apsaugai ir jų identifikavimui organizacijos tinkle, kad svarbi ir konfidenciali informacija nenuitektų tretiesiems asmenims.

Šiame darbe analizuojamos teksto analizės technikos, kurios yra naudojamos sistemose, kurios siekia užtikrinti duomenų nepraradimą. Esant informacijos trūkumui apie šių sistemų naudojamas teksto analizės technikas ir kurios iš jų yra efektyviausios, pateikiamos šešios siūlomos teksto analizės technikos remiantis daliniu dokumentų turinio sutapimu, skaičiuojant maišos funkcijas dokumentų turiniui.

Sukonstravus šešis maišos funkcijų skaičiavimo technikų prototipus, remiantis siūlomomis realizacijomis, atlikti bandymai ir eksperimentai su paruoštais testiniais dokumentais. Gauti rezultatai palyginti ir suformuluotos išvados.

Darbą sudaro 6 dalys: įvadas, esamos situacijos ir metodų analizė, siūlomas metodas ir jo realizacija, rezultatai, išvados, literatūros sąrašas.

Darbo apimtis – 67 p. teksto be priedų, 29 iliustr., 18 lent., 32 bibliografiniai šaltiniai.

Atskirai pridedami darbo priedai.

Prasminiai žodžiai: elektroninių dokumentų sauga, dokumentų ženklavimas, maišos funkcijos, teksto turinio analizė, dokumentų dalinis sutapimas.

Vilnius Gediminas Technical University

Faculty of Fundamental Sciences faculty

Information systems department

ISBN ISSN

Copies No.

Date-....-....

Information and Information Technologies Security study programme bachelor (master) thesis.

Title: **Research on electronic documents marking and leakage prevention methods**

Author **Jūratė Bosevičiūtė**

Academic supervisor doc. dr. **Nikolaj Goranin**

Thesis language



Lithuanian



Foreign (English)

Annotation

Every year the number of data loss or information leakage incidents in organizations' systems increases. Therefore, it is crucial to focus more on the security of important documents and their identification in the organization's network in order to avoid sensitive and confidential information leakage to third parties.

In this thesis, text analysis techniques which are used by data security systems are analyzed. As the information of the texts analysis techniques used by security systems and their effectiveness is lacking, six text analysis techniques are offered based on partial documents coincidence and counting of hash functions.

Six prototypes of the computing hash functions techniques were created and, on the basis of proposed realizations, tests and experiments with prepared documents were conducted. The results were compared and conclusions obtained.

Structure: introduction, analysis of current situation and techniques, proposed method and its' realization, results, conclusions and references.

Thesis consist of: 67 p. text without appendixes, 29 pictures, 18 tables, 32 bibliographical entries.

Appendixes included.

Keywords: security of electronic documents, electronic documents marking, hash functions, text content analysis, documents partial coincidence.

Turinys

1.	Įvadas	8
1.1.	Tyrimo objektas	8
1.2.	Darbo tikslas ir uždaviniai	8
1.3.	Temos naujumas	9
1.4.	Temos aktualumas.....	9
1.5.	Tyrimo metodika.....	9
1.6.	Mokslinė darbo vertė	9
1.7.	Darbo rezultatai.....	9
1.8.	Darbo struktūra	10
2.	Esama Informacijos saugos situacija ir metodai	11
2.1.	Esama informacijos saugumo situacija	11
2.2.	DLP sistemos	13
2.3.	Turinio analizės technikos	16
3.	Siūlomas dalinio sutapimo metodas ir jo realizacija.....	26
3.1.	Tyrimo eiga	26
3.2.	Maišos funkcijų skaičiavimo prototipai	29
4.	Taikytų dalinio sutapimo strategijų rezultatai.....	35
4.1.	Eksperimentas, kai pridėtas tekstas dokumento pabaigoje	35
4.2.	Eksperimentas, kai pridėtas tekstas dokumento pradžioje.....	40
4.3.	Eksperimentas, kai pridėtas tekstas dokumento viduryje	46
4.4.	Eksperimentas, kai pakeisti atsitiktiniai žodžiai dokumentų tekste	52
4.5.	Greitaveika	58
5.	Išvados	63
6.	Literatūros sąrašas.....	65
7.	Priedai	68
7.1.	Prototipo realizacija	68
7.2.	Maišos funkcijų lyginimas	74

Paveikslų sąrašas

1 pav. Incidentų statistika, susijusi su informacijos praradimu ir nutekėjimu [1]	11
2 pav. Tinklo stebėjimas	15
3 pav. El. pašto integravimas	15
4 pav. Įgaliojimas (angl. <i>proxy</i>)	15
5 pav. Kreditinių kortelių taisyklėmis pagrįsta išraiška [21]	18
6 pav. Luno algoritmas [23]	19
7 pav. Būsenų transformatorius [30]	24
8 pav. Eksperimento eiga	26
9 pav. Parengti dokumentai	27
10 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui	35
11 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais	37
12 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems žodžiams tekste	38
13 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas antram ir trečiam sakiniui tekste	39
14 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui	41
15 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais	42
16 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems žodžiams tekste ..	44
17 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems sakiniams tekste ..	45
18 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui	47
19 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais	48
20 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas tik kas k -iesiems žodžiams tekste	50
21 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems sakiniams tekste ..	51
22 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui	53
23 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais	54
24 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems žodžiams tekste ..	56
25 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems sakiniams tekste ..	57
26 pav. Greitaveika skaičiuojant maišos funkcijas skirtingam kiekiui žodžių be persidengimų	59
27 pav. Greitaveika skaičiuojant maišos funkcijas skirtingam kiekiui žodžių su persidengimais	60
28 pav. Greitaveika skaičiuojant maišos funkcijas kas k -iesiems žodžiams	61
29 pav. Greitaveika skaičiuojant maišos funkcijas kas 2 ir 3 sakiniui	62

Lentelių sąrašas

1 lentelė. Maišos algoritmų lyginimas [32]	23
2 lentelė. Testiniai dokumentai	27
3 lentelė. Atitikimų procentas pagal dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas.....	36
4 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas su persidengimais	37
5 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius žodžius skaičiuojant maišos funkcijas	39
6 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius sakinius skaičiuojant maišos funkcijas	40
7 lentelė. Atitikimų procentas pagal dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas.....	42
8 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas su persidengimais	43
9 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius žodžius skaičiuojant maišos funkcijas	44
10 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius sakinius skaičiuojant maišos funkcijas	45
11 lentelė. Atitikimų procentas pagal dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas.....	47
12 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas su persidengimais	49
13 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius žodžius skaičiuojant maišos funkcijas	50
14 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius sakinius skaičiuojant maišos funkcijas	51
15 lentelė. Atitikimų procentas pagal dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas.....	53
16 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas su persidengimais	55
17 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir k -uosius žodžius skaičiuojant maišos funkcijas	56
18 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius sakinius skaičiuojant maišos funkcijas	57

Terminai

DLP – duomenų apsaugos sistema (angl. *Data Loss Protection*);

Maiša – tai matematinė transformacija, atliekama verčiant tam tikro ilgio simbolių eilutę į fiksuoto ilgio reikšmę.

Maišos algoritmas – maišos funkcijų iš anksto taisyklėmis nustatytas ir apibrėžtas nuoseklus vykdymas.

Maišos funkcija – tai funkcija $f = f(x)$ argumentui x priskiriant pseudoatsitiktinį skaičių, vadinamą maišos kodu. Tam pačiam argumentui funkcija turi teikti tą patį rezultatą, todėl ji nėra atsitiktinė.

1. ĮVADAS

Vis daugiau dokumentų yra perkeliama į elektroninę erdvę, todėl šiuo metu yra labai svarbus šių dokumentų saugumo ir jų identifikavimo organizacijos tinkle klausimas. Yra nemažai dokumentų, kurie yra persiunčiami tretiesiems asmenims be autoriaus žinios, taip pažeidžiant saugumą ir konfidencialumą. Taip pat platinama svarbi ir konfidenciali organizacijų informacija, kuri gali paveikti neigiamai ne tik pačią organizaciją, bet ir su ja susijusius asmenis (darbuotojus, klientus ir t. t.). Tam užkirsti kelią yra kuriamos programinės įrangos, skirtos elektroninių dokumentų apsaugai ir jų plitimo stabdymui bei stebėjimui. Šios sistemos (pvz., „Symantec Data Loss prevention“, „Data Loss Prevention Software Blade“, „McAfee Total Protection for Data Loss Prevention“, „MyDLP“ ir kitos) yra komercinė. Korporacijų sukurti dokumentų stebėjimo, teksto analizės ir identifikavimo mechanizmai, atlikti tyrimai ir bandymai nėra publikuojami ir viešai prieinami, todėl nėra aišku, kurios iš teksto analizės technikų yra efektyviausios.

Saugumo sprendimai, metodai greitai sensta ir dažnai po kurio laiko jau nebegali užtikrinti apsaugos. Tai parodo ir paskutinių metų statistika, incidentų skaičius kiekvienai metais didėja ir padaro vis daugiau žalos organizacijoms ir jų klientams, todėl norint užkirsti tam kelią, svarbu nuolat vykdyti tyrimus ir siūlyti naujus metodus informacijos saugumo srityje.

1.1. Tyrimo objektas

Duomenų nutekėjimo prevencijos metodai.

1.2. Darbo tikslas ir uždaviniai

Darbo tikslas – pasiūlyti pagerintą elektroninių dokumentų ženklavimo metodą, skirtą dokumentų identifikavimui ir nutekėjimo prevencijai.

Uždaviniai:

1. Atlikti sukurtų sistemų ir jų naudojamų technikų apžvalgą bei analizę;
2. Pasiūlyti elektroninių dokumentų ženklavimo metodą;
3. Sukonstruoti siūlomo metodo prototipą ir atlikti bandymus;
4. Pateikti rezultatus ir išvadas.

1.3. Temos naujumas

Šiuo metu didelė dalis asmenų ir organizacijų, siunčiančių nekontroliuojamą srautą konfidencialių dokumentų tik parodo, kad elektroninių dokumentų saugos ir jų judėjimo identifikavimo ir sekimo problema yra aktuali ir jai vis dar ieškoma geriausių ir efektyviausių sprendimų.

Taip pat, saugumo prasme, tokio pobūdžio temos išlieka vis dar aktualios. Remiantis statistiniais duomenimis įsilaužimo ir duomenų praradimo incidentų kiekvienais metais daugėja ir tampa vis didesne problema. Sprendimai greitai sensta, jie tampa perprantami ir greitai atsiranda saugumo spragų. Todėl turi būti nuolat ieškoma naujų ir inovatyvių sprendimų informacijos saugumo ir konfidencialumo užtikrinimo tema.

1.4. Temos aktualumas

Organizacijos, turinčios konfidencialius dokumentus, kurie turi didelę reikšmę organizacijai tiek fiziniams asmenims (darbuotojams/klientams), nėra užtikrinti, kad jie nebus persiųsti tretiesiems asmenims ir nebus paviešinti ar paskelbti internete, to nenorint autoriui ir pačiai organizacijai. Tam užkirsti kelią jau yra sukurta ne mažai sprendimų, tačiau jie yra komerciniai ir jų naudojamos tikslios metodikos ir tyrimai, kurie iš metodų ir turinio analizės technikų yra geresni, nėra viešai prieinami. Dėl šių problemų yra organizacijų, kurios taiko per menką apsaugą nuo duomenų ir informacijos praradimo.

1.5. Tyrimo metodika

Analitinėje darbo dalyje, analizuojant esamas sistemas naudojami lyginamosios analizės ir vertinimo metodai. Siūlomos realizacijos kūrimui naudojami loginės indukcijos ir veiklos modeliavimo metodai, o praktinei realizacijai – konstravimo, eksperimentinės ir statistinės analizės metodai.

1.6. Mokslinė darbo vertė

Pasiūlyti ir sukonstruoti šeši metodai dokumentų ženklinimui ir identifikavimui.

1.7. Darbo rezultatai

Darbo metu yra gauti tokie rezultatai:

- pasiūlytos 6 dalinio turinio analizės strategijos;

- sukonstruoti prototipai realizuojantys skirtingas strategijas;
- atliktas eksperimentas su prototipais naudojant paruoštus testinius dokumentus;
- atlikti bandymai parodė, kad viena strategija nėra tinkamiausia visų tipų dokumentų ženklavimui.

1.8. Darbo struktūra

Šiame skyriuje yra pateikiama baigiamojo darbo struktūra:

Antrame skyriuje yra nagrinėjamos esamos, duomenų nuo paradimo sistemos ir jų naudojamos turinio analizės technikos;

Trečiame skyriuje pateikiama siūlomo dokumentų ženklavimo metodo realizacija dokumentų identifikavimui;

Ketvirtame skyriuje pateikiami pasiūlyto metodo taikymo rezultatai ir jų aprašymas;

Penktame skyriuje pateikiamos išvados;

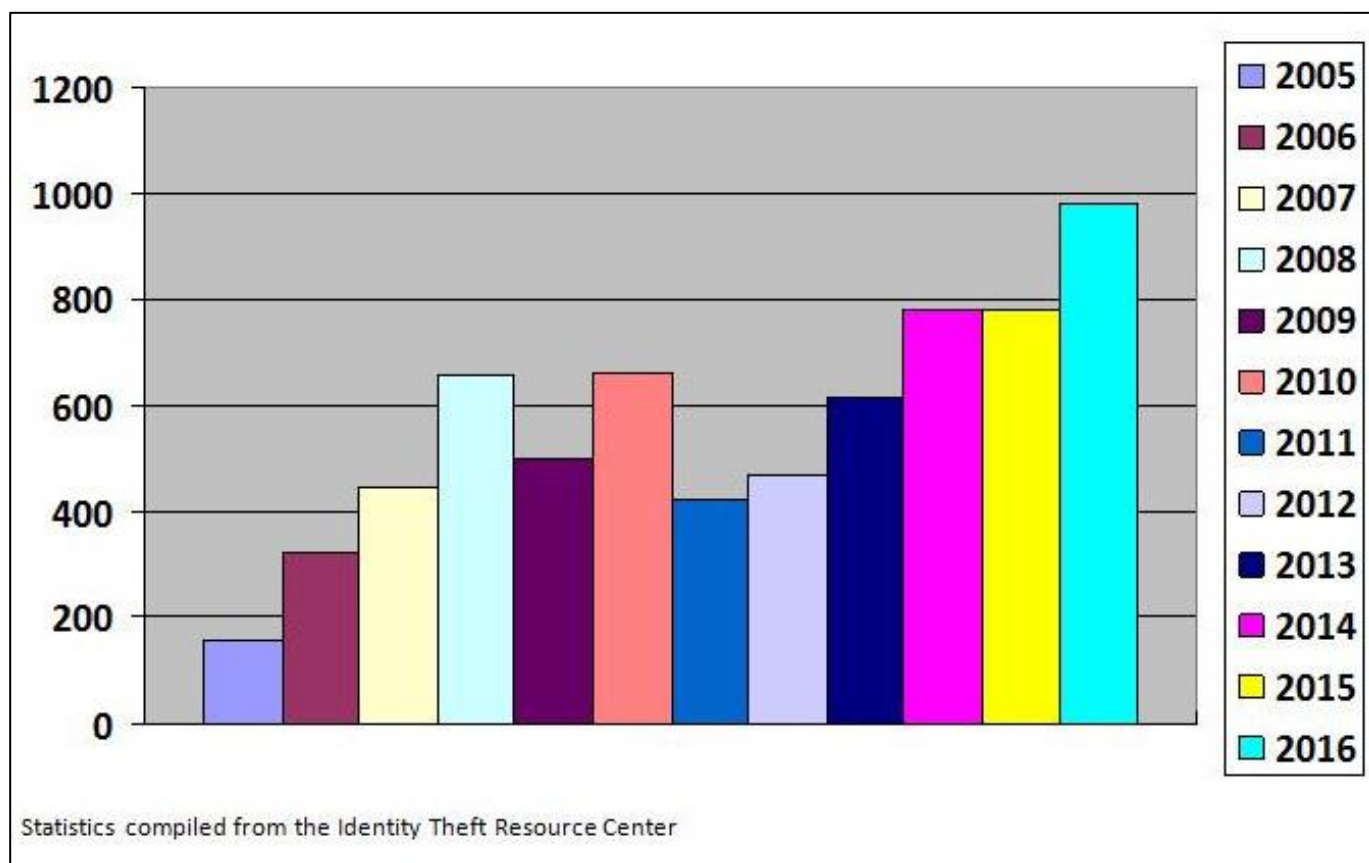
Šeštame skyriuje pateikiamas naudotas literatūros sąrašas;

Septintame skyriuje pateikiami priedai.

2. ESAMA INFORMACIJOS SAUGOS SITUACIJA IR METODAI

2.1. Esama informacijos saugumo situacija

Kiekvienais metais daugėja įsilaužimų į informacines sistemas, duomenų praradimo ar informacijos nutekėjimo incidentų. Remiantis statistiniais duomenimis, pateiktais ITRC (angl. *Identity Theft Resource Center*) [1], 2016 metais tokių incidentų buvo užfiksuota daugiausiai, lyginant nuo 2005 m. Šie incidentai pasireiškia įvairiomis formomis: konfidencialios informacijos kopijavimu, perdavimu, peržiūrėjimu, pavogimu, praradimu ar panaudojimu.



1 pav. Incidentų statistika, susijusi su informacijos praradimu ir nutekėjimu [1]

Duomenys yra klasifikuojami kaip struktūriniai arba nesisteminiai. Kaip rašoma „PC MAG“ žodyne [2], struktūriniai duomenys egzistuoja stacionariuose, fiksuotuose failuose (pvz., skaičiuoklėse, duomenų bazėse), o nesisteminiai duomenys yra susiję su laisvos formos teksta, esančiais tekstiniuose dokumentuose, PDF failuose, el. pašto žinutėse ir pan. Kaip teigia puslapio „PC MAG“ autoriai [1], daugiausiai incidentų susijusių su nesisteminiais duomenimis, o tai kelia didelę grėsmę organizacijoms, nes remiantis autoriaus Burke Brian E. skaičiais [3], net 80 % visų duomenų sudaro nesisteminiai duomenys.

Duomenų nuo praradimo prevencijos programinės įrangos yra kuriamos siekiant aptikti galimas duomenų saugumo priežastis ir pažeidimus, stebėti, nustatyti ir blokuoti slaptus duomenis. Slapta ir jautri informacija negali būti atskleidžiama pašaliniam darbuotojams ar asmenims, norint išvengti netyčinių kenksmingų veiksmų ar klaidų. Tokia slapta ar jautri informacija gali būti apie privačius asmenis ar kompanijas, intelektines nuosavybes, kredito kortelių duomenis ir t. t., priklausomai nuo organizacijos veiklos.

Analizuojant literatūrą, susijusią su informacijos praradimu, yra dažniausiai naudojamos dvi sąvokos: „duomenų praradimas“ ir „duomenų nutekėjimas“. Kaip rašo knygoje autoriai Shabtai A., Elovici Y. ir Rokach L. [4], sąvokos „duomenų praradimas“ ir „duomenų nutekėjimas“ yra glaudžiai susijusios ir dažnai naudojamos kaip sinonimai, tačiau jų reikšmės šiek tiek skiriasi. „duomenų praradimas“ reiškia informacijos praradimą, kai šiuos duomenis perima tretis asmenys, o „duomenų nutekėjimas“ – informacijos nutekėjimą tretiesiems asmenims, tačiau pati informacija yra neprarandama ir lieka savo kilmės pusėje.

Sprendžiant informacijos nutekėjimo problemas technologinėmis priemonėmis, šios priemonės gali būti skirstoma į šias kategorijas: standartinės saugumo priemonės, pažangios/protingos saugumo priemonės, prieigos kontrolės ir šifravimo bei duomenų nuo praradimo prevencinės sistemos (angl. *data loss protection (DLP)*) [5].

Standartinės saugumo priemonės. Standartinės saugumo priemonės yra tokios kaip: ugniasienės, įsilaužimo aptikimo sistemos ir antivirusinės programos. Tokios saugumo priemonės yra pagrindinis mechanizmas, siekiant apsisaugoti nuo įsilaužėlių ir jų atakų. Ugniasienės riboja pašaliečių prieigą prie vidinio tinklo, o įsibrovimo aptikimo sistema – aptinka prašalaičių bandymus įsibrauti į vidinį tinklą. Viduje išvengti išpuolių padeda antivirusinės programos, kurios aptinka Trojos arklius, įdiegtus kompiuteryje [5].

Pažangios/protingos saugumo priemonės. Išplėstinės apsaugos priemonės, įdarbinančios mašinas, algoritmus aptikti nenormalias prieigas prie duomenų (t. y. duomenų bazių ar informacijos paieškų sistemų) arba nenormalius elektorinio pašto pasikeitimus su kenkėjiškais ketinimais. Naudotojų veiklos stebėjimas, aptinkant neįprastas naudotojo prieigas prie duomenų [5].

Prieigos kontrolės ir šifravimo bei DLP sistemos. Kaip rašo Phua C. savo straipsnyje [5], specialūs DLP sprendimai yra skirti aptikti ir užkirsti kelią leidimo neturintiems darbuotojams kopijuoti, siųsti tyčia (ar netyčia) slaptus duomenis be leidimo, tačiau, kuriems leista naudotis slaptais duomenimis. Papildant autoriaus Ouellet E. paruošta technine ataskaita [6], siekiant suklasifikuoti tam tikrą informaciją, kaip „jautrius“ duomenis, naudojami įvairūs mechanizmai, tokie kaip: tikslų duomenų atitikimas, struktūrizuotų

duomenų žymėjimas, statistiniai metodai, taisyklių ir įprastų išraiškų sutapimai, daliniai sutapimai, paskelbti žodynai, konceptualios sąvokos ir žodžiai.

2.2. DLP sistemos

Analizuojant informaciją aiškėja, kad šiuo metu vis daugiau organizacijų teikia vis daugiau dėmesio pažangioms ir vis tobulinamoms DLP sistemoms, kurios, deja, beveik visos šiuo metu yra komercinės. Šiuo metu populiariausios DLP sistemos yra: „Symantec Data Loss prevention“, „Data Loss Prevention Software Blade“, „McAfee Total Protection for Data Loss Prevention“ ir „MyDLP“.

Kaip rašo autoriai: Readshaw Neil I., Ramanathan J. ir George Bray G. [7], DLP tikslas yra prižiūrėti ir saugoti kritinę informaciją. DLP skiriasi nuo kitų saugos įrankių (pvz., ugniasienės) tuo, kad DLP koncentruojasi tik į „jautrius“ duomenis, kuriuos būtinai reikia apsaugoti. DLP kategorizuoja „jautrią“ informaciją, naudoja įvairias turinio analizės technikas, kontroliuoja informacijos srautus ir audituoja incidentus [8]. DLP sistema veikia naudotojo darbiname kompiuteryje ar serveryje, kuriame kontroliuoja duomenų srautus tarp grupių ir atskirų tipų naudotojų. Kontroliuojami srautai gali būti ne tik persiuntimas elektroniniu paštu, įkėlimas į tinklą, persiuntimas bet ir kopijavimas į kitas laikmenas (pvz., USB raktą) [9].

Išanalizavus esamą situaciją pastebėta, kad didžiausia grėsmė kyla nestruktūrizuotiems duomenims (tekstiniais dokumentais), padaryta išvada, kad dokumentų stebėjimui ir analizei įmonės tinkle, efektyviausios yra DLP sistemos, kurios taiko įvairias turinio analizės ir duomenų perėmimo technikas. Remiantis tuo, kad didelę reikšmę nestruktūrizuotiems duomenims turi turinio analizės technikos, kurios naudojamos tinkle stebimiems dokumentams/informacijai identifikuoti kaip „jautriais“ (saugomais nuo praradimo), pasirinkta gilesnė šių technikų analizė.

2.2.1. Duomenų būsenos ir jų perėmimo technikos

Šiame skyriuje aprašomos duomenų būsenos ir DLP sistemų naudojami metodai, kuriuos taiko dokumentams ir turiniui fiksuoti, tikrinimui naudojant turinio analizės technikas.

„Data at-rest“. Nurodo informaciją, kuri yra saugoma kliento kietajame diske, tinklo saugykloje, nuotoliniame failų serveryje ar net atsarginėse laikmenoje (pvz., CD). Tokia informacija yra paliekama nenaudojama, ir turi didelę grėsmę patekti pašaliniams asmenims už tinklo ribų [11]. Kaip teigia knygos autoriai: Shabtai A., Elovici Y. ir Rokach L. [4], siekiant apsaugoti šį duomenų etapą, sistemos naudoja metodus, tokius kaip: prieigos kontrolė, duomenų šifravimas ir t. t.

Tokiai informacijai DLP sistemos naudoja turinio aptikimą, kuris paremtas priklausomai nuo to, kur informacija yra saugoma [13]. Yra pagrindinės keturios turinio aptikimo technikos:

- **Nuotolinis skenavimas.** Specialus skenavimo serveris pasiekia saugyklas per viešą tinklą ar naudojant administratoriaus teises.
- **Agentų skenavimas.** Įdiegtas agentas sistemoje (serveryje), todėl skenavimas atliekamas lokaliai. Agentas naudoja vietinį CPU, tačiau esant dideliems kiekiams duomenų, skenavimas užtrunka daug greičiau nei skenuojant nuotoliniu būdu.
- **Laikinas agento skenavimas.** Tai laikinas agentas sistemoje (serveryje), kuris nuskenavęs duomenis išsina ir nepalieka veikiančių procesų. Ši technika geresnė tada, kai yra sistema, kurioje nenorima, kad agentas nuolatos apkrautų sistemą veikiančiais procesais.
- **Sistemos integracija.** Tiesioginis integravimas (dažnai naudojant agentus) su dokumentų, turinio valdymo sistemomis ar duomenų bazėmis. Ši integraciją palaiko ne tik matomumą, bet ir leidžia suprasti kontekstą, metaduomenis, atlikti veiksmus per sistemą.

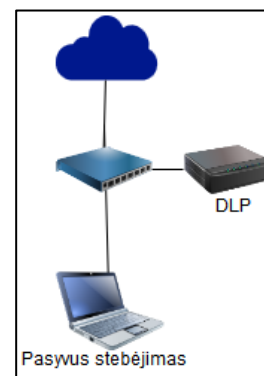
„**Data in-use**“. Nurodo naudojamus duomenis, kurie yra šiuo metu sąveikaujantys su naudotojais. Shabtai A., Elovici Y. ir Rokach L. knygoje rašo [4], kad DLP sistemos, kurios apsaugo duomenis, gali tik stebėti ir žymėti tam tikrą neteisėtą veiklą. Ši veikla apima ekrano vaizdo surinkimą, kopijavimą, kopijuoti/įklijuoti operacijas, spausdinimo, siuntimo faksu operacijas su slaptais duomenimis.

Naudojant DLP sistemas yra diegiami agentai į pabaigos taškus (angl. *endpoints*), kurie apima [13]:

- **Turinio aptikimas.** Nustatomi saugomo turinio politiko pažeidimai;
- **Failinės sistemos apsauga.** Kontrolė ir riboto vykdymo užtikrinimas su failais. Dažniausiai naudojama užkertant kelią įkelti failus į nešiojamas laikmenas (pvz., USB).
- **Tinklo apsauga.** Kontrolė ir tinklo operacijų vykdymas. Kontroliuoja srautą spausdinant, siunčiant faksu.
- **Branduolio apsauga.** Bendresnis sprendimas apsaugant duomenis, juos naudojant, yra ribojimas funkcijų „Copy & Paste“, „Print screen“ ir kitų programų.

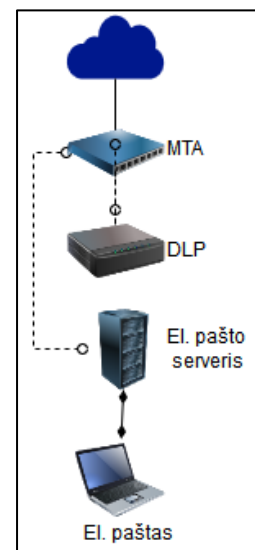
„**Data in-motion**“. Nurodo duomenis, kurie yra teikiami tinklu į paskirties tašką. Šie tinklai gali būti vidiniai ir išoriniai. DLP sistemos apsaugo duomenis, kai jie yra siunčiami tinklu per įvairius komunikacijos kanalus [12].

- **Tinklo stebėjimas.** Remiantis Mogull R. parašyta ataskaita [13], DLP sprendimų pagrindas yra pasyvus tinklo stebėjimas. Tinklo stebėjimo komponentas yra įprastai dislokuojamas arba statomas ant SPAN kanalo. Šis komponentas atlieka visų paketų surinkimą, sesijos peržiūrą ir turinio analizę realiu laiku. Kai kurie gamintojai šį DLP sprendimą diegia į bendrosios paskirties serverio aparatinę įrangą, o kiti – dislokuoja atskirą specializuotą prietaisą.



2 pav. Tinklo stebējimas

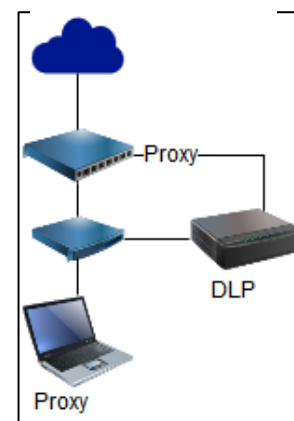
- **Elektroninio pašto integravimas.** Tai kitas, vienas iš svarbiausių, komponentų. El. paštu galima perduoti daug vertingos informacijos ir nemažais pajėgumais. Todėl dauguma DLP produktų įdedamas į MTA (angl. *Mail Transport Agent*), kurį galima tiesiog pridėti į el. pašto grandinę. Šio metodo silpnybė – jis nesuteikia prieigos prie vidinio elektroninio pašto. Esant Exchange serveryje, vidinės žinutės niekuomet nepasieks išorinio MTA, jei nėra prasmės jų siųsti į išorę. Norint stebėti vidinį el. paštą, reikia tiesioginės „Exchange/Lotus“ integracijos, kuri yra stebėtinai labai reta [13].



3 pav. El. pašto integravimas

- **Tiltas.** Naudojant tiltą, turime tiesiog sistemą su dviem tinklo plokštėmis, kurios vykdo turinio analizę. Jei įvyksta kas nors blogo, tos sesijos ryšys nutraukiamas. Tiltas nėra geriausias sprendimas DLP nuo tada, kai jis gali nesustabdyti blogo srauto prieš tai, kol jis yra perduodamas. Labai mažai produktų naudoja šį metodą [13].

- **Įgaliotinis.** Paprasčiau tariant, tai yra protokolas/programa specifikuojanti srautą, kol jis dar neišėjo ir leidžia atlikti gilesnę srauto analizę. Dažniausiai įgaliotinis (angl. *proxy*) naudojamas HTTP, FTP ir IM protokolams. Keli DLP sprendimai prideda ir savo pačių įgaliotinį. Integracija išėjimui į tinklą daroma dažniausiai per iCAP protokolą, leidžiant



**4 pav. Įgaliojimas (angl.
proxy)**

įgaliotiniui perimti srautą ir jį perduoti analizei DLP produktui. Jei yra pažeidimas, ryšys yra nutraukiamas [13].

- **TCP apnuodijimas.** Šis metodas skirtas tinklui stebėti, kai pamatomas pažeidimas, yra įterpiamas TCP anuliovimo paketas, panaikinant ryšį. Tai veikia kiekvienam TCP protokolui, bet nėra labai veiksmingas. Vienu atveju, kai kurie protokolai nuolat bandys atgauti srautą. Jei TCP apnuodysime vieną elektorinę žinutę, serveris ją bandys išsiųsti dar 3 dienas kas 15 minučių. Kita problema – problemos pastebėjimas per vėlai, kai ji jau yra perduota [13].

2.3. Turinio analizės technikos

Šiame skyriuje aprašytos DLP sistemų naudojamos turinio analizės technikos.

2.3.1. Statistinė analizė

Kaip rašo Walsh B. [14], iš visų statistinių metodų dažniausiai naudojama Bajeso (angl. *Bayesian*) analizė, kuri skirta identifikuoti dokumentą kaip jautrų, remiantis dokumentų turinio panašumais, tačiau kaip teigia Alfikri Zakiy F. ir Purwarianti A. [15], palaikomos vektoriaus mašinos (angl. *Support vector machine (SVM)*) algoritmas geresnis klasifikuojant tekstą, kur tai galima matyti eksperimentuose, kuriuos aprašė autoriaus Joachims T. [16] ir Sebastiani F. [17]. Tačiau Bajeso algoritmas naudojamas dažniau todėl, kad jis tinkamas naudoti pagrindines teksto klasifikavimo problemas ir tai galima pamatyti taip pat tuose pačiuose pateiktuose eksperimentuose [16][17].

2.3.2. Bajeso analizė

Kaip nurodo Robinson G. [18], remiantis Bajeso analize, turinys – žodis po žodžio yra analizuojamas vertinant tikimybę, ar dokumente yra talpinama jautri informacija, aptikus sekantį žodį. Tada bendra tikimybė yra apskaičiuojama pagal Bajeso formulę. Remiantis reikšmingumo lygmeniu, hipotezė, kad dokumente yra talpinama jautri informacija yra patvirtinama arba atmetama.

Toliau pateikiamas Bajeso analizės veikimo principas remiantis Robinson G. [18] ir Lee Peter M. [19] pateiktomis formulėmis ir aprašymu:

$$\text{Bajeso formulė: } \Pr(S|W) = \frac{\Pr(W|S)\Pr(S)}{\Pr(W|S)\Pr(S) + \Pr(W|H)\Pr(H)} \quad (1)$$

$\Pr(S|W)$ – tikimybė, kad dokumento turinyje yra saugoma (konfidenciali) informacija, žinant, kad jame yra abejotinų išsireiškimų (žodžių);

$\Pr(S)$ – bendra tikimybė, kad bet kokiam dokumente yra saugoma (konfidenciali) informacija;

$\Pr(W|S)$ – tikimybė, kad saugomas išsireiškimas (žodis, kodas) bus dokumente, kuriame skelbiama saugoma (konfidenciali) informacija;

$\Pr(H)$ – bendra tikimybė, kad dokumente nėra saugomos (konfidencialios) informacijos;

Dažniausiai $\Pr(S) = 0,5$ ir $\Pr(H) = 0,5$. Kadangi tikimybė yra lygi, kad dokumente bus skelbiama saugoma informacijos arba ne. Taigi toks filtras yra nešališkas, tikrinant dokumentą nėra iš ankstinio nusistatymo, kad dokumente yra talpinama jautri informacija. Taigi ši prielaida leidžia supaprastinti formulę taip:

$$\Pr(S|W) = \frac{\Pr(W|S)}{\Pr(W|S) + \Pr(W|H)} \quad (2)$$

Ši formulė atitinka klausimą „Kiek procentų, kad saugomos informacijos yra dokumente?“.

Sprendžiant iš vieno „jautraus“ dėmens, ar dokumente yra talpinami saugomi (konfidencialūs) duomenys ar ne, teikia didelę tikimybę klaidų, todėl remiantis Bajeso formule stengiamasi remtis keliais dėmenimis ir bandant juos apjungti nustatyti bendrą tikimybę, ar dokumente yra saugoma (konfidenciali) informacija.

Dauguma Bajeso algoritmų, filtruojančių dokumentus, yra pagrįsti formule, kuri yra teisinga (tikimybinio požiūriu) tik tada, kai informacija (žodžiai/ išsireiškimai) esantys dokumente yra nepriklausomi įvykiai. Ši sąlyga paprastai nėra tenkinama, nes sakiniuose sudėlioti žodžiai dažniausiai yra vieni nuo kitų priklausomi, kad sudarytų logišką sakinį. Bet šiuo atveju tai yra naudingas idealizavimas, ypač, kai statistinė koreliacija tarp žodžių nėra žinoma. Remiantis tuo, galima išvesti sekančia formulę iš Bajeso teoremos:

$$p = \frac{p_1 p_2 \dots p_N}{p_1 p_2 \dots + (1-p_1)(1-p_2) \dots (1-p_N)} \quad (3)$$

p – tikimybė, kad dokumente yra saugoma (konfidenciali) informacija;

p_1 – tikimybė $p(S|W_1)$, kad dokumente yra saugoma (konfidenciali) informacija, aptikus pirmą žodį (pvz., „Slaptažodis“);

p_2 – tikimybė $p(S|W_2)$, kad dokumente yra saugoma (konfidenciali) informacija, aptikus antrą žodį (pvz., „Slaptas“);

...

...

ir t. t.

p_N – tikimybė $p(S|W_N)$, kad dokumente yra saugoma (konfidenciali) informacija, aptikus N -ąjį žodį (pvz., „Katinas“);

Tikimybė p yra paprastai lyginama su reikšmingumo lygmeniu. Jei p yra mažiau už reikšmingumo lygmenį, tai dokumentas laikomas įprastu, jei daugiau – saugomu/konfidencialiu.

2.3.3. Konceptija/žodynai

Jautrūs dokumentai identifikuojami pagal jų miglotame turinyje naudojamus žodžius, kurie yra įtraukti į konfigūruojamą žodyną. T. y., gali būti nurodomos konkrečios saugomos frazės, žodžiai ir pan. Naudinga identifikuojant nestruktūrizuotus dokumentus, idėjas [13]. Aptikus saugomą turinį, dokumentas iš karto yra fiksuojamas kaip konfidencialus.

Kaip rašo autoriai: Osman A., Salim N. ir Elhadi A. [20], taikant koncepciją svarbu tekstą išskaidyti į sakinius, temas, pastraipas ar kitas reikšmingas dalis, kurias norima identifiкуoti. Žodynai turi būti sudaromi ir laikomi duomenų bazėje, kad būtų patikrinami pagal visus galimus linksnius, gimines ir pan. Taip pat užtikrinant efektyvesnį identifikavimą turi būti pašalinami nereikalingi žodžiai, pvz., anglų kalboje naudojami artikeliai „an“, „a“, „the“ ir pan.

2.3.4. Taisyklėmis pagrįstų / reguliarių išsireiškimų identifikavimas

Šio turinio analizės technikos tikslas, kad duomenų eilutės sutapimas būtų identifiкуotas, kaip jautri informacija, remiantis aprašytomis taisyklėmis. Tokios turinio analizės technikos bendras pavyzdys pateiktas Mogull R. [13] ir Devlin R. [21], kur aprašyta jautrių duomenų paieška remiantis kreditinių kortelių sudarymo numeriais.

Paprasta, taisyklėmis pagrįsta išraiška (naudojant „regex“ funkcijas), kuria remiantis gali būti identifiкуota jautri informacija, pateikiama žemiau:

```
^(?:4[0-9]{12}(?:[0-9]{3})?          # Visa
| (?:5[1-5][0-9]{2}|
  222[1-9]|22[3-9][0-9]|2[3-6][0-9]{2}|
  27[01][0-9]|2720)[0-9]{12}      # MasterCard
| 3[47][0-9]{13}                  # American Express
)$
```

5 pav. Kreditinių kortelių taisyklėmis pagrįsta išraiška [21]

Detalus šių taisyklių aprašymas ir panaudojamas aprašytas [22]:

- Visa: $^4[0-9]\{12\}(?:[0-9]\{3\})?\$$. Visų „Visa“ kortelių numeriai prasideda skaičiumi 4. Naujos kortelės turi 16 skaitmenų, o senos – 13.
- MasterCard: $^(?:5[1-5][0-9]\{2\}|22[1-9]|22[3-9][0-9]|2[3-6][0-9]\{2\}|27[01][0-9]|2720)[0-9]\{12\}\$$. „MasterCard“ kortelių numeriai prasideda nuo skaičių nuo 51 iki 55 ar su skaičiais nuo 2221 iki 2720. Visos kortelės turi 16 skaitmenų.
- American Express: $^3[47][0-9]\{13\}\$$ American Express kortelių numeriai prasideda nuo 34 arba 37 ir turi 15 skaitmenų.

Naudojantis šiomis taisyklėmis, stebint organizacijos tinką, fiksuojamas galimas informacijos nutekėjimas.

Be standartinių kreditinių kortelių formatų, kredito kortelės naudoja algoritmą, žinomą kaip Luno (angl. *Luhn*), kurio formulė matematiškai apskaičiuoja galiojančios kortelės numerį [21]. Luno algoritmas yra labai paprastas: kas antras skaičius yra sudauginamas, jei gaunasi dviženklis skaičius – skaičiai sudedami. Gavus skaičių eilutę, visi skaičiai yra sudedami. Jei gautas skaičius baigiasi 0, tai numeris yra teisingas [23].

4	9	9	2	7	3	9	8	7	1	6
	x2		x2		x2		x2		x2	
	18		4		6		16		2	
4	9	9	4	7	6	9	7	7	2	6
4 + 9 + 9 + 4 + 7 + 6 + 9 + 7 + 7 + 2 + 6 = 70										

6 pav. Luno algoritmas [23]

Taip pat, tokiu principu, pagal atitinkamas taisykles gali būti identifikuojami ir kiti jautrūs duomenys, pavyzdžiui asmens kodai, elektroninių paštų adresai, telefono numeriai ir t. t.

2.3.5. Tikslų duomenų sutapimas

Šis metodas skirtas konkrečios informacijos sutapimui identifikuoti. Pvz., nurodomi konkretūs (tikslūs) kreditinių kortelių numeriai, asmens kodai, tekstinė eilutė ir t. t., kurie yra laikomi saugomais (konfidencialiais) ir jie turi būti identifikuoti. Šiuo atveju, kitaip, nei taisyklėmis pagrįstu metodu, taisyklės nėra apibrėžiamos, ieškoma nurodytų tikslų skaičių sekos ar tekstinės eilutės atitikimo [13].

Greitesniam tikslios informacijos aptikimui yra naudojamos maišos funkcijos (angl. *hash functions*) [8]. Konkrečiai, saugomai informacijai: asmens kodams, kreditinių kortelių numeriams, tekstinėms eilutėms ir pan. yra apskaičiuojamos maišos funkcijos naudojant pasirinktą maišos funkcijos algoritmą (MD5, SHA-1, SHA-256 ir t. t.). Toliau organizacijos tinkle judantiems failų turiniams taip pat yra skaičiuojamos maišos funkcijos ir lyginamos su saugomos informacijos maišos funkcijomis, jei jos sutampa – dokumentas fiksuojamas kaip talpinantis saugomą (konfidencialią) informaciją.

2.3.6. Tikslūs failų sutapimai

Ši technika skirta failų nustatymui pagal failo maišą (angl. *hash*). Dokumentas yra laikomas saugomu, jei jo maiša atitinka (angl. *hash*) nuodytą konfigūracijoje. Dažnai ši technika dar laikoma kontekstine, o ne turinio analize, nes iš esmės pats turinys nėra analizuojamas. Šis metodas yra geras tuomet, jei failas nėra koreguojamas. Jei failas buvo koreguotas – jis bus neidentifikuojamas, nes jo maiša bus pasikeitusi [13].

1. Sugeneruojama jautraus dokumento maišos funkcija pagal pasirinktą algoritmą (pvz., MD5, SHA1 ar SHA-256 ir t. t.);
2. maišos suma (angl. *checksum*) išsaugoma, failų tikrinimui;
3. failų maišos suma tikrinama su konfidencialių failų maišų sumomis, jei ji atitinka – failas identifikuojamas kaip saugomas (konfidencialus).

2.3.7. Dalinis sutapimas

Ši technika skirta pilnam ar daliniam jautrių duomenų sutapimui identifikuoti [21]. Yra nurodomas visas jautraus failo turinys pagal kurį yra tikrinama/lyginama pagal pilną ar dalinį turinio sutapimą. Užfiksavus atitinkamą dalį informacijos iš saugomų failų, dokumentas fiksuojamas saugomu (konfidencialiu) [24].

Dažniausiai tokia technika remiasi metodika, kuri yra žinoma kaip ciklinė maiša (angl. *cycling hasing*), kur maiša skaičiuojama failo turinio fragmentui, kompensuojant iš anksto nustatytą skaičių simbolių ir tada skaičiuojama kita maiša, ir taip pridėdinėjamos dokumento segmento maišos iki galo [24]. Tikrinami dokumentai yra taip pat praleidžiami per tokią pačią maišos techniką ir tuomet maišo funkcijos yra lyginamos su saugomo (jautraus) dokumento maišos reikšmėmis, jei aptinkama atitinkamų maišos funkcijų, toks dokumentas identifikuojamas kaip slaptas (konfidencialus). Greitesniam dalinių failų sutapimui nustatyti yra sukurtas „Karp-Rabin“ algoritmas paremtas lyginant k -ąsias eilutes [25].

Remiantis apibrėžimu pateiktu Charras C. ir Lecroq T. [26], „Karp-Rabin“ algoritmas remiasi skaičiavimu $hash(x)$. Skaičiavimas gali būti vykdomas pastovioje vietoje $O(m)$ kartų. Paieškos fazėje yra pakankama lyginti $hash(x)$ su $hash(y[j \dots j + m - 1])$, kur $0 \leq j \leq n - m$. Jei atitikimas randamas, nėra būtina tikrinti atitikimo $x = y[j \dots j + m - 1]$ žodis į žodį. Ieškojimo frazė taikant Karp-Rabin algoritmą yra $O = m \times n$, kai pvz., ieškoma a^m iš a^n . Tikėtinas sutapimų skaičius yra $O(m + n)$.

Taigi, realizuojant dalinį sutapimą gali būti įvairių strategijų. Kaip rašo Charras C. ir Lecroq T. [26], strategijos gali būti šios:

- **Absoliutus parinkimas** – remiantis šia strategija maišos funkcijos yra apskaičiuojamos kiekvienam žodžių junginiui su persidengimais. T. y. esant r žodžių tekste: $0, 1, 2, \dots, r$, maiša pvz., skaičiuojama 3 žodžiams yra apskaičiuojama tokioms žodžių aibėms: $[0,1,2], [1,2,3], \dots, [r-2, r-1, r]$.
- **Pozicinis parinkimas** – remiantis šia strategija, maišos funkcijos yra apskaičiuojamos žodžiams, pasirinktose dokumento pozicijose.
- **Atsitiktinis parinkimas** – remiantis šia technika, nustatytam skaičiui žodžių yra apskaičiuojamos maišos funkcijos, tačiau žodžių pozicijos yra parenkamos atsitiktinės.
- **Visas parinkimas** – ši strategija panaši kaip absoliutaus parinkimo tačiau šiuo atveju nėra persidengimo. Pavyzdžiui, jei maišos funkcija skaičiuojama 3 žodžiams, tai remiantis šia strategija, kai dokumentą sudaro r žodžių, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 3, 6, 9, 12, \dots, r-2$. T. y. maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2], [3,4,5], [6,7,8], [9,10,11], [12,13,14], \dots, [r-2, r-1, r]$.
- **Pirmųjų žodžių parinkimas** – strategija panaši, kaip viso parinkimo, tačiau šiuo atveju yra nustatytas teksto ilgis (pirmųjų žodžių skaičius), kuriems bus skaičiuojamos maišos funkcijos. Tarkime tekstą sudaro t žodžių. Imame pirmuosius iš teksto r žodžių ir skaičiuojame maišos funkcijas pvz., 3 žodžiams. Tai taikant šią strategiją, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 3, 6, \dots, 3r$.
- **Pirmųjų žodžių slenkantis parinkimas** – šioje strategijoje taip pat parenkamas nustatyta teksto ilgis (pirmųjų žodžių skaičius), kuriems bus skaičiuojamos maišos funkcijos. Šiuo atveju taikomas persidengimas. Tarkime tekstą sudaro t žodžių. Imame pirmuosius iš teksto r žodžių ir skaičiuojame maišos funkcijas pvz., 3 žodžiams. Tai taikant šią strategiją, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, \dots, r$.
- **Dažnumu pagrįstas parinkimas** – parenkamos rečiausios frazės ir joms skaičiuojamos maišos funkcijos.

- **Retumu dokumente pagrįstas parinkimas** – apskaičiuojamas dokumente naudojamų frazių dažnumas, išrūšiuojamas pagal dažnumą ir parenkant nustatytą skaičių r rečiausių frazių, skaičiuojamos joms maišos funkcijos.
- **K-ojo žodžio sakinyje parinkimas** – tai strategija, kai maišos funkcijos skaičiuojamos žodžiams kiekvienam k -ajam žodžiui sakiniuose.
- **K-ojo sakinio parinkimas** – tai strategija, kai maišos funkcijos skaičiuojamos kiekvienam k -ajam sakiniui;
- **K-osios pastraipos parinkimas** – tai strategija, kai maišos funkcijos skaičiuojamos kiekvienai k -ajai pastraipai.

Analizuojant literatūrą, ieškant informacijos apie naudojamas technikas DLP sistemose realizuojant dalinį sutapimą, pateikiamos informacijos yra gana mažai. Kaip teigia Bretinger F. ir Baggili I. [24], dalinio sutapimo metodas jiems atrodo daugiausiai žadantis, tačiau sunku rasti aktualių publikuojamų tyrimų šia tema. Tikriausiai dėl to, kad dauguma DLP sistemų yra komercinės ir naudoja uždarus, vidinius sprendimus.

Todėl nuspręsta sukonstruoti keletą maišos skaičiavimo dokumento turiniui prototipų, taikant skirtingas strategijas. Juos išbandyti, atlikti eksperimentą ir nustatyti, kuri iš technikų būtų efektyviausia naudoti organizacijos tinkle, dokumentų identifikavimui, stebėjimui ir sekimui.

Taikant vieną iš strategijų, maišos funkcijos yra apskaičiuojamos saugomiems dokumentams, tuomet taikant tas pačias strategijas yra skaičiuojamos maišos funkcijos visiems organizacijos tinkle judantiems dokumentams ir skaičiuojamos maišos funkcijos. Gautos maišos funkcijos, jų aibė, lyginama su saugomų dokumentų maišos funkcijų aibėmis. Todėl svarbu, kad pasirinkta technika, strategija bei maišos algoritmas būtų efektyvus.

Vienos iš dažniausiai naudojamų maišos funkcijų yra MD5, SHA1 ir SHA256. Maišos funkcijos turi būti atsparios kolizijoms ir inversijai. Kaip rašo Black J. [28] ir Henson V. [29], lyginant maišos funkcijas yra svarbiausia atsparumas kolizijoms. Jei du skirtingi žodžių junginiai turės tą pačią apskaičiuotą maišos funkciją, tuomet gali būti klaidų nustatant dokumentų sutapimus. Generuojant atsitiktines žodžių eilutes, kolizijas galime pradėti stebėti po $2^{b/2}$ eilutės, kur b – bitai. Tai reiškia, kad skaičiuojant maišos funkcijas MD4 ar MD5 algoritmais, kolizijų gali būti po 2^{64} eilutės (MD4 ir MD5 maišos funkcijos rezultatas sudarytas iš 128 bitų), o skaičiuojant SHA1 (maišos funkcijos rezultatas sudarytas iš 160 bitų) algoritmu – po 2^{80} eilutės. Tikimybė, ar bus kolizijų priklauso nuo to, kiek bitų sudaro maišos funkcijos algoritmo rezultatą. SHA1 maišos funkcijos rezultatą sudaro 160 bitų, SHA256 – 256 bitai ir SHA512 – 512 bitai. Tačiau kaip teigia Black J. [28] ir Henson V. [29], pakankamai užtenka naudoti 160 bitų maišos funkcijos

algoritmus tokius kaip SHA1 ir RIPEMD-160 lyginant maišos funkcijos. Tikimybė, kad bus atsitiktinė kolizija yra labai maža, $p = 2^{-160}$.

2.3.7.1. Maišos funkcijų palyginimas

Kitas, labai svarbus maišos funkcijų skaičiavimo algoritmo kriterijus yra sparta. 1 lentelėje, pateikiami spartos rezultatai labiausiai paplitusiems kriptografijos algoritmams [32].

1 lentelė. Maišos algoritmų lyginimas [32]

Algoritmo pavadinimas	MB/sekundę	Ciklai per baitą
AES-64	1489	1,2
AES-128	794	2,2
CRC-32	256	6,8
MD5	258	6,8
SHA-1	155	11,3
SHA-256	81	21,5
SHA-512	99	17,6
Tiger 217	217	8,0
Whirlpool 58	58	30,0
RIPEMD-160	108	16,1
RIPEMD-320	111	15,8
RIPEMD-128	155	11,3
RIPEMD-256	159	11,0

Kaip rašo straipsnyje Kavaliūnas K. [32], kuo mažiau megabaitų per sekundę yra apdorojama, tuo daugiau ciklų per baitą yra atliekama. Tokie algoritmai yra sparčiausi. Iš pateiktos lentelės matyti, kad pagal spartą Whirlpool ir SHA256 yra pirmaujantys.

2.3.7.2. Maišos funkcijos algoritmas SHA-256

Kadangi SHA256 algoritmas yra sparčiausias, atsparus kolizijomis ir šiuo metu yra vis labiau populiarėjantis, nuspręsta atlikti platesnę jo analizę ir jį naudoti skaičiuojant maišos funkcijas prototipuose.

Maišos funkcija yra atspari kolizijoms dar nereiškia, kad nėra kolizijų, tiesiog jas labai sunku aptikti.

Mendel F., Nad T. ir Schaffer M. [30;31] keliuose darbuose smulkiai aprašo SHA-256 maišos funkcijos algoritmą [30;31]. SHA-256 algoritmas – tai kartotinė maišos funkcija, kuri apdoroja įvestą tekstą

naudodama t , 512 bitų žinučių blokus m_j . 256 bitų maišos reikšmė skaičiuojama naudojant suspaudimo funkciją f :

$$\begin{aligned} h_0 &= IV \\ h_{j+1} &= f(h_j, m_j) \text{ , kai } 0 \leq j < t \\ hash &= h_t \end{aligned} \quad (4)$$

Žinutės išskaidymas išskaido 512 bitų žinutės bloką į 16 žodžių M_i , $i = 0, \dots, 15$ ir toliau juos išskaido į 64 išskaidytus žinutės žodžius W_i , kaip parodyta žemiau:

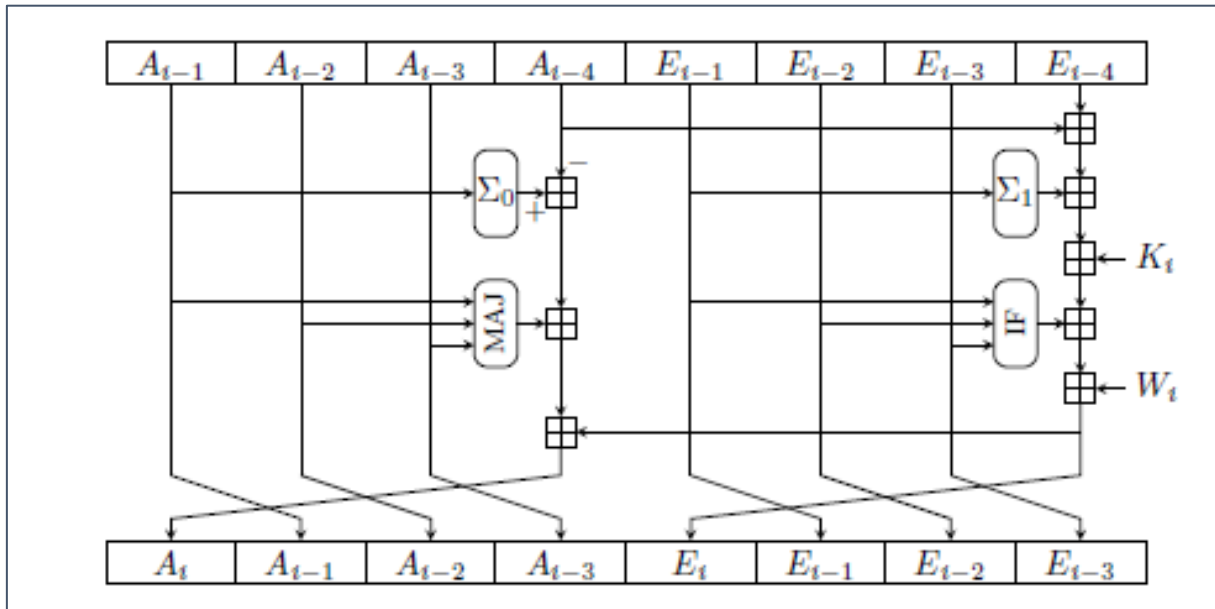
$$W_i = \begin{cases} M_i, & 0 \leq i < 16 \\ \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + (W_{i-16}), & 16 \leq i < 64 \end{cases} \quad (5)$$

Funkcijos $\sigma_0(x)$ ir $\sigma_1(x)$ yra apibrėžtos:

$$\sigma_0(x) = (x \ggg 7) \oplus (x \ggg 18) \oplus (x \ggg 3) \quad (6)$$

$$\sigma_1(x) = (x \ggg 17) \oplus (x \ggg 19) \oplus (x \ggg 10) \quad (7)$$

Toliau pateiktas būsenos atnaujinimo transformatorius aprašymas iš [30], kurio iliustracija pateikta žemiau:



7 pav. Būsenų transformatorius [30]

Būsenos atnaujinimo transformatorius prasideda nuo 256-bitų jungimo reikšmės

$h_j = (A_{i-4}, \dots, A_{-1}, E_{-4}, \dots, E_{-1})$, kurios yra pakeičiamas taikant pažingsnines funkcijas 64 kartus. Žingsnyje $i = 0, \dots, 63$ išplėstas 32-bitų žodis W_i yra panaudojamas skaičiuojant du būsenų kintamuosius E_i ir A_i , kaip nurodyta žemiau:

$$E_i = A_{i-4} + E_{i-4} + \sum_1(E_{i-1}) + IF(E_{i-1}, E_{i-2}, E_{i-3}) + K_i + W_i \quad (8)$$

$$A = E_i + A_{i-4} + \sum_0(A_{i-1}) + MAJ(A_{i-1}, A_{i-2}, A_{i-3}) \quad (9)$$

Loginės funkcijos IF ir MAJ naudojamos kiekviename žingsnyje apibrėžiamos:

$$IF(x, y, z) = xy \oplus xz \oplus z \quad (10)$$

$$MAJ(x, y, z) = xy \oplus yz \oplus xz \quad (11)$$

Tiesinės funkcijos $\sum_0$ ir $\sum_1$ yra apibrėžiamos :

$$\sum_0(x) = (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22) \quad (12)$$

$$\sum_1(x) = (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22) \quad (13)$$

Prieš paskutinį būsenos pakeitimo transformatoriaus žingsnį, prieš tai jungta reikšmė yra pridedama į būsenos pakeitimo atsakymą. Gaunamas rezultatas jungiant reikšmes h_{j+1} , sekančiam žinutės blokui m_{j+1} .

3. SIŪLOMAS DALINIO SUTAPIMO METODAS IR JO REALIZACIJA

Šiame skyriuje aprašoma siūlomas metodas ir tyrimo eiga.

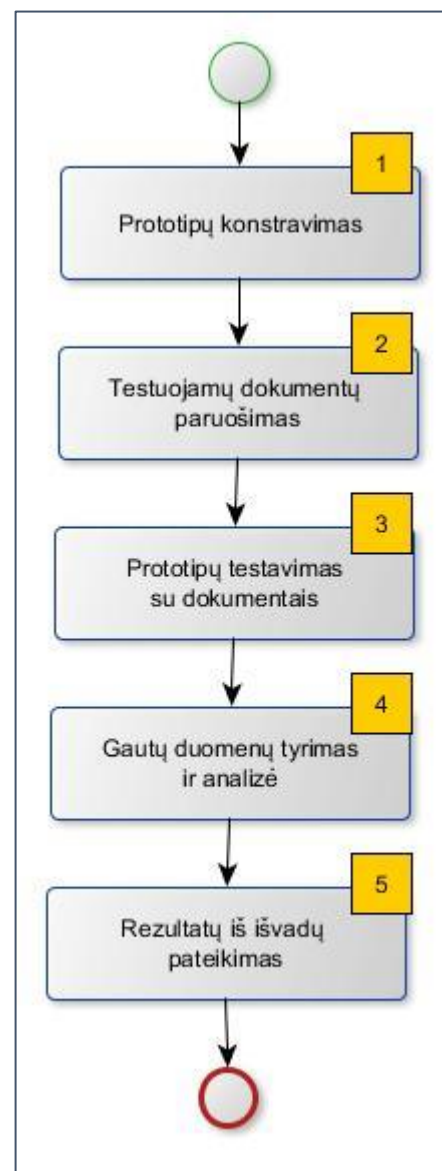
Sutinkant su autorių Bretinger F. ir Baggili I. žodžiais [24], kad dalinio sutapimo turinio analizė technika atrodo daugiausiai žadanti, nuspręsta taikant keletą strategijų skaičiuoti maišos funkcijas dokumentų turiniui ir atlikti tyrimą, kuriuo būtų nustatoma, kuri strategija yra efektyviausia.

3.1. Tyrimo eiga

1. Prototipų konstravimas. Pasirinkta sukonstruoti 6 prototipus, kurie realizuotų skirtingas teksto atrinkimo strategijas tekstui ir maišos funkcijų skaičiavimą. Visuose prototipuose maišos funkcijų skaičiavimui pasirinktas kolizijoms atsparus SHA256 algoritmas.

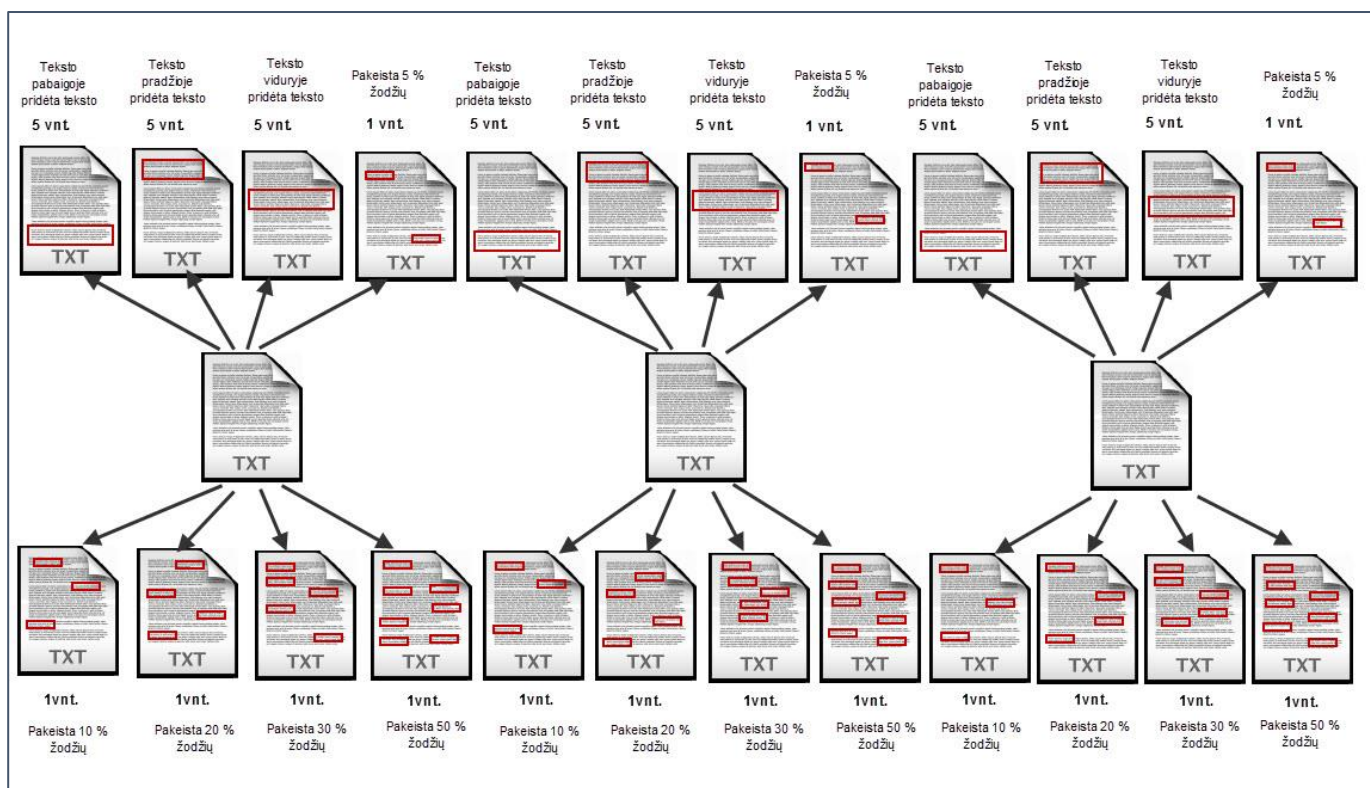
2. Testuojamų dokumentų paruošimas. Dokumentai paruošti TXT formatu. 1 dokumentas sudarytas iš 13 332 žodžių, skirtas prototipų greitaveikai testuoti. Kiti, 60 vnt. dokumentų, skirti turinio atitikimo identifikavimo testavimui, iš kurių :

- 15 dokumentų, kuriuose talpinamas originalus tekstas su įterptu tekstu dokumento pabaigoje;
- 15 dokumentų, kuriuose talpinamas originalus tekstas su įterptu tekstu dokumento pradžioje;
- 15 dokumentų, kuriuose talpinamas originalus tekstas su įterptu tekstu dokumento viduryje;
- 3 dokumentai, kuriuose talpinamas originalus tekstas su pakeistais 5% žodžių;
- 3 dokumentai, kuriuose talpinamas originalus tekstas su pakeistais 10 % žodžių;
- 3 dokumentai, kuriuose talpinamas originalus tekstas su pakeistais 20 % žodžių;
- 3 dokumentai, kuriuose talpinamas originalus tekstas su pakeistais 30 % žodžių;



8 pav. Eksperimento eiga

- 3 dokumentai, kuriuose talpinamas originalus tekstas su pakeistais 50 % žodžių.



9 pav. Parengti dokumentai

2 lentelė. Testiniai dokumentai

Dokumentų pavadinimai	Kiekis	Pridėta / pakeista žodžių	Pastabos
test1.1.	3 vnt.	~ 2 %	Dokumento gale pridėta papildomai ~ 2 % teksto.
test1.2.	3 vnt.	~ 7 %	Dokumento gale pridėta papildomai ~ 7 % teksto.
test1.3.	3 vnt.	~ 11 %	Dokumento gale pridėta papildomai ~ 11 % teksto.
test1.4.	3 vnt.	~ 16 %	Dokumento gale pridėta papildomai ~ 16 % teksto.
test1.5.	3 vnt.	~ 25 %	Dokumento gale pridėta papildomai ~ 25 % teksto.
test1.6.	3 vnt.	~ 2 %	Dokumento pradžioje pridėta papildomai ~ 2 % teksto.
test1.7.	3 vnt.	~ 7 %	Dokumento pradžioje pridėta papildomai ~ 7 % teksto.
test1.8.	3 vnt.	~ 11 %	Dokumento pradžioje pridėta papildomai ~ 11 % teksto.
test1.9.	3 vnt.	~ 16 %	Dokumento pradžioje pridėta papildomai ~ 16 % teksto.

test1.10.	3 vnt.	~ 25 %	Dokumento pradžioje pridėta papildomai ~ 25 % teksto.
test1.11.	3 vnt.	~ 2 %	Dokumento viduryje pridėta papildomai ~ 2 % teksto.
test1.12.	3 vnt.	~ 7 %	Dokumento viduryje pridėta papildomai ~ 7 % teksto.
test1.13.	3 vnt.	~ 11 %	Dokumento viduryje pridėta papildomai ~ 11 % teksto.
test1.14.	3 vnt.	~ 16 %	Dokumento viduryje pridėta papildomai ~ 16 % teksto.
test1.15.	3 vnt.	~ 25 %	Dokumento viduryje pridėta papildomai ~ 25 % teksto.
test1.16.	3 vnt.	~ 5 %	Atsitiktinai pakeista 5 % žodžių.
test1.17.	3 vnt.	~ 10 %	Atsitiktinai pakeista 10 % žodžių.
test1.18.	3 vnt.	~ 20 %	Atsitiktinai pakeista 20 % žodžių.
test1.19.	3 vnt.	~ 30 %	Atsitiktinai pakeista 20 % žodžių.
test1.20	3 vnt.	~ 50 %	Atsitiktinai pakeista 30 % žodžių.

3. Prototipų testavimas su dokumentais. Visi sukurti dokumentai naudojami kiekvieno prototipo testavimui. Kiekvienam dokumentui prototipas apskaičiuoja maišos funkcijas. Gautos maišos funkcijos išsaugomos XLSX formato faile.

Prototipų, realizuojančių skirtingas maišos skaičiavimo strategijas, greitaveika testuojama naudojant didesnę dokumentą, sudarytą iš 13 332 žodžių. Prototipų maišos funkcijų vidutinis skaičiavimo laikas fiksuojamas naudojant „NetBeans IDE 8.2“ funkciją „Profile“.

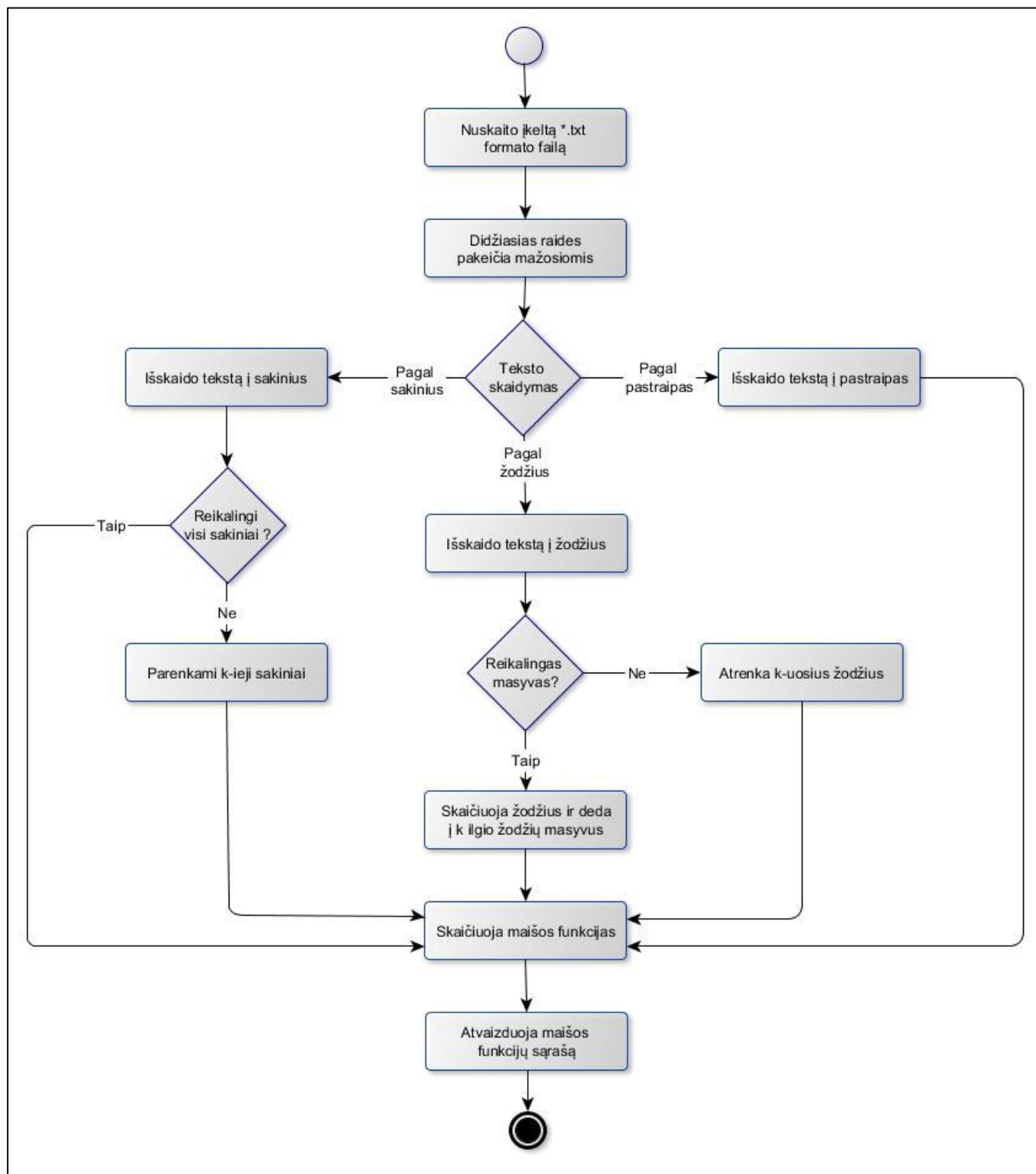
4. Gautų duomenų tyrimas ir analizė. Kiekvieno prototipo ir dokumento maišos funkcijos lyginamos su originalaus dokumento maišos funkcijomis. Gaunamas sutampančių maišos funkcijų skaičius. Vykdomi skaičiavimai ir braižomos diagramos naudojant „SAS studio“ paketą. Realizacija pateikiama skyriuje 7.2 „Maišos funkcijų lyginimas“.

5. Rezultatų ir išvadų pateikimas. Gautų skaičių ir diagramų pateikimas, jų interpretacija ir išvados.

3.2. Maišos funkcijų skaičiavimo prototipai

Pasirinkta konstruoti 6 prototipus JAVA kalba, kurie realizuotų skirtingas maišų skaičiavimo strategijas tekstui. Visuose prototipuose maišos funkcijų skaičiavimui pasirinktas kolizijoms atsparus SHA256 algoritmas.

Žemiau pateikta bendra visų 6 prototipų veiklos diagrama:



Prototipas Nr. 1. Maišos funkcijos skaičiuojamos visam dokumento turiniui, išskaidant jį į pasirinkto ilgio žodžių aibes. Maišos funkcijos skaičiuojamos be persidengimų.

Šios strategijos realizavimui testuojama su skirtingu žodžių skaičiumi aibėse (nuo 1 iki 10):

1. Maišos funkcijos skaičiuojamos kiekvienam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, 3, \dots, r$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0], [1], [2], \dots, [r]$.
2. Maišos funkcijos skaičiuojamos 2 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 2, 4, 6, \dots, r-1$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1], [2,3], [4,5], \dots, [r-1, r]$.
3. Maišos funkcijos skaičiuojamos 3 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 3, 6, 9, \dots, r-2$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2], [3,4,5], [6,7,8], \dots, [r-2, r-1, r]$.
4. Maišos funkcijos skaičiuojamos 4 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 4, 8, 12, \dots, r-3$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3], [4,5,6,7], [8,9,10,11], \dots, [r-3, r-2, r-1, r]$.
5. Maišos funkcijos skaičiuojamos 5 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 5, 10, 15, \dots, r-4$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4], [5,6,7,8,9], [10,11,12,13,14], \dots, [r-4, r-3, r-2, r-1, r]$.
6. Maišos funkcijos skaičiuojamos 6 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 6, 12, 18, \dots, r-5$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4,5], [6,7,8,9,10,11], [12,13,14,15,16,17], \dots, [r-5, r-4, r-3, r-2, r-1, r]$.
7. Maišos funkcijos skaičiuojamos 7 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 7, 14, 21, \dots, r-6$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4,5,6], [7,8,9,10,11,12,13], [14,15,16,17,18,19,20], \dots, [r-6, r-5, r-4, r-3, r-2, r-1, r]$.
8. Maišos funkcijos skaičiuojamos 8 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 8, 16, 24, \dots, r-7$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4,5,6,7], [8,9,10,11,12,13,14,15], [16,17,18,19,20,21,22,23], \dots, [r-7, r-6, r-5, r-4, r-3, r-2, r-1, r]$.
9. Maišos funkcijos skaičiuojamos 9 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 9, 18, 27, \dots, r-8$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4,5,6,7,8], [9,10,11,12,13,14,15,16,17], [18,19,20,21,22,23,24,25,26], \dots, [r-8, r-7, r-6, r-5, r-4, r-3, r-2, r-1, r]$.
10. Maišos funkcijos skaičiuojamos 10 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 10, 20, 30, \dots, r-9$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms:

$[0,1,2,3,4,5,6,7,8,9]$, $[10,11,12,13,14,15,16,17,18,19]$, $[20,21,22,23,24,25,26,27,28,29]$, ..., $[r-9, r-8, r-7, r-6, r-5, r-4, r-3, r-2, r-1, r]$.

Realizacija pateikiama skyriuje 7.1.1 „Prototipas Nr.1“.

Prototipas Nr. 2. Maišos funkcijos skaičiuojamos kiekvienam teksto sakiniui be persidengimų. T. y. esant r ilgio tekstui, maišos funkcijos skaičiuojamos kiekvienam teksto sakiniui.

Realizacija pateikiama skyriuje 7.1.2 „Prototipas Nr.2“.

Prototipas Nr. 3. Maišos funkcijos skaičiuojamos kiekvienai teksto pastraipai. T. y. esant r ilgio tekstui, maišos funkcijos skaičiuojamos kiekvienai teksto pastraipai.

Realizacija pateikiama skyriuje 7.1.3 „Prototipas Nr.3“.

Prototipas Nr. 4. Maišos funkcijos skaičiuojamos visam dokumento turiniui, išskaidant jį į pasirinkto ilgio žodžių aibes. Maišos funkcijos skaičiuojamos **su persidengimais**. Maišos funkcijos skaičiuojamos kiekvienam žodžių junginiui.

Šios strategijos realizavimui testuojama su skirtingu žodžių skaičiumi aibėse (nuo 3 iki 10) :

1. Maišos funkcijos skaičiuojamos 3 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, 3, \dots, r$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2], [1,2,3], [2,3,4], \dots, [r-2, r-1, r]$.
2. Maišos funkcijos skaičiuojamos 4 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, 3, \dots, r$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3], [1,2,3,4], [2,3,4,5], \dots, [r-3, r-2, r-1, r]$.
3. Maišos funkcijos skaičiuojamos 5 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, 3, \dots, r$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4], [1,2,3,4,5], [2,3,4,5,6], \dots, [r-4, r-3, r-2, r-1, r]$.
4. Maišos funkcijos skaičiuojamos 6 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, 3, \dots, r$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4,5], [1,2,3,4,5,6], [2,3,4,5,6,7], \dots, [r-5, r-4, r-3, r-2, r-1, r]$.
5. Maišos funkcijos skaičiuojamos 7 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, 3, \dots, r$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4,5,6], [1,2,3,4,5,6,7], [2,3,4,5,6,7,8], \dots, [r-6, r-5, r-4, r-3, r-2, r-1, r]$.
6. Maišos funkcijos skaičiuojamos 8 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, 3, \dots, r$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4,5,6,7], [1,2,3,4,5,6,7,8], [2,3,4,5,6,7,8,9], \dots, [r-7, r-6, r-5, r-4, r-3, r-2, r-1, r]$.

7. Maišos funkcijos skaičiuojamos 9 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, 3, \dots, r$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4,5,6,7,8]$, $[1,2,3,4,5,6,7,8,9]$, $[2,3,4,5,6,7,8,10]$, ..., $[r-8,r-7,r-6,r-5,r-4,r-3,r-2, r-1,r]$.
8. Maišos funkcijos skaičiuojamos 10 žodžiams. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos nuo šių pozicijų: $0, 1, 2, 3, \dots, r$. Maišos funkcijos skaičiuojamos šioms žodžių aibėms: $[0,1,2,3,4,5,6,7,8,9]$, $[1,2,3,4,5,6,7,8,9,10]$, $[2,3,4,5,6,7,8,9,10,11]$, ..., $[r-9,r-8,r-7,r-6,r-5,r-4,r-3,r-2, r-1,r]$.

Realizacija pateikiama skyriuje 7.1.4 „Prototipas Nr.4“.

Prototipas Nr. 5. Skirtas realizuoti kas k -ojo žodžio parinkimą r ilgio tekste ir maišos funkcijų skaičiavimą. T. y. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas k -ajam žodžiui.

Šios strategijos realizavimui testuojama su skirtingais k -aisiais žodžiais:

1. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 2-ajam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos žodžiams, esantiems šiose pozicijose: $1, 3, 5, \dots, r$.
2. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 3-ajam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos žodžiams, esantiems šiose pozicijose: $2, 5, 8, \dots, r$.
3. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 4-ajam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos žodžiams, esantiems šiose pozicijose: $3, 7, 11, \dots, r$.
4. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 5-ajam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos žodžiams, esantiems šiose pozicijose: $4, 9, 14, \dots, r$.
5. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 6-ajam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos žodžiams, esantiems šiose pozicijose: $5, 11, 17, \dots, r$.
6. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 7-ajam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos žodžiams, esantiems šiose pozicijose: $6, 13, 20, \dots, r$.
7. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 8-ajam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos žodžiams, esantiems šiose pozicijose: $7, 15, 23, \dots, r$.
8. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 9-ajam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos žodžiams, esantiems šiose pozicijose: $8, 17, 26, \dots, r$.
9. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 10-ajam žodžiui. T. y. r ilgio tekstui, maišos funkcijos skaičiuojamos žodžiams, esantiems šiose pozicijose: $9, 19, 29, \dots, r$.

Realizacija pateikiama skyriuje 7.1.5 „Prototipas Nr. 5“.

Prototipas Nr. 6. Skirtas realizuoti kas k -ojo sakinio parinkimą r ilgio tekste ir maišos funkcijų skaičiavimą. T. y. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas k -ajam sakiniui.

Šios strategijos realizavimui testuojama skaičiuojant skirtingiems k -iesiems sakiniams:

1. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 2-ajam sakiniui.
2. r ilgio tekste maišos funkcijos skaičiuojamos kiekvienam kas 3-ajam sakiniui.

Realizacija pateikiama skyriuje 7.1.6 „*Prototipas Nr. 6*“.

4. TAIKYTŲ DALINIO SUTAPIMO STRATEGIJŲ REZULTATAI

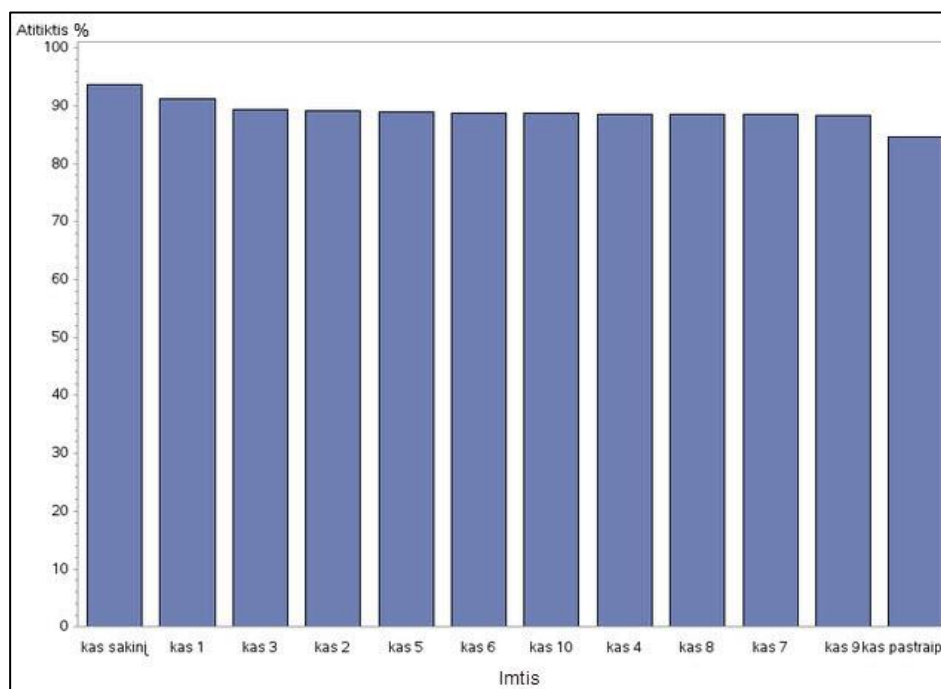
Šiame skyriuje pateikti eksperimentų su prototipais rezultatai. Rezultatai išskaidyti ir apibendrinti pagal atliktus skirtingus eksperimentus, naudojant skirtingai koreguotus dokumentus.

4.1. Eksperimentas, kai pridėtas tekstas dokumento pabaigoje

Tyrimas atliekamas su 15 vnt. dokumentų, kuriuose esanti originalių dokumentų informacija yra papildyta skirtingo ilgio tekstu dokumentų pabaigoje.

4.1.1. Maišos funkcijų skaičiavimas be persidengimų

Pirmiausiai maišos funkcijos skaičiuojamos visam dokumentų turiniui be persidengimų skirtingam kiekiui žodžių, lyginant dokumentų maišos funkcijas su originalaus dokumento maišos funkcijomis. Žemiau pateiktoje diagramoje (žr. 10 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas skirtingam kiekiui žodžių. Eksperimentas atliekamas su prototipais Nr.1, Nr. 2, ir Nr 3.



10 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui

Iš pateiktų rezultatų matome, kad skaičiuojant maišos funkcijas skirtingo kiekio žodžiams, kai dokumento gale yra įterptas tekstas, sutapimo procentas yra panašus ~ 89 %. Tiksliausiai nustatomas dokumentų sutapimas skaičiuojant maišos funkcijas kiekvienam sakiniui (~ 93 %), kiekvienam žodžiui

(~ 91 %) ir 3 žodžiams (~ 89 %). Skaičiuojant maišos funkcijas kiekvienai pastraipai, dokumentų sutapimo atitiktis yra mažiausia ~ 84 %, todėl galima teigti, kad šis metodas efektyvus skaičiuojant maišos funkcijas bet kokiam kiekiui žodžių, kai papildomas tekstas įterptas dokumentų gale.

Toliau pateikiama lentelė, kurioje matyti vidutinis atitikimų procentas, pagal skirtingus dokumentus (pridėtą skirtingą kiekį žodžių).

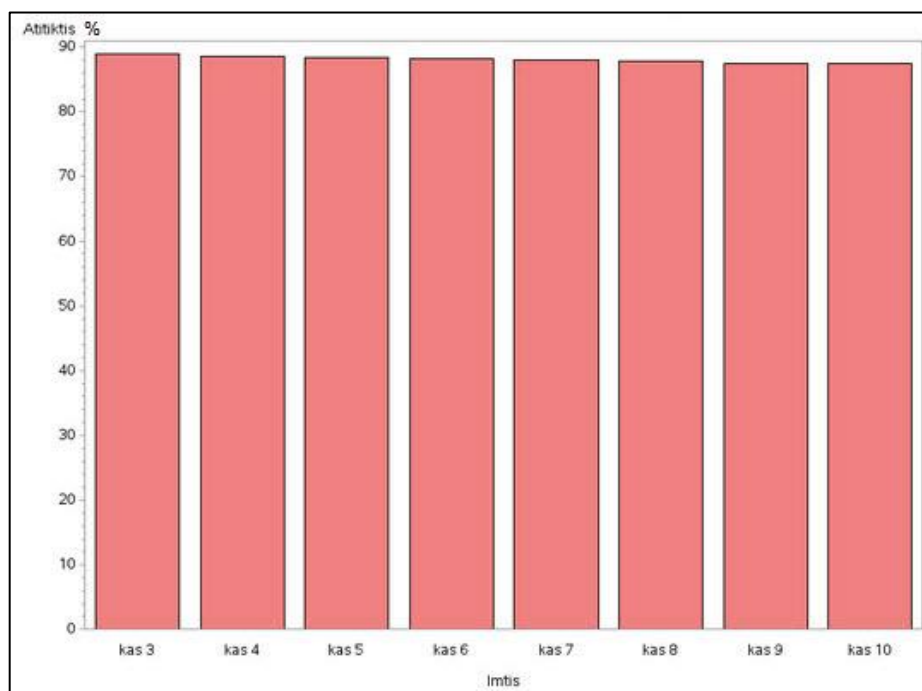
3 lentelė. Atitikimų procentas pagal dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas

	kas1	kas 2	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10	kas Sak	kas Pastr
test 1.1	98,15659	97,3179	97,52588	97,01854	97,01143	96,42857	96,37006	97,11808	95,94181	96,28571	101,5315	91,72932
test 1.2	92,09096	92,67441	93,07808	92,25493	92,77321	92,68363	92,41407	92,54252	91,5787	92,83759	99,27347	89,82456
test 1.3	91,59876	89,63581	89,83137	89,23022	89,58757	89,60819	89,77245	89,00867	89,33333	89,62963	98,44369	84,58333
test 1.4	88,73149	85,79407	85,76315	85,02086	85,49708	85,38308	84,76761	85,23116	85,07844	85,51698	90,76815	79,9253
test 1.5	85,20691	80,39387	80,50957	79,71528	80,1634	79,94118	79,2556	79,27083	79,77899	79,60784	78,3513	76,98413

Iš pateiktų rezultatų matyti, kad atitikimų procentas mažėja, didinant įterptų žodžių kiekį. Skaičiuojant maišos funkcijas be persidengimų, metodas būtų neefektyvus, kai originalaus dokumento gale būtų papildomai pridėta daugiau kaip 50 % teksto.

4.1.2. Maišos funkcijų skaičiavimas su persidengimais

Toliau maišos funkcijos skaičiuojamos visam dokumentų turiniui su persidengimais skirtingam kiekiui žodžių, lyginant dokumentų maišos funkcijas su originalaus dokumento maišos funkcijomis. Žemiau pateiktoje diagramoje (žr. 11 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas skirtingam kiekiui žodžių.



11 pav. Atitikties su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais

Iš gautų rezultatų matyti, kad skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais, kai dokumento gale yra įterptas tekstas, nustatytas vidutinis sutapimo procentas yra panašus ~ 88 %. Tiksliausiai nustatomas dokumentų sutapimas skaičiuojant maišos funkcijas trimis žodžiams (~ 88,49 %).

Galima teigti, kad šis metodas efektyvus skaičiuojant maišos funkcijas bet kokiam kiekiui žodžių, kai papildomas tekstas įterptas dokumentų gale.

Toliau pateikiama lentelė, kurioje matyti vidutinis atitikimų procentas, pagal skirtingus dokumentus (pridėtą skirtingą kiekį žodžių).

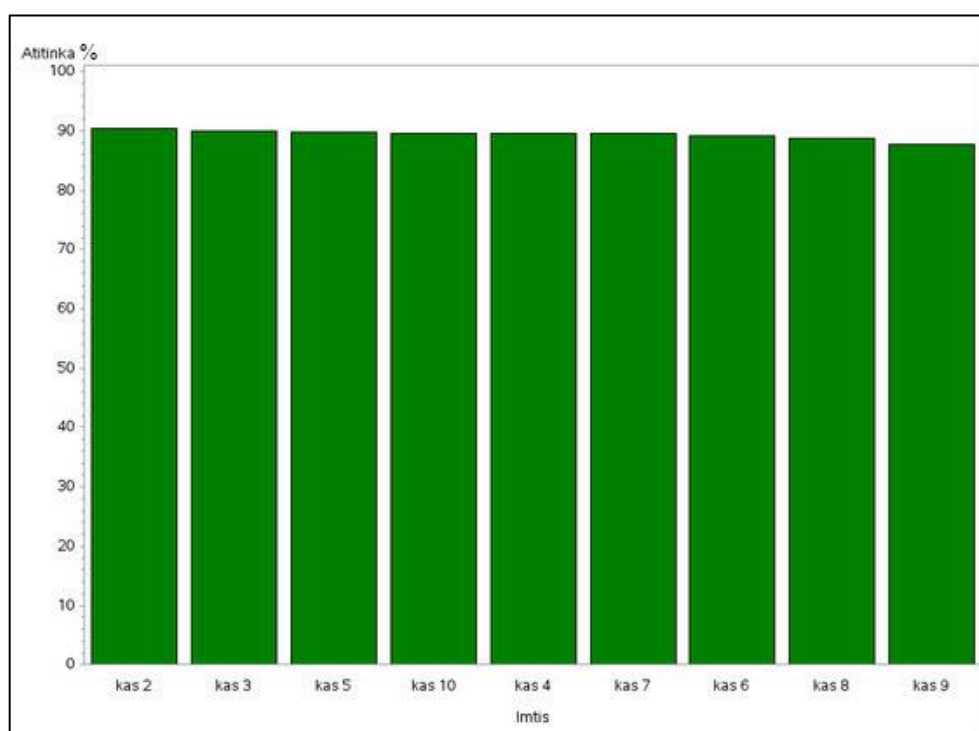
4 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas su persidengimais

	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10
test 1.1	97,00312	96,77774	96,55237	96,327	96,10162	95,87625	95,65087	95,50468
test 1.2	92,45306	92,23843	92,0238	91,80917	91,59454	91,37991	91,16528	91,02539
test 1.3	89,29529	89,08798	88,88066	88,67335	88,46603	88,25872	88,0514	87,91735
test 1.4	85,28466	85,01674	84,81885	84,62096	84,42306	84,22517	84,02728	83,89898
test 1.5	80,19881	79,81817	79,56774	79,31731	79,13199	78,94666	78,76133	78,64086

Iš pateiktų rezultatų matyti, kad vidutinis atitikimų procentas mažėja, didinant įterptų žodžių kiekį. Skaiciuojant maišos funkcijas be persidegimų, metodas būtų neefektyvus, kai originalaus dokumento gale būtų papildomai pridėta daugiau kaip 50 % žodžių.

4.1.3. Maišos funkcijų skaičiavimas kas k -iesiems žodžiams

Maišos funkcijos skaičiuojamos ne visam dokumento turiniui, o tik kas k -ajam žodžiui. Žemiau pateiktoje diagramoje (žr. 12 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas imant skirtingus k -uosius žodžius. Eksperimentas atliekamas su prototipu Nr. 5.



12pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems žodžiams tekste

Iš pateiktų rezultatų matome, kad skaičiuojant maišos funkcijas skirtingiems k -iesiems žodžiams, kai dokumento gale yra įterptas tekstas, apskaičiuotas vidutinis sutapimo procentas yra panašus ~ 89 % . Tiksliausiai nustatomas dokumentų sutapimas skaičiuojant maišos funkcijas kas antram žodžiui (nustatyta vidutinė atitiktis ~ 90,47 %).

Galima teigti, kad šis metodas efektyvus skaičiuojant maišos funkcijas bet kuriam k -ajam žodžiui, kai papildomas tekstas įterptas dokumentų gale.

Toliau pateikiama lentelė, kurioje matyti vidutinis atitikimų procentas, pagal skirtingus dokumentus (pridėtą skirtingą kiekį žodžių).

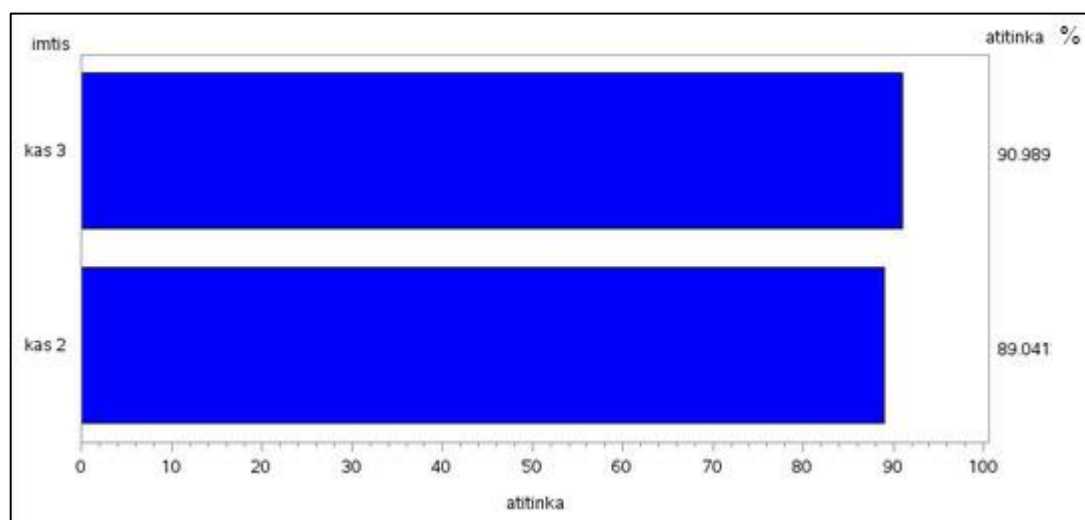
5 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius žodžius skaičiuojant maišos funkcijas

	kas 2	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10
test 1.1	97,92334	97,66941	97,91774	97,6695	97,67154	97,26747	98,05246	96,41026	97,68699
test 1.2	93,12148	93,91345	92,27804	93,30269	92,40526	93,50168	92,22222	91,64872	93,37039
test 1.3	91,25826	90,71138	90,72032	90,35548	90,46647	90,6265	88,97222	89,72391	90,66609
test 1.4	87,19282	85,77686	85,9928	86,71133	84,98651	85,6679	85,69105	82,97672	86,39157
test 1.5	82,8478	81,42451	80,66015	81,36486	80,36473	80,4386	79,27632	78,27106	80,01961

Iš pateiktų rezultatų matyti, kad atitikimų procentas mažėja, didinant įterptų žodžių kiekį. Skaičiuojant maišos funkcijas k -siems žodžiams, metodas būtų neefektyvus, kai originalaus dokumento gale būtų papildomai pridėta daugiau kaip 50 % teksto.

4.1.4. Maišos funkcijų skaičiavimas kas k -iesiems sakiniams

Maišos funkcijos skaičiuojamos ne visam dokumento turiniui, o tik kas k -ajam sakiniui (pasirinkta skaičiuoti kas antram ir kas trečiam sakiniui). Žemiau pateiktoje diagramoje (žr. 13 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas kas antram ir kas trečiam sakiniui. Eksperimentas atliekamas su prototipu Nr.6.



13 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas antram ir trečiam sakiniui tekste

Iš pateiktų rezultatų matome, kad skaičiuojant maišos funkcijas kas antram ir kas trečiam sakiniui skiriasi. Skaičiuojant maišos funkcijas kas antram sakiniui, vidutinis sutapimo procentas yra ~ 90,99 %, o skaičiuojant kas trečiam sakiniui ~ 89 % .

Šis metodas efektyvus skaičiuojant maišos funkcijas dokumentams, kurių gale yra įterpta papildomo teksto.

Toliau pateikiama lentelė, kurioje matyti vidutinis atitikimų procentas, pagal dokumentuose įterptų žodžių skaičių.

6 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius sakinius skaičiuojant maišos funkcijas

	kas 2	kas 3
test 1.1	93,78307	97,61905
test 1.2	92,12615	93,45238
test 1.3	93,46093	93,45238
test 1.4	88,15789	88,35859
test 1.5	77,67857	82,06349

Iš pateiktų rezultatų matyti, kad atitikimų skaičius mažėja, didinant įterptų žodžių kiekį. Skaičiuojant maišos funkcijas be persidėgimų būtų neefektyvų, kai originalaus dokumento gale būtų papildomai pridėta 50 % žodžių.

4.1.5. Išvados

1. Vykdamas eksperimentą su dokumentais, kurių pabaigoje buvo įterptas tekstas, nustatyta, kad tiksliausiai dokumentų sutapimas identifiкуotas skaičiuojant maišos funkcijas kiekvienam sakiniui. Taikant šią strategiją nustatytas ~ 93 % vidutinis dokumentų sutapimas. Mažiausias sutapimas nustatytas skaičiuojant maišos funkcijas kiekvienai pastraipai (~ 84 %).

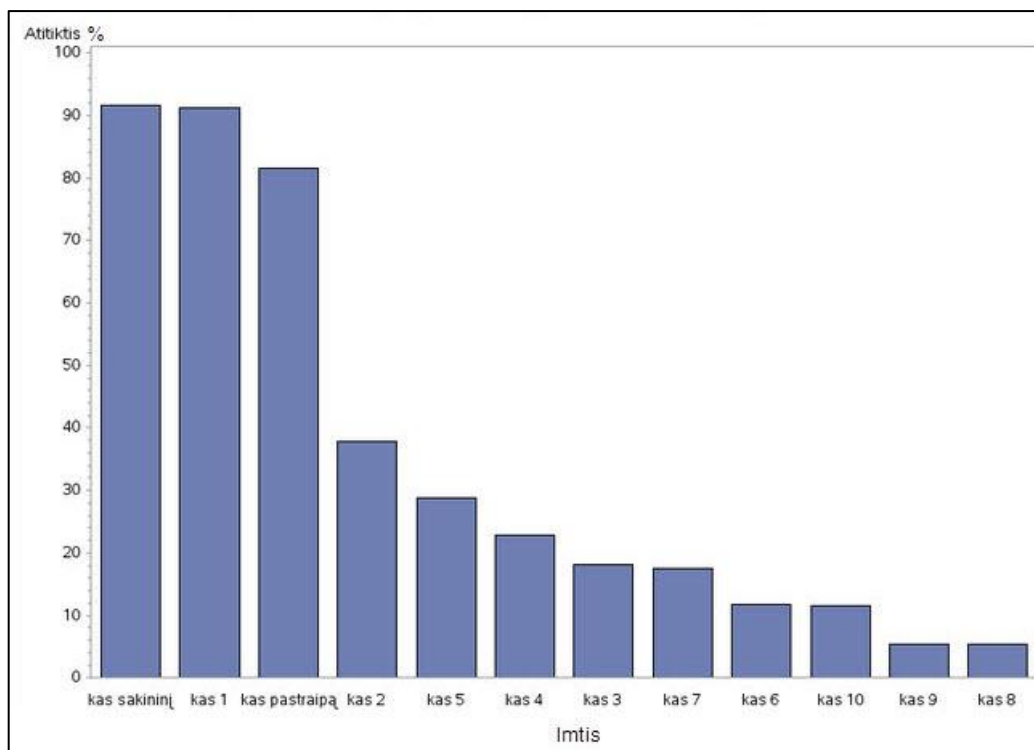
2. Taikant kitas maišos funkcijų skaičiavimo strategijas, reikšmingo skirtumo nenustatyta. Visos metodo strategijos yra efektyvios nustatant dokumentų sutapimą, kai tekstas yra įterptas dokumentų pabaigoje, todėl geriau naudoti tas, kurių geresnė greitisveika.

4.2. Eksperimentas, kai pridėtas tekstas dokumento pradžioje

Eksperimentas atliekamas su 15 vnt. dokumentų, kuriuose esanti originalių dokumentų informacija yra papildyta skirtingo ilgio tekstu dokumento pradžioje.

4.2.1. Maišos funkcijų skaičiavimas be persidengimų

Pirmiausiai maišos funkcijos skaičiuojamos visam dokumentų turiniui be persidengimų skirtingam kiekiui žodžių, lyginant dokumentų maišos funkcijas su originalaus dokumento maišos funkcijomis. Žemiau pateiktoje diagramoje (žr. 14 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas skirtingam kiekiui žodžių. Eksperimentas atliekamas su prototipais Nr.1, Nr. 2, ir Nr 3.



14 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui

Iš pateiktų rezultatų matyti, kad skaičiuojant maišos funkcijas skirtingo kiekio žodžiams, kai dokumento pradžioje yra įterptas tekstas, nustatytas sutapimo procentas skirtingas. Tiksliausiai nustatoma dokumentų atitiktis skaičiuojant maišos funkcijas kiekvienam sakiniui (nustatyta vidutinė atitiktis su originaliu dokumentu ~ 91,5 %), kiekvienam žodžiui (~ 91,1 %) ir pastraipoms (~ 81,5 %).

Skaičiuojant maišos funkcijas nuo 2 iki 10 žodžių, metodas neefektyvus, nustatomas sutapimas mažiau nei 50 %.

Toliau pateikiama lentelė, kurioje matyti vidutinis atitikimų procentas, pagal dokumentuose įterptų žodžių skaičių.

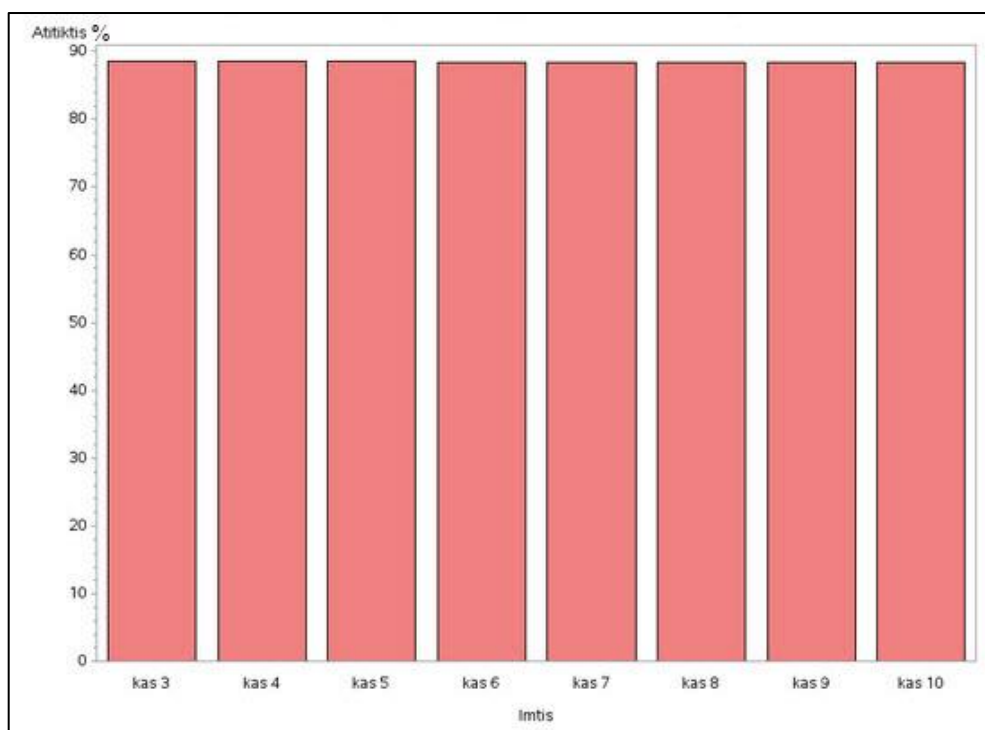
7 lentelė. Atitikimų procentas pagal dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas

	kas1	kas 2	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10	kas Sak	kas Pastr
test 1.6	98,14329	65,77284	32,80021	32,03883	28,7234	31,88406	0	0	0	31,00775	100,2102	87,1345
test 1.7	94,1111	33,96823	0,416796	0	30,33708	0	30,15873	0	0	0	98,02383	89,35673
test 1.8	89,54812	29,13998	0,200803	27,15054	28,93773	0	30,30303	0	0	0	93,19511	78,14815
test 1.9	88,83537	30,20479	29,15701	28,29132	28,94737	0	0	0	0	0	89,52314	77,59104
test 1.10	85,08791	30,00879	27,7232	27,11111	27,54902	27	26,74419	26,66667	27,27273	26,66667	76,72528	75,13228

Iš pateiktų rezultatų matyti, kad taikant techniką skaičiuojant maišos funkcijas 2 – 10 žodžių be persidengimų, nenustatomas sutapimas net kai yra įterpta tik ~ 2 % papildomo teksto.

4.2.2. Maišos funkcijų skaičiavimas su persidengimais

Maišos funkcijos skaičiuojamos visam dokumentų turiniui su persidengimais skirtingam kiekiui žodžių, lyginant dokumentų maišos funkcijas su originalaus dokumento maišos funkcijomis. Žemiau pateiktoje diagramoje (žr. 15 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas skirtingam kiekiui žodžių su persidengimais. Eksperimentas atliekamas su prototipu Nr. 4.



15 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais

Patektuose rezultatuose matyti, kad skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais, kai dokumento pradžioje įterptas tekstas, nustatytas sutapimo procentas yra taip pat panašus visais atvejais $\sim 88,5\%$. Tiksliausiai nustatytas dokumentų sutapimas skaičiuojant maišos funkcijas trimis žodžiams (nustatytas sutapimas su originaliais dokumentais $\sim 88,56\%$).

Galima teigti, kad šis metodas efektyvus skaičiuojant maišos funkcijas bet kokiam kiekiui žodžių su persidengimais, kai papildomas tekstas įterptas dokumentų pradžioje.

Toliau pateikiama lentelė, kurioje matyti vidutinis atitikimų skaičius, pagal dokumentuose įterptų žodžių skaičių.

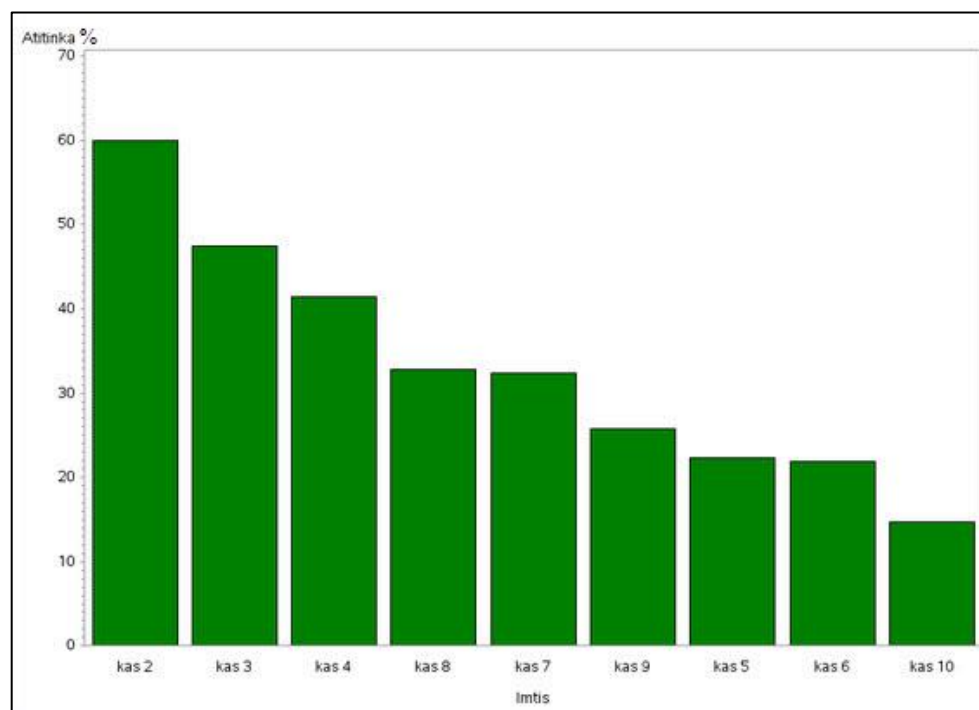
8 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas su persidengimais

	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10
test 1.6	97,29437	97,36024	97,36024	97,36024	97,36024	97,36024	97,36024	97,36024
test 1.7	92,7313	92,79492	92,79492	92,79492	92,79492	92,79492	92,79492	92,79492
test 1.8	86,78176	86,77623	86,70943	86,64263	86,57583	86,50903	86,44223	86,44223
test 1.9	85,64184	85,63029	85,63029	85,63029	85,63029	85,63029	85,63029	85,63029
test 1.10	80,32877	80,07808	79,95761	79,83713	79,78176	79,72639	79,67102	79,61565

Iš pateiktų rezultatų matyti, kad atitikimų procentas mažėja, didinant įterptų žodžių kiekį. Skaičiuojant maišos funkcijas su persidengimais, metodas būtų neefektyvus, kai originalaus dokumento priekyje būtų papildomai pridėta 50 % teksto.

4.2.3. Maišos funkcijų skaičiavimas kas k -iesiems žodžiams

Maišos funkcijos skaičiuojamos ne visam dokumento turiniui, o tik kas k -ajam žodžiui. Žemiau pateiktoje diagramoje (žr. 16 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas imant skirtingus k -uosius žodžius. Eksperimentas atliekamas su prototipu Nr. 5.



16 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems žodžiams tekste

Iš pateiktų rezultatų matome, kad skaičiuojant maišos funkcijas skirtingam k -ajam žodžiui, kai dokumento priekyje yra įterptas tekstas, nustatytas vidutinis sutapimo procentas yra skirtingas. Tiksliausiai nustatomas dokumentų sutapimas skaičiuojant maišos funkcijas kas antram žodžiui, nustatyta vidutinė atitiktis $\sim 60\%$.

Galime teigi, kad metodas neefektyvus, nustatomas per mažas atitikimo procentas.

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų procentas, pagal dokumentuose įterptų žodžių skaičių.

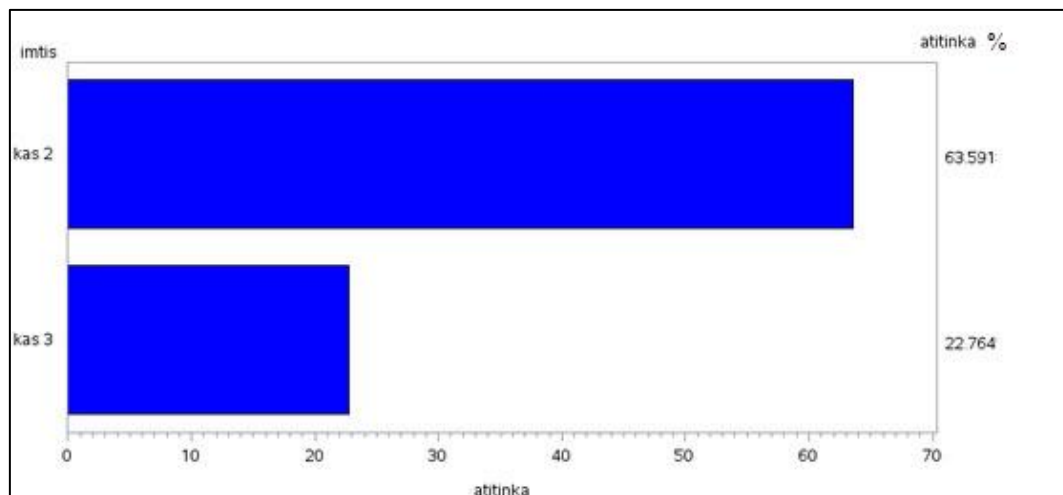
9 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius žodžius skaičiuojant maišos funkcijas

	kas 2	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10
test 1.6	51,78568	21,64996	20,02565	12,69231	15,03993	43,48204	17,06523	11,49541	5,847357
test 1.7	48,85113	70,21242	41,26231	11,57494	12,41295	8,003848	11,10035	14,20654	9,103457
test 1.8	48,24594	43,37815	41,16172	37,39334	14,01513	65,71347	38,69053	34,78134	11,62281
test 1.9	67,43827	62,59004	42,63001	14,03792	34,13006	10,31417	38,70755	37,05637	12,32838
test 1.10	83,71343	39,65926	61,87277	36,26593	33,84352	34,1876	58,26136	31,5499	34,84196

Iš pateiktų rezultatų matyti, kad skaičiuojant maišos funkcijas kas 2 – 4 žodį, pridėjus papildomo teksto, nustatytas atitikimo procentas didesnis. Skaičiuojant maišos funkcijas kas 5 – 10 žodį, sutapimas nenustatomas.

4.2.4. Maišos funkcijų skaičiavimas kas k -iesiems sakiniams

Maišos funkcijos skaičiuojamos ne visam dokumento turiniui, o tik kas k -ajam sakiniui (kas antram ir kas trečiam sakiniui). Žemiau pateiktoje diagramoje (žr. 17 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas kas antram ir kas trečiam sakiniui. Eksperimentas atliekamas su prototipu Nr. 6.



17 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems sakiniams tekste

Iš pateiktų rezultatų matyti, kad skaičiuojant maišos funkcijas kas antram ir kas trečiam sakiniui reikšmingai skiriasi. Skaičiuojant maišos funkcijas kas antram sakiniui, vidutinis sutapimo procentas yra ~ 63,6 %, o skaičiuojant kas trečiam sakiniui tik ~ 22,8 % .

Galime teigti, kad šis metodas yra neefektyvus, kai dokumento pradžioje yra įterptas papildomas tekstas. Nustatomas per mažas atitikimo procentas.

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų procentas, pagal dokumentuose įterptų žodžių skaičių.

10 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius sakinius skaičiuojant maišos funkcijas

	kas 2	kas 3
test 1.6	62,30159	0
test 1.7	61,57059	29,16667
test 1.8	88,33272	53,7037
test 1.9	57,60234	0
test 1.10	48,14815	30,95238

Iš pateiktų rezultatų matome, kad didinant įterpto teksto kiekį, skaičiuojant maišos funkcijas kas 2 sakinius, sutapimas nenustatomas pridėjus $\sim 16\%$ papildomo teksto ir daugiau. Skaičiuojant maišos funkcijas kas 3 sakinius, sutapimas nenustatomas net pridėjus tik $\sim 2\%$ žodžių (1 sakinį).

4.2.5. Išvados

1. Vykdant eksperimentą su dokumentais, kurių pradžioje buvo įterptas tekstas, nustatyta, kad tiksliausiai dokumentų sutapimas identifikuotas skaičiuojant maišos funkcijas kiekvienam sakiniui, nustatytas sutapimas $\sim 91,5\%$.

2. Skaičiuojant maišos funkcijas skirtingiems žodžių kiekiams su persidengimais, reikšmingo skirtumo tarp pasirinkto skaičiaus žodžių nenustatyta. Visos metodų strategijos yra efektyvios nustatant dokumentų sutapimą, kai tekstas yra įterptas dokumentų pradžioje, todėl geriau naudoti tas, kurių geresnė greitisveika.

3. Skaičiuojant maišos funkcijas be persidengimų nuo 2 iki 10 žodžių, metodas neefektyvus, nustatomas sutapimas mažesnis nei 60% .

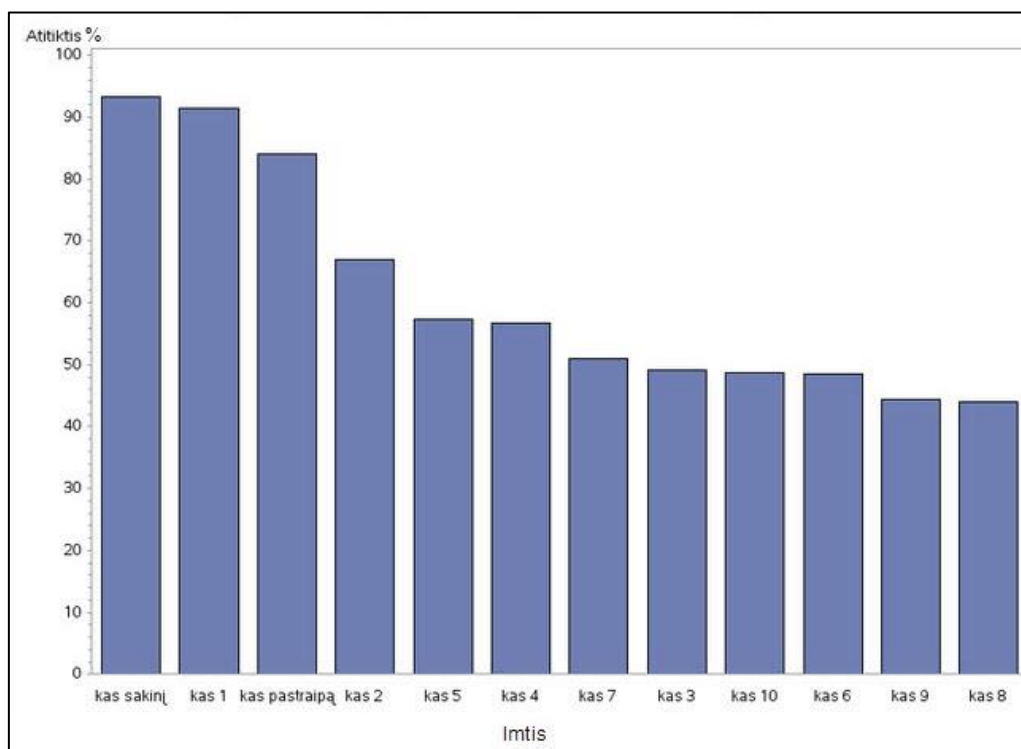
4. Skaičiuojant maišos funkcijas skirtingiems k -iesiems žodžiams ar sakiniams, metodas neefektyvus, nustatytas per mažas sutapimų procentas.

4.3. Eksperimentas, kai pridėtas tekstas dokumento viduryje

Tyrimas atliekamas su 15 vnt. dokumentų, kuriuose esanti originalių dokumentų informacija yra papildyta skirtingo ilgio tekstu dokumento viduryje.

4.3.1. Maišos funkcijų skaičiavimas be persidengimų

Pirmiausiai maišos funkcijos skaičiuojamos visam dokumentų turiniui be persidengimų skirtingam kiekiui žodžių, lyginant dokumentų maišos funkcijas su originalaus dokumento maišos funkcijomis. Žemiau pateiktoje diagramoje (žr. 18 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas skirtingam kiekiui žodžių. Eksperimentas atliekamas su prototipais Nr. 1, Nr. 2 ir Nr. 3.



18 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui

Iš diagramoje pateiktų duomenų matyti, kad skaičiuojant maišos funkcijas skirtingo kiekio žodžiams, kai dokumento viduryje yra įterptas tekstas, nustatymo sutapimo procentas skirtingas. Tiksliausiai nustatoma dokumentų atitiktis skaičiuojant maišos funkcijas kiekvienam sakiniui (~ 93,4 %), kiekvienam žodžiui (~ 91,5 %) ir kas pastraipą (~ 84 %). Skaičiuojant maišos funkcijas 9 ar 8 žodžiams nustatyta vidutinė atitiktis yra mažiausia ~ 44 % .

Metodas yra neefektyvus skaičiuojant maišos funkcijas 3 – 10 žodžių, nes nustatoma mažesnė už 60 % atitiktis.

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų procentas, pagal dokumentų viduryje įterptų žodžių skaičių.

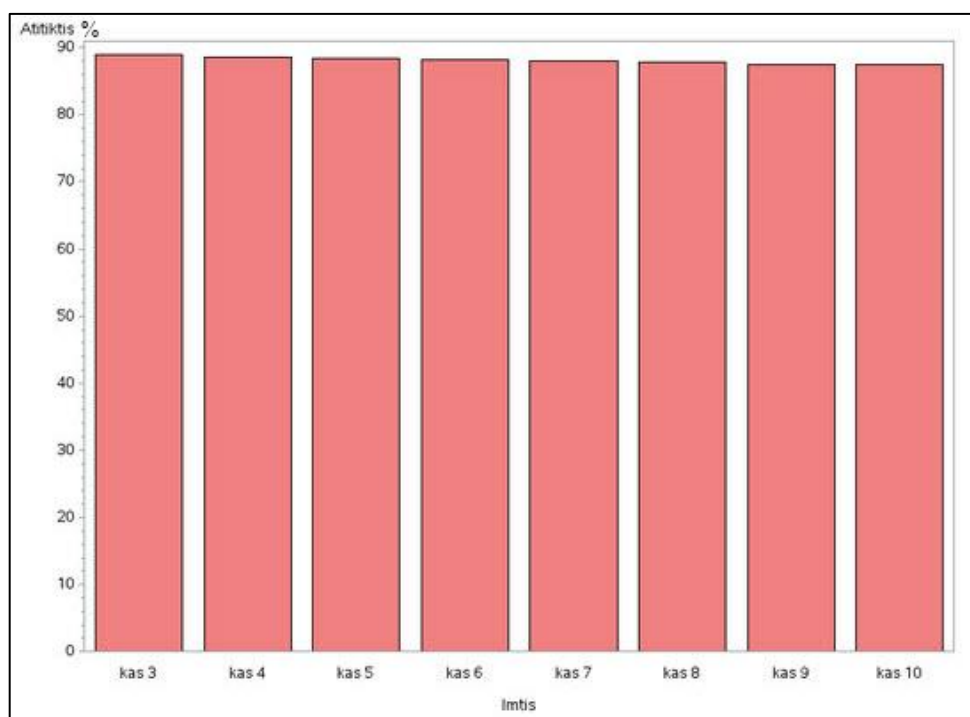
11 lentelė. Atitikimų procentas pagal dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas

	kas1	kas 2	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10	kas Sak	kas Pastr
test 1.11	98,07854	91,49152	67,76052	66,80074	72,66851	66,64596	48,38355	47,98475	48,05901	72,36082	101,5315	89,34837
test 1.12	94,04188	66,67139	47,33716	46,44879	69,06755	46,93246	68,82482	45,96996	46,07035	45,75405	99,22342	91,53439
test 1.13	91,53129	68,0794	45,38062	66,63132	61,37905	45,16374	49,95005	44,0384	44,55556	44,07407	97,47167	84,09357
test 1.14	88,67002	65,13618	43,50187	63,41306	42,74854	42,98912	47,59138	42,21373	42,47798	41,95595	89,82684	77,08333
test 1.15	85,20801	43,65066	41,47603	40,45139	40,84967	40,76471	40,35786	39,93056	40,54929	39,86928	78,79307	77,64706

Iš lentelėje pateiktų rezultatų matyti, kad skaičiuojant maišos funkcijas 7 – 9 žodžiams, net įterpus tik ~ 2 % papildomo teksto, nenustatomas sutapimas didesnis nei 60 %. Patikimiausiai apskaičiuojamas maišos funkcijų sutapimas (tikėtina pridėjus iki 50 % papildomo teksto), skaičiuojant maišos funkcijas kiekvienam žodžiui, sakiniams ir pastraipoms.

4.3.2. Maišos funkcijų skaičiavimas su persidengimais

Maišos funkcijos skaičiuojamos visam dokumentų turiniui su persidengimais skirtingam kiekiui žodžių, lyginant dokumentų maišos funkcijas su originalaus dokumento maišos funkcijomis. Žemiau pateiktoje diagramoje (žr. 19 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas skirtingam kiekiui žodžių. Eksperimentas atliekamas su prototipu Nr. 4.



19 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais

Iš pateiktų rezultatų matome, kad skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais, kai dokumento viduryje įterptas tekstas, nustatytas vidutinis sutapimo procentas yra panašus ~ 88,1 %. Tiksliausiai nustatytas dokumentų sutapimas skaičiuojant maišos funkcijas trims žodžiams (~ 88,86 %).

Galima teigti, kad šis metodas efektyvus skaičiuojant maišos funkcijas bet kokiam kiekiui žodžių su persidengimais, kai papildomas tekstas įterptas dokumentų viduryje.

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų procentas, pagal dokumentų viduryje įterptų žodžių skaičių.

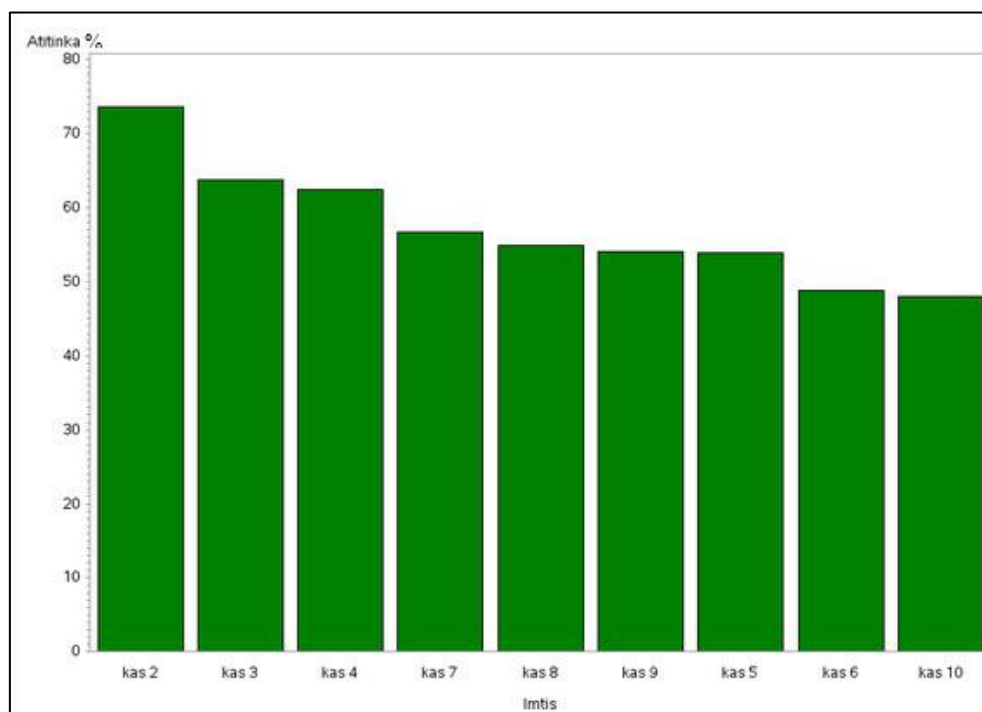
12 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas su persidengimais

	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10
test 1.11	97,08229	96,85692	96,63155	96,40617	96,1808	95,95542	95,73005	95,58385
test 1.12	92,45306	92,23843	92,0238	91,80917	91,59454	91,37991	91,16528	91,02539
test 1.13	89,29529	89,08798	88,88066	88,67335	88,46603	88,25872	88,0514	87,91735
test 1.14	85,22537	84,95759	84,75984	84,5621	84,36435	84,1666	83,96885	83,84055
test 1.15	80,25418	79,87354	79,62311	79,37268	79,18736	79,00203	78,81671	78,69623

Iš pateiktų rezultatų matyti, kad atitikimų skaičius mažėja, didinant įterptų žodžių kiekį. Skaičiuojant maišos funkcijas be persidengimų, metodas būtų neefektyvus, kai originalaus dokumento viduryje būtų papildomai pridėta daugiau kaip 50 % žodžių.

4.3.3. Maišos funkcijų skaičiavimas kas k -iesiems žodžiams

Maišos funkcijos skaičiuojamos tik kas k -ajam žodžiui tekste. Žemiau pateiktoje diagramoje (žr. 20 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas imant skirtingus k -uosius žodžius. Eksperimentas atliekamas su prototipu Nr. 5.



20 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas tik kas k -iesiems žodžiams tekste

Iš pateiktos diagramos matome, kad skaičiuojant maišos funkcijas skirtingam k -ajam žodžiui, kai dokumento viduryje yra įterptas tekstas, nustatytas sutapimo procentas yra skirtingas, vidurkis $\sim 57,4 \%$. Tiksliausiai nustatomas dokumentų sutapimas skaičiuojant maišos funkcijas kas antram žodžiui, nustatytas sutapimas $\sim 73,6 \%$, kas trečiam žodžiui ($\sim 63,7 \%$) ir kas ketvirtam žodžiui ($\sim 62,5 \%$).

Metodas neefektyvus skaičiuojant maišos funkcijas kas 5 – 10 žodžiui, sutapimas yra arti 50 %, todėl ar dokumentai sutampa yra tik spėjimas.

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų procentas, pagal dokumentų viduryje įterptų žodžių skaičių.

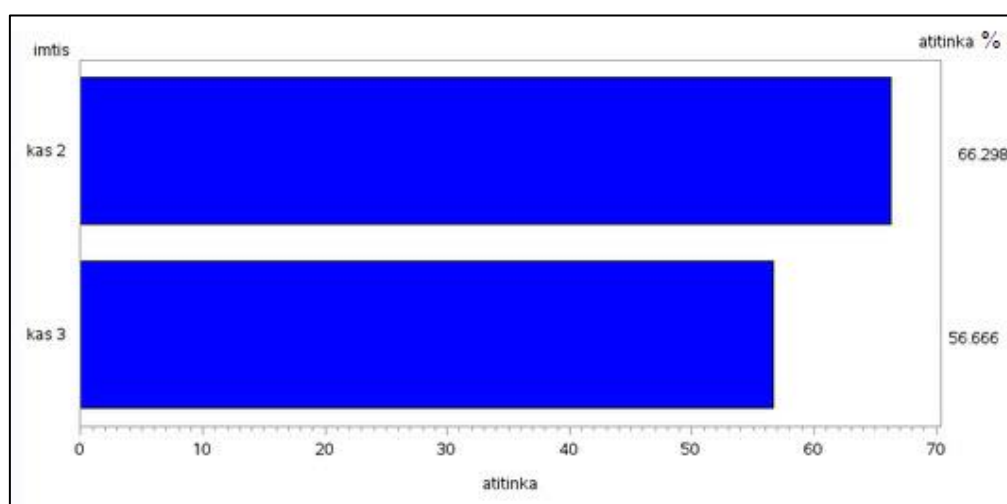
13 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius žodžius skaičiuojant maišos funkcijas

	kas 2	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10
test 1.11	76,52535	57,61368	59,68412	54,28321	53,15683	70,42832	56,5972	51,45438	52,16035
test 1.12	76,21433	75,31124	73,79329	52,13836	51,50253	49,79203	48,51852	51,42196	51,74726
test 1.13	61,7186	68,64542	58,58283	69,54996	49,6788	71,07176	55,43966	62,23891	47,25191
test 1.14	72,93387	68,65304	53,2015	48,06433	46,22222	44,93656	49,36802	64,61224	43,98875
test 1.15	80,40285	48,46033	67,34513	45,94471	43,5323	47,09183	64,39959	40,33523	44,79012

Geriausiai atitikimas nustatomas skaičiuojant maišos funkcijas kas antram žodžiui. Skaičiuojant maišos funkcijas kas 3 – 6 ir 8 – 10 žodžiams, sutapimas nenustatomas net pridėjus tik ~ 2 % papildomo teksto.

4.3.4. Maišos funkcijų skaičiavimas kas k -iesiems sakiniams

Maišos funkcijos skaičiuojamos kas k -ajam sakiniui (kas antram ir kas trečiam sakiniui) tekste. Žemiau pateiktoje diagramoje (žr. 21 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas kas antram ir kas trečiam sakiniui. Eksperimentas atliekamas su prototipu Nr. 6.



21 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems sakiniams tekste

Iš pateiktų rezultatų matome, kad skaičiuojant maišos funkcijas kas antram ir kas trečiam sakiniui skiriasi. Skaičiuojant maišos funkcijas kas antram sakiniui, vidutinis sutapimo procentas yra ~ 66,3 %, o skaičiuojant kas trečiam sakiniui ~ 56,7 % . Abejais atvejais ar dokumentai sutampa yra beveik tik spėjimas, todėl šis metodas, kai dokumento viduryje yra įterptas tekstas, yra neefektyvus.

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų procentas, pagal dokumentų viduryje įterptų žodžių skaičių.

14 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius sakinius skaičiuojant maišos funkcijas

	kas 2	kas 3
test 1.11	45,79832	48,69048
test 1.12	89,81481	48,40715
test 1.13	65,66416	67,63792

test 1.14	82,69841	63,25758
test 1.15	47,51674	55,34091

Metodas pakankamai efektyvus skaičiuojant maišos funkcijas kas 2 sakiniui, kai teksto viduryje yra įterpta papildomai 7 – 16 % teksto.

4.3.5. Išvados

1. Vykdamas eksperimentą su dokumentais, kurių viduryje įterptas tekstas, nustatyta, kad tiksliausiai dokumentų sutapimas identifiкуotas skaičiuojant maišos funkcijas kiekvienam sakiniui, nustatytas sutapimas ~ 93,4 % , toliau kiekvienam žodžiui (~ 91,5 %) ir pastraipoms (~ 84 %).

2. Skaičiuojant maišos funkcijas skirtingiems žodžių kiekiams su persidengimais, reikšmino skirtumo tarp pasirinkto skaičiaus žodžių nenustatyta, metodas efektyvus, todėl taikant šią techniką geriau atsižvelgti į greitaveiką;

3. Metodai neefektyvūs, kai maišos funkcijos skaičiuojamos:

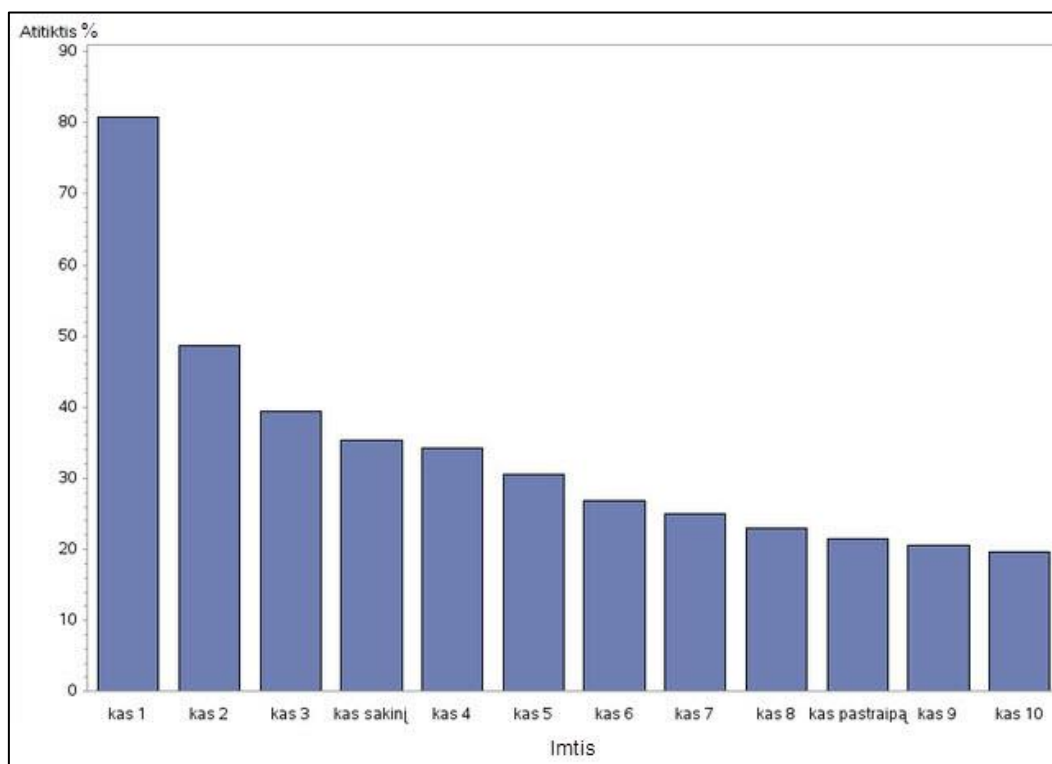
- 3 – 10 žodžių be persidengimų ;
- kas 5 – 10 žodžiui;
- kas 3 sakinius (kas 2 sakinius pakankamai efektyvus skaičiuojant maišos funkcijas dokumentams, kuriuose buvo įterptą papildomai 7 – 16 % teksto).

4.4. Eksperimentas, kai pakeisti atsitiktiniai žodžiai dokumentų tekste

Tyrimas atliekamas su 15 vnt. dokumentų, kuriuose yra originalių dokumentų informacija. Dokumentuose pakeisti atsitiktiniai žodžiai.

4.4.1. Maišos funkcijų skaičiavimas be persidengimų

Pirmiausiai maišos funkcijos skaičiuojamos visam dokumentų turiniui be persidengimų skirtingam kiekiui žodžių, lyginant dokumentų maišos funkcijas su originalaus dokumento maišos funkcijomis. Žemiau pateiktoje diagramoje (Žr. 22 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas skirtingam kiekiui žodžių. Eksperimentas atliekamas su prototipais Nr. 1, Nr. 2 ir Nr. 3.



22 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui

Pateikti rezultatai rodo, kad skaičiuojant maišos funkcijas skirtingo kiekio žodžiams, kai dokumente yra pakeisti atsitiktiniai žodžiai, nustatymo sutapimo procentas skirtingas. Tiksliausiai nustatoma dokumentų vidutinė atitiktis skaičiuojant maišos funkcijas kiekvienam žodžiui (~ 80,9 %). Skaičiuojant maišos funkcijas 2 – 10 žodžių be persidengimų, sakiniams ir pastraipoms, metodas neefektyvus, kai tekste pakeisti atsitiktiniai žodžiai.

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų procentas, pagal dokumente atsitiktinai pakeistų žodžių skaičių.

15 lentelė. Atitikimų procentas pagal dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas

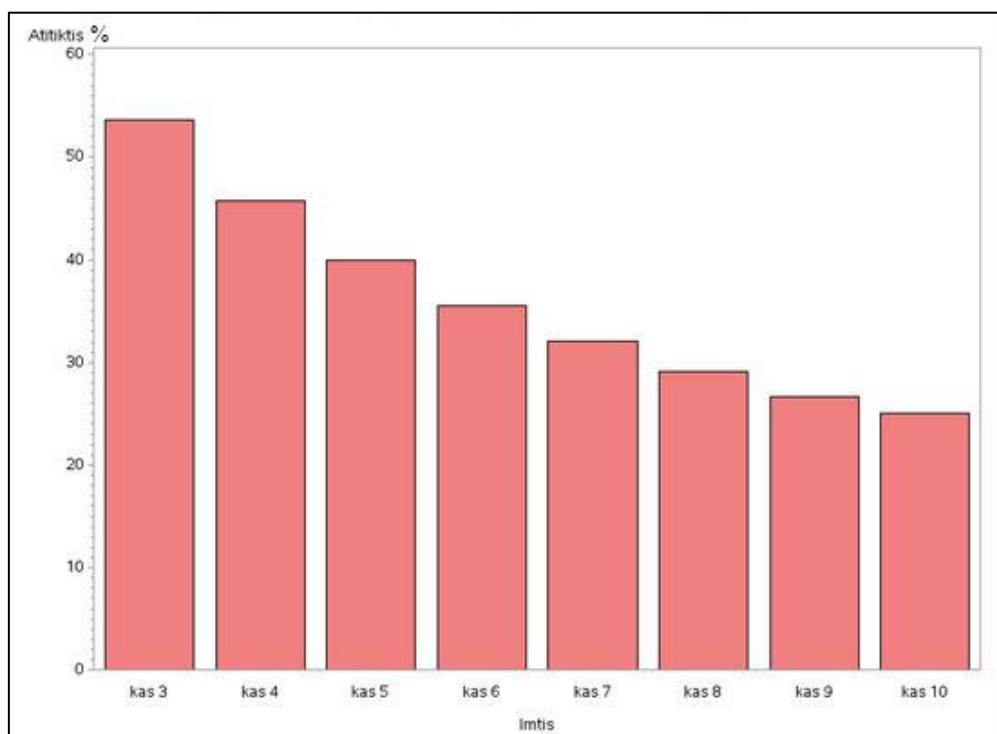
	kas1	kas 2	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10	kas Sak	kas Pastr
test 1.16	94,73759	89,92245	84,41882	80,68898	75,12924	72,87897	68,18656	66,09322	62,17391	58,92857	57,59898	37,30159
test 1.17	90,78102	58,0329	50,3653	45,27005	41,37595	36,38515	34,01348	30,9386	25,9025	26,83107	39,39974	21,69312
test 1.18	83,08232	37,5731	27,70122	21,56273	17,35557	14,51744	10,86365	9,040609	8,722003	7,267574	30,42784	19,31217
test 1.19	74,17674	35,35237	23,18624	15,73458	13,58873	8,940591	9,326477	6,640523	5,30303	4,166667	27,65006	15,60847
test 1.20	61,45436	22,76944	11,43315	7,50279	5,365854	1,99005	2,339181	2	0,757576	0,833333	22,12644	13,22751

Iš pateiktos lentelės matyti, kad didinant atsitiktinai pakeistų žodžių skaičių, geriausiai nustatomas atitikimas skaičiuojant maišos funkcijas kiekvienam žodžiui. Skaičiuojant maišos funkcijas 10 žodžių, sakiniams ir pastraipoms technika neefektyvi net pakeitus tik ~ 5 % žodžių.

Skaičiuojant maišos funkcijas 2 ar 9 žodžiams, metodas neefektyvus pakeitus ~ 10 % žodžių.

4.4.2. Maišos funkcijų skaičiavimas su persidengimais

Maišos funkcijos skaičiuojamos visam dokumentų turiniui su persidengimais skirtingam kiekiui žodžių. Žemiau pateiktoje diagramoje (žr. 23 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas skirtingam kiekiui žodžių su persidengimais. Eksperimentas atliekamas su prototipu Nr. 4.



23 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais

Matyti, kad skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais, kai dokumente atsitiktinai pakeisti žodžiai, nustatytas sutapimo procentas skiriasi. Tiksliausiai nustatomas dokumentų sutapimas skaičiuojant maišos funkcijas trimis žodžiams, nustatytas vidutinis sutapimas yra ~ 53,6 %, todėl galime teigti, kad šis metodas yra visiškai neefektyvus skaičiuojant maišos funkcijas dokumentams, kurių tekste yra atsitiktinai pakeisti žodžiai.

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų procentas, pagal dokumente atsitiktinai pakeistų žodžių skaičių.

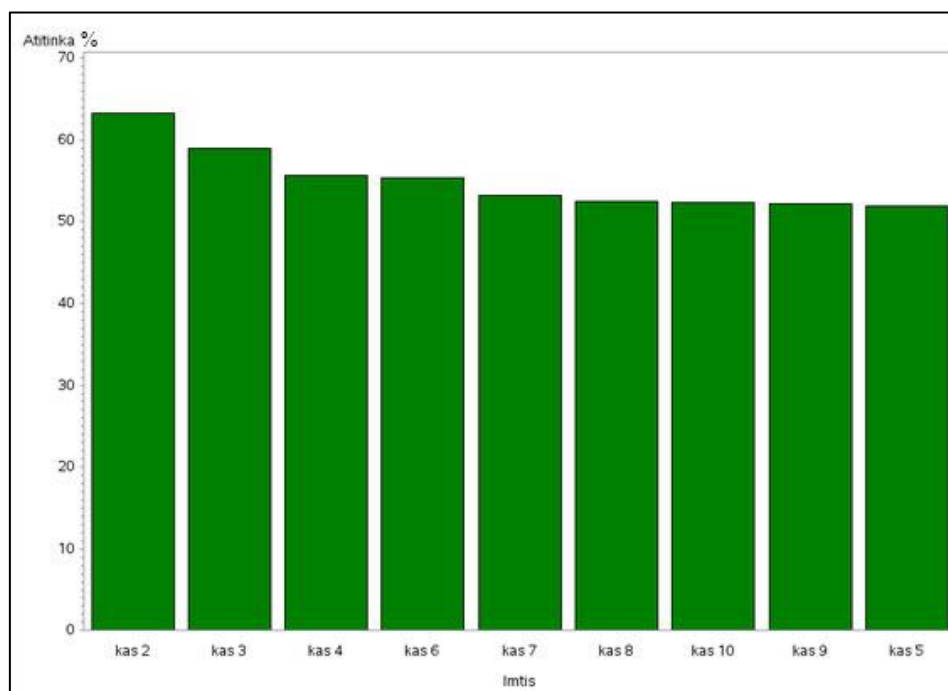
16 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir žodžių kiekį skaičiuojant maišos funkcijas su persidengimais

	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10
test 1.16	85,04104	80,47941	76,3801	72,95887	69,60499	66,25111	62,89723	60,82723
test 1.17	72,64077	64,83549	58,23958	52,3891	47,21832	42,63078	38,43982	36,18317
test 1.18	53,79299	43,48905	35,9849	30,35123	26,05413	22,56234	19,64633	17,83985
test 1.19	39,431	29,36466	22,42252	17,89427	14,53556	12,10308	10,32586	9,156185
test 1.20	17,01463	10,71224	6,554656	4,246315	2,845616	2,200895	1,7216	1,323408

Iš pateiktų rezultatų matyti, kad skaičiuojant maišos funkcijas 3 – 10 žodžių su persidengimais, kai yra pakeista ~ 5 % žodžių, yra efektyvus. Taip pat skaičiuojant maišos funkcijas 3 – 4 žodžiams, kai pakeista ~ 10 % žodžių, metodas yra efektyvus. Visais kitais atvejais, metodas yra neefektyvus, nenustatomi sutapimai.

4.4.3. Maišos funkcijų skaičiavimas kas k -iesiems žodžiams

Maišos funkcijos skaičiuojamos ne visam dokumento turiniui, o kas k -ajam žodžiui. Žemiau pateiktoje diagramoje (žr. 24 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas imant skirtingus k -uosius žodžius. Eksperimentas atliekamas su prototipu Nr. 5.



24 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems žodžiams tekste

Iš pateiktų rezultatų matyti, kad skaičiuojant maišos funkcijas skirtingam k -ajam žodžiui, kai dokumente pakeisti atsitiktiniai žodžiai, nustatytas sutapimo procentas yra skirtingas, vidurkis $\sim 55,1 \%$. Tiksliausiai nustatomas dokumentų sutapimas skaičiuojant maišos funkcijas kas antram žodžiui, nustatytas vidutinis sutapimas yra $\sim 63,34 \%$. Toliau, skaičiuojant maišos funkcijas kas 3 – 10 žodžiui, metodas neefektyvus, nes nustatomas sutapimas yra mažesnis už 60% .

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų procentas, pagal dokumente atsitiktinai pakeistų žodžių skaičių.

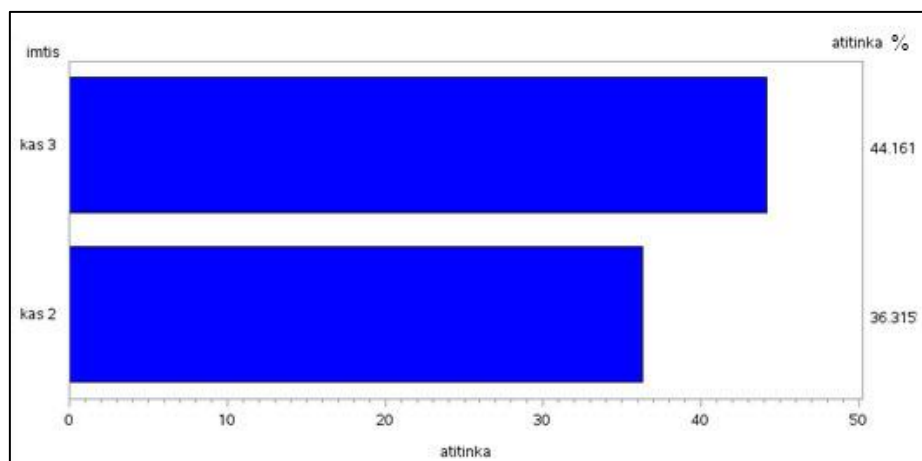
17 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir k -uosius žodžius skaičiuojant maišos funkcijas

	kas 2	kas 3	kas 4	kas 5	kas 6	kas 7	kas 8	kas 9	kas 10
test 1.16	93,36231	88,40844	88,42063	88,53578	87,66239	83,71354	86,3011	89,04873	89,61631
test 1.17	57,92307	55,71616	53,51269	49,35362	51,27795	49,62715	49,56685	47,07212	47,29097
test 1.18	69,664	56,40214	52,81045	46,7726	51,22256	46,68401	48,60086	49,39181	49,92939
test 1.19	50,53097	50,14585	45,34889	40,3582	47,83638	44,68256	42,42236	39,4386	42,67001
test 1.20	45,2133	44,14876	38,16457	34,75626	38,6652	41,56598	35,79041	35,81676	32,19621

Iš pateiktų rezultatų matyti, kad skaičiuojant maišos funkcijas kas 2 – 10 žodžiui, kai yra pakeista $\sim 5 \%$ žodžių – yra efektyvus, nustatomas sutapimas. Visais kitais atvejais nustatomas per mažas sutapimas.

4.4.4. Maišos funkcijų skaičiavimas kas k -iesiems sakiniams

Maišos funkcijos skaičiuojamos ne visam dokumento turiniui, o kas k -ajam sakiniui (kas antram ir kas trečiam sakiniui) tekste. Žemiau pateiktoje diagramoje (žr. 25 pav.) yra pateikiamas dokumentų sutapimo su originaliu dokumentu vidurkis procentais, skaičiuojant maišos funkcijas kas antram ir kas trečiam sakiniui. Eksperimentas atliekamas su prototipu Nr. 6.



25 pav. Atitiktis su originaliu dokumentu skaičiuojant maišos funkcijas kas k -iesiems sakiniams tekste

Iš gautų rezultatų matome, kad skaičiuojant maišos funkcijas kas antram ir kas trečiam sakiniui skiriasi. Skaičiuojant maišos funkcijas kas antram sakiniui, vidutinis sutapimo procentas yra $\sim 36,3 \%$, o skaičiuojant kas trečiam sakiniui $\sim 44,2 \%$, todėl šis metodas yra neefektyvus, kai dokumente yra pakeisti atsitiktiniai žodžiai.

Toliau pateikiama lentelė, kurioje nurodomas vidutinis atitikimų skaičius, pagal dokumente atsitiktinai pakeistų žodžių skaičių.

18 lentelė. Atitikimų procentas pagal skirtingus dokumentus ir kas k -uosius sakinius skaičiuojant maišos funkcijas

	kas 2	kas 3
test 1.16	51,13191	63,91941
test 1.17	37,51337	46,337
test 1.18	33,29768	42,23443
test 1.19	31,63102	37,10623
test 1.20	28,00357	31,20879

Esant skirtingam pakeistam kiekiui žodžių, nustatomas per mažas sutapimas, metodas yra neefektyvus.

4.4.5. Išvados

1. Vykdamas eksperimentą su dokumentais, kuriuose pakeisti atsitiktiniai žodžiai, nustatyta, kad tiksliausiai dokumentų sutapimas identifikuotas skaičiuojant maišos funkcijas kiekvienam žodžiui, nustatytas sutapimas ~ 80,1 % . Toliau, skaičiuojant maišos funkcijas kas antram žodžiui (~ 63,34 %), tačiau labai „jautrus“ pakeistų žodžių kiekiui.

2. Metodai yra neefektyvūs, kai maišos funkcijos skaičiuojamos:

- 2 – 10 žodžių be persidengimų (efektyvus tik kai pakeista ~ 5 % žodžių);
- sakiniams;
- pastraipoms;
- 3 – 10 žodžių su persidengimais (efektyvus tik kai pakeista ~ 5 % žodžių);
- kas 2 – 10 žodžiui (efektyvus tik kai pakeista ~ 5 % žodžių).

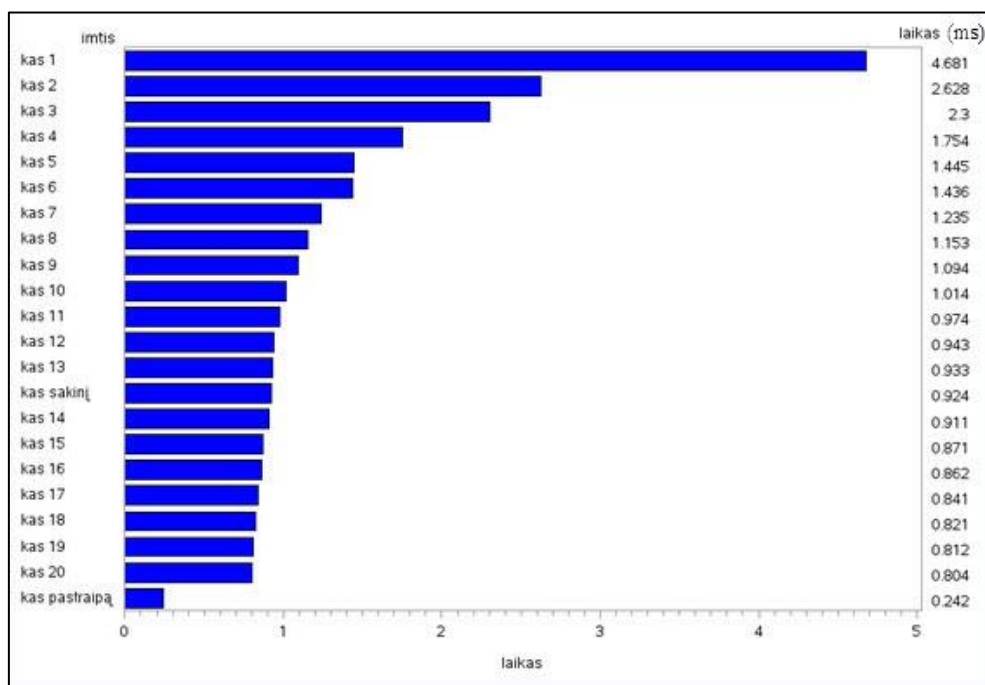
4.5. Greitaveika

Greitaveikos testavimui buvo naudotas didesnis dokumentas, sudarytas iš 13 332 žodžių (dokumentas sudarytas iš atsitiktinių straipsnių). Laiko skaičiavimui naudota „NetBeans IDE 8.2“ funkcija „Profile“.

Kiekvieno prototipo algoritmas buvo leidžiamas tris kartus, fiksuotas vidutinis greitis.

4.5.1. Maišos funkcijų skaičiavimas be persidengimų

Pirmiausiai greitaveika testuojama skaičiuojant maišos funkcijas skirtingam kiekiui žodžių be persidengimų, po pastraipą ir sakinį. T. y. testuojama prototipų Nr. 1, Nr. 2 ir Nr. 3 greitaveika.

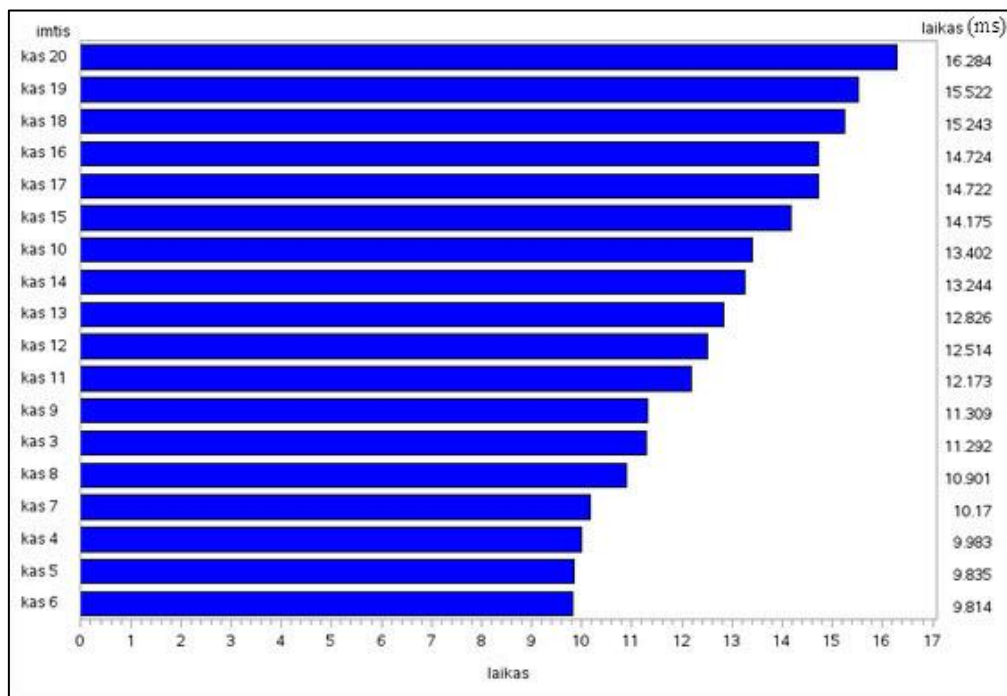


26 pav. Greitaveika skaičiuojant maišos funkcijas skirtingam kiekiui žodžių be persidengimų

Iš pateiktų rezultatų matome, kad maišos funkcijų skaičiavimo greitis eksponentiškai trumpėja didinant žodžių kiekį. Vidutinis maišos funkcijų skaičiavimo laikas $\sim 1,3$ ms. Trumpiausiai maišos funkcijos skaičiuojamos pastraipoms, užtrunka $\sim 0,24$ ms, o ilgiausiai $\sim 4,68$ ms kiekvienam žodžiui. Geriausiai atitiktis nustačiusios technikos, skaičiuojant maišos funkcijas kiekvienam sakiniui, skaičiavimo trukmė trumpesnė nei vidurkis $\sim 0,92$ ms.

4.5.2. Maišos funkcijų skaičiavimas su persidengimais

Greitaveika testuojama skaičiuojant maišos funkcijas skirtingam kiekiui žodžių su persidengimais. T. y. testuojama prototipo Nr. 4 greitaveika.



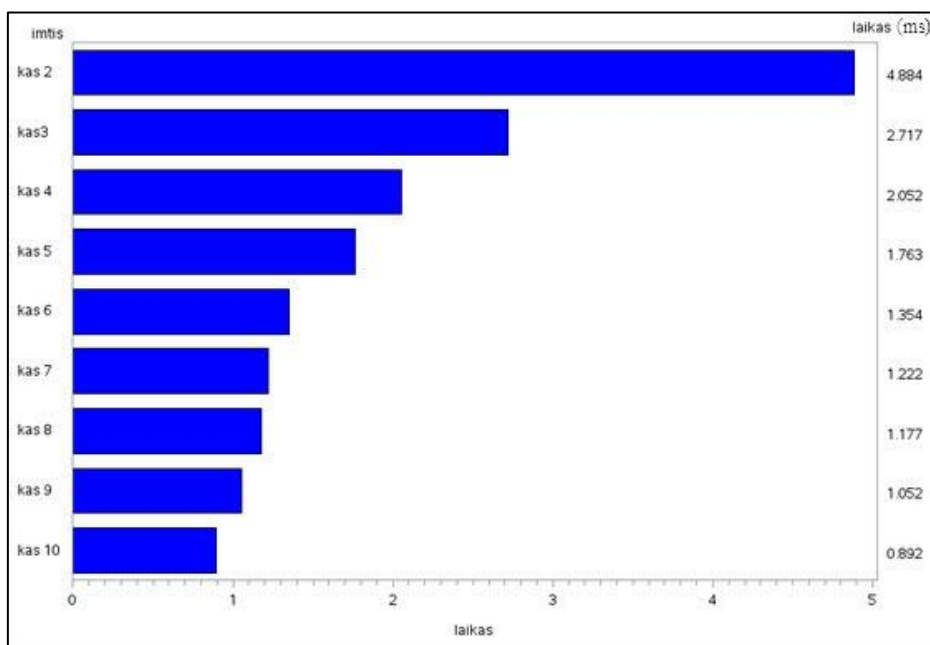
27 pav. Greitaveika skaičiuojant maišos funkcijas skirtingam kiekiui žodžių su persidengimais

Iš gautų rezultatų matome, kad greitaveikos laikas žymiai ilgesnis, nei skaičiuojant maišos funkcijas be persidengimų. Vidutinis maišos funkcijų skaičiavimo laikas yra ~ 12,67 ms. T. y. trunka beveik 10 kartų ilgiau nei skaičiuojant maišos funkcijas be persidengimų.

Skaičiuojant maišos funkcijas šia technika, trumpiausiai užtrunka skaičiuoti maišos funkcijas šešiams, penkiems ir keturiems žodžiams (užtrunka ~ 9,88 ms). Ilgiausiai užtrunka skaičiuojant maišos funkcijas 20 žodžių (~ 16,3 ms).

4.5.3. Maišos funkcijų skaičiavimas kas k -iesiems žodžiams

Toliau greitaveika testuojama skaičiuojant maišos funkcijas kas k -iesiems žodžiams. T. y. testuojama prototipo Nr. 5 greitaveika.

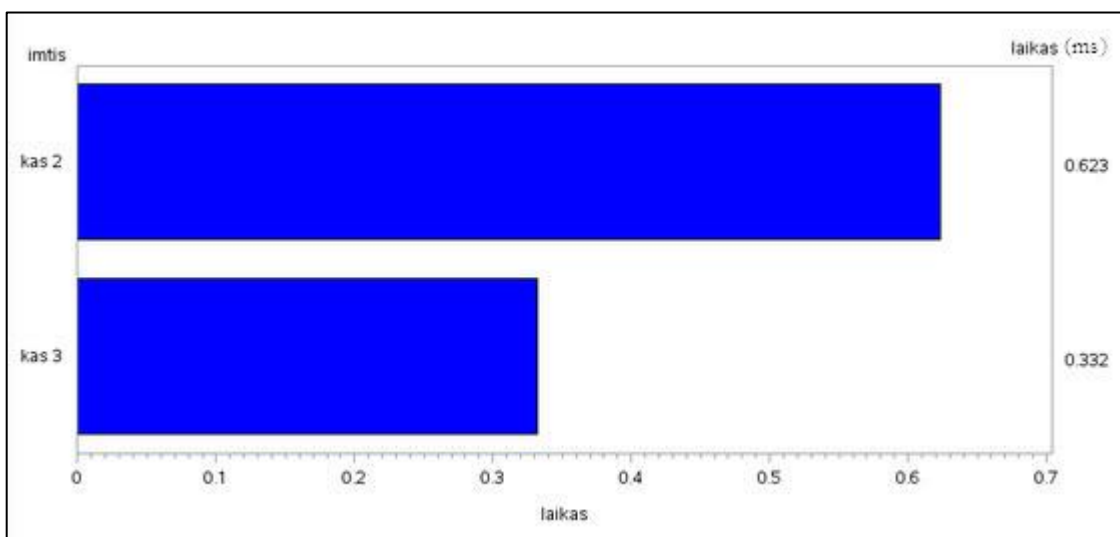


28 pav. Greitaveika skaičiuojant maišos funkcijas kas k -iesiems žodžiams

Iš pateiktų rezultatų matome, kad maišos funkcijų skaičiavimo greitis eksponentiškai trumpėja imant didesnę atstumą tarp žodžių, nes maišos funkcijos skaičiuojamos mažesniai kiekiui žodžių. Vidutinis maišos funkcijų skaičiavimo laikas yra $\sim 1,9$ ms. Trumpiausiai maišos funkcijos skaičiuojamos kas 10 žodžiui ($\sim 0,89$ ms), o ilgiausiai – kas 2 žodžiui ($\sim 4,88$ ms).

4.5.4. Maišos funkcijų skaičiavimas kas k -iesiems sakiniams

Greitaveika testuojama skaičiuojant maišos funkcijas kas antram ir kas trečiam sakiniui. Testuojama prototipo Nr. 6 greitaveika.



29 pav. Greitaveika skaičiuojant maišos funkcijas kas 2 ir 3 sakiniui

Iš pateiktų rezultatų matome, kad maišos funkcijų skaičiavimo greitis, kaip ir tikėtasi, skiriasi skaičiuojant maišos funkcijas kas 2 ir kas 3 sakinius. Maišos funkcijų skaičiavimas kas 2 sakinius užtrunka $\sim 0,62$ ms, o kas 3 sakinius $\sim 0,33$ ms.

4.5.5. Išvados

1. Geresnė greitaveika prototipų, kurie skaičiuoja maišos funkcijas didesniai kiekiui žodžių. Greičiausiai apskaičiuojamos maišos funkcijos, skaičiuojant pastraipoms, vidutinis užfiksuotas laikas yra $\sim 0,24$ ms.

2. Blogiausiai greitaveika fiksuota skaičiuojant maišos funkcijas skirtingam žodžių kiekiui su persidengimais. Fiksuotas vidutinis laikas $\sim 12,67$ ms yra 10 kartų didesnis nei užfiksuotas vidutinis laikas be persidengimų.

3. Skaičiuojant maišos funkcijas sakiniams, užfiksuotas vidutinis laikas $\sim 0,92$ ms, kuris yra beveik 0,4 ms mažesnis nei vidurkis skaičiuojant maišos funkcijas pagal skirtingą žodžių kiekį be persidengimų.

5. IŠVADOS

1. Išanalizavus literatūrą nustatyta, kad viena iš daugiausiai žadančių teksto analizės technikų yra dalinio sutapimo. Tai daugumos komercinių sistemų naudojama technika, tačiau nėra aprašoma, kokias dalinio sutapimo strategijas taiko skaičiuojant maišos funkcijas. Taip pat nėra daug mokslinių straipsnių, kurie analizuotų ir lygintų skirtingas maišų skaičiavimo strategijas, todėl nuspręsta pasiūlyti metodą ir atlikti eksperimentą.

2. Sukonstruoti maišos funkcijų skaičiavimo prototipai, kurie realizavo skirtingas strategijas:

- maišos funkcijų skaičiavimas be persidengimų visam dokumento turiniui, išskaidant jį į pasirinkto ilgio žodžių aibes (1–10 žodžių);
- maišos funkcijų skaičiavimas kiekvienam sakiniui;
- maišos funkcijų skaičiavimas kiekvienai pastraipai;
- maišos funkcijų skaičiavimas su persidengimais visam dokumento turiniui, išskaidant jį į pasirinkto ilgio žodžių aibes (3–10 žodžių);
- maišos funkcijų skaičiavimas kas k -ajam žodžiui dokumente (kas 2–10 žodžiui);
- maišos funkcijų skaičiavimas kas k -ajam sakiniui dokumente (kas 2 ir kas 3 žodžiui);

Eksperimentuojant prototipus sukurti 61 vnt. dokumentų, kurie išskaidyti į penkis tipus:

- dokumento gale įterptas tekstas;
- dokumento pradžioje įterptas tekstas;
- dokumento viduryje įterptas tekstas;
- pakeisti atsitiktiniai žodžiai tekste;
- ilgo turinio dokumentas skirtas prototipų greitaveikai;

Pagal dokumentų tipus išskaidyti gauti eksperimentų rezultatai, todėl kiekvienu atveju nustatomas geresnis prototipas ir jo realizacija.

3. Vykdam eksperimentus su dokumentais, kurių teksto pradžioje, pabaigoje ar viduryje yra įterpta papildomo teksto nustatyta:

- kai dokumento gale įterptas tekstas, visi metodai yra efektyvūs;
- tiksliausiai dokumentų sutapimas identifiktuotas skaičiuojant maišos funkcijas kiekvienam sakiniui ir kiekvienam žodžiui. Nustatytas vidutinis atitikimas su originaliais dokumentais skaičiuojant sakiniams ~ 92,6 %, o žodžiams ~ 91,2 %. Atsižvelgiant ir į greitaveiką – geriau

skaičiuoti maišos funkcijas sakiniams (trunka $\sim 0,9$ ms), kai tuo tarpu kiekvienam žodžiui trunka $\sim 4,7$ ms;

- skaičiuojant maišos funkcijas be persidengimų nustatyta, kad kai dokumento priekyje yra įterptas tekstas, maišos funkcijų skaičiavimas 2–10 žodžių yra neefektyvus. Kai dokumento viduryje yra įterptas tekstas, maišos funkcijų skaičiavimas 3–10 žodžių yra neefektyvus, nustatytas sutapimas yra mažesnis už 60 %, todėl tokios metodo realizacijos turėtų būti nenaudojamos;
- skaičiuojant maišos funkcijas skirtingiems kas k -iesiems žodžiams, kai dokumento pradžioje įterptas tekstas, metodas neefektyvus. Kai dokumento viduryje įterptas tekstas, skaičiuojant maišos funkcijas kas 2 ar kas 3 žodžiui, metodas efektyvus. Neefektyvus skaičiuojant maišos funkcijas kas 4–10 žodžiui, nustatytas sutapimas mažesnis už 60 %.

4. Vykdamas eksperimentus su dokumentais, kuriuose pakeisti atsitiktiniai žodžiai nustatyta:

- tiksliausiai dokumentų sutapimas identifikuotas skaičiuojant maišos funkcijas kiekvienam žodžiui, nustatytas sutapimas $\sim 80,1$ %, skaičiuojant maišos funkcijas kas antram žodžiui ($\sim 63,34$ %), tačiau labai „jautrus“ pakeistų žodžių kiekiui;
- metodai yra neefektyvūs, kai maišos funkcijos skaičiuojamos:
 - 2–10 žodžių be persidengimų (efektyvūs tik tada, kai pakeista ~ 5 % žodžių);
 - sakiniams;
 - pastraipoms;
 - 3–10 žodžių su persidengimais (efektyvus tik tada, kai pakeista ~ 5 % žodžių);
 - kas 2–10 žodžiui (efektyvus tik tada, kai pakeista ~ 5 % žodžių).

5. Nustatyta, kad universaliausias metodas, tinkantis visiems dokumentų tipams yra maišos funkcijų skaičiavimas kiekvienam žodžiui.

6. LITERATŪROS SĄRAŠAS

1. 2016 Data Breach Statistics: By the Numbers [interaktyvus] 2016. [Žiūrėta 2017-03-14]. Prieiga per internetą: <http://losspreventionmedia.com/insider/data-protection/2016-data-breach-statistics-by-the-numbers/>;
2. Unstructured-data [interaktyvus] 2009. [Žiūrėta 2016-06-02]. Prieiga per internetą: <http://www.pcmag.com/encyclopedia/term/53486/unstructured-data>;
3. Burke B., 2008. Information Protection and Control survey: Data Loss Prevention and Encryption trends, *IP special report*. Framingham: IDC;
4. Shabtai A., Elovici Y., Rokach L., 2012. *A Survey of data leakage detection and prevention solutions*. New York: Springer;
5. Phua C., 2009. Protecting organisations from personal data breaches, *Computer Fraud and Security* 1: 13–18;
6. Ouellet E., 2012. Magic Quadrant for Content-Aware Data Loss Prevention, *Technical Report* 1: 15–28;
7. Readshaw Neil I., Ramanathan J., George Bray G., 2012. *Method and apparatus for associating data loss protection (DLP) policies with endpoint..* United States patent, US 2012/0151551;
8. Wuchner T., Pretschner A., 2012. Data loss prevention based on data-driven usage control. *2012 IEEE 23rd International Symposium on Software Reliability Engineering*;
9. Sabah Al-Fedagi, 2012. A Conceptual Foundation for Data Loss Prevention. *International Journal of Digital Content Technology and its Applications*, 5(3): 293–303;
10. Takebayshi T., Tsuda H., Hasebe T., Masuoka R., 2009. Data Loss Prevention Technologies. *FUJITSU Sci. Tech*, 46(1): 47–55;
11. Costante E., Vavilis S., Etalle S., Petkovic M., & Zannone N., 2013. Database anomalous activities: detection and quantification, *2013 International Conference on Security and Cryptography*;
12. Gugelmann D., Studerus P., Lenders V., Ager B., 2015. Can content-based data loss prevention solutions prevent data leakage in web traffic. *IEEE Security Privacy* 13(4): 52–59;
13. Mogull R., 2010. Understanding and Selecting a Data Loss Prevention Solution. *SANS Institute*;

14. Walsh B., 2002. Introduction to Bayesian Analysis. *Lecture notes for EEB 596z*;
15. Alfikri Zakiy F., Purwarianti A., 2014. Detailed Analysis of Extrinsic Plagiarism Detection System Using Machine Learning Approach (Naive Bayes and SVM). *Telkomnika Indonesian Journal of Electrical Engineering* 12(11);
16. Joachims T., 1998. *Text categorization with support vector machines: Learning with many relevant features*. Berlin: Springer 137–142;
17. Sebastiani F., 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)* 34(1): 1–47;
18. Robinson G., 2003. *A statistical approach to the spam problem*. *Linux Journal*, 107(3);
19. Lee Peter M., 2012. *Bayesian Statistics*. New Yourk: Wiley;
20. Osman A., Salim N, Elhadi A., 2013. A tree-based conceptual matching for plagiarism detection, *ICCEEE*;
21. Randy D., 2015. Data Loss Protection. *The SANS institute InfoSec* 1–30;
22. Finding or Veryfing Credit Card Numbers [interaktyvus] 2016. [Žiūrėta 2017-01-07]. Prieiga per internetą: <http://www.regular-expressions.info/creditcard.html>;
23. Luno formulė [interaktyvus] 2015. [Žiūrėta 2017-01-07]. Prieiga per internetą: https://lt.wikipedia.org/wiki/Luno_formul%C4%97;
24. Breitinger F., Baggili I., 2014. File Detection on Network Traffic Using Approximate Matching, *Journal of Digital Forensics, Security nad Law* 9: 2–23;
25. Richard Karp M. and Michael Rabin. O., 1987. Pattern-matching algorithms, *IBM Journal of Research and Development*, 31(2): 249–260;
26. Charras C. and Lecroq T., 2004. Hanbook of exact string matching algorithms. *Citeseer*, 31–32;
27. Hoad Timothy C., Zobel J., 2003. Methods for Identifying Versioned and Plagiarized Documents. *Journal of the American society for information science and technology*, 54(3): 203–215;
28. Black J., 2006. Compare-by-hash: A Reasoned Analysis. *USENIX Annual Technical Conference*, 85–90;
29. Henson V., 2003. An analysis of compare-by-hash.. *USENIX Annual Technical Conference*, 13–18;

30. Mendel, F., Nad, T., Schaffer, M., 2011. Finding SHA-2 Characteristics: Searching through a Minefield of Contradictions. *17th International Conference on the Theory and Application of Cryptology and Information Security*, 288–307;
31. Mendel, F., Nad, T., Schaffer, M., 2013. Improve Local Collisions: New Attacks on Reduced SHA256. *LNCS 7881*, 262–278;
32. Kavaliūnas K., 2008. Kriptografijos maišos algoritmų lyginimas. *11-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“ straipsnių rinkinys*, 31–37;

7. PRIEDAI

7.1. Prototipo realizacija

```
package hashgeneratorutils;
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class HashGeneratorUtils {
    private HashGeneratorUtils() {

    }
    public static String generateSHA256(String message) throws
    HashGenerationException {
        return hashString(message, "SHA-256");
    }

    private static String hashString(String message, String algorithm)
        throws HashGenerationException {

        try {
            MessageDigest digest = MessageDigest.getInstance(algorithm);
            byte[] hashedBytes = digest.digest(message.getBytes("UTF-8"));

            return convertByteArrayToHexString(hashedBytes);
        } catch (NoSuchAlgorithmException | UnsupportedEncodingException ex) {
            throw new HashGenerationException(
                "neimanoma sugeneruoti hash is gauto teksto", ex);
        }
    }

    private static String convertByteArrayToHexString(byte[] arrayBytes) {
        StringBuffer stringBuffer = new StringBuffer();
        for (int i = 0; i < arrayBytes.length; i++) {
            stringBuffer.append(Integer.toString((arrayBytes[i] & 0xff) + 0x100, 16)
                .substring(1));
        }
        return stringBuffer.toString();
    }
}
```

```
package hashgeneratorutils;

public class HashGenerationException extends Exception {

    public HashGenerationException() {
        super();
    }

    public HashGenerationException(String message, Throwable throwable) {
        super(message, throwable);
    }
}
```

```

    }

    public HashGenerationException(String message) {
        super(message);
    }

    public HashGenerationException(Throwable throwable) {
        super(throwable);
    }
}

```

7.1.1. Prototipas Nr.1

```

package hashgeneratorutils;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.logging.Level;
import java.util.logging.Logger;

public class StringHashGeneratorExample3 {

    public static void main(String[] args) throws IOException {
        String contents = new
String(Files.readAllBytes(Paths.get("C:/Users/Jurate/Desktop/neslapti/NEtest4.txt")))
;
        String content = contents.toLowerCase();
        String[] words = content.split("\\s+|\\.\\.\\.|\\|\\|,|\\|\\r\\|\\n\\|\\|\\-|\\|\\?|\\|\\:|\\|\\;|\\|\\!");
        int counter = 0;
        String[] fiveWords = new String[5];

        for (String word : words) {
            if (!word.trim().equals("")) {
                String hash;
                try {
                    hash = HashGeneratorUtils.generateMD5(word);
                    fiveWords[counter] = word;

                    if (counter >= 4) {
                        String fiveWordsJoin = String.join(" ", fiveWords);
                        hash = HashGeneratorUtils.generateMD5(fiveWordsJoin);
                        System.out.println(fiveWordsJoin + " " + hash);

                        fiveWords = new String[5];
                        counter = 0;
                    } else {
                        counter++;
                    }

                } catch (HashGenerationException ex) {

                    Logger.getLogger(StringHashGeneratorExample3.class.getName()).log(Level.SEVERE, null,
ex);
                }
            }
        }
    }
}

```

```
    }
}
}
```

7.1.2. Prototipas Nr.2

```
package hashgeneratorutils;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.logging.Level;
import java.util.logging.Logger;

public class StringHashGeneratorEverySentence {

    public static void main(String[] args) throws IOException {
        String contents = new
String(Files.readAllBytes(Paths.get("C:/Users/Jurate/Desktop/mag/mazas 3/test
3.21.txt")));
        String content = contents.toLowerCase();
        String[] words = content.split("\\.|\\r\\n|\\?|\\;|\\!");

        for (String word : words) {
            if (!word.trim().equals("")) {
                String hash;
                try {
                    hash = HashGeneratorUtils.generateSHA256(word);
                    System.out.println(hash);

                } catch (HashGenerationException ex) {

Logger.getLogger(StringHashGeneratorEverySentence.class.getName()).log(Level.SEVERE,
null, ex);

                }
            }
        }
    }
}
```

7.1.3. Prototipas Nr.3

```
package hashgeneratorutils;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.logging.Level;
import java.util.logging.Logger;

public class StringGeneratorEveryEnter {
```

```

    public static void main(String[] args) throws IOException {
        String contents = new
String(Files.readAllBytes(Paths.get("C:/Users/Jurate/Desktop/mag/mazas 3/test
3.21.txt")));
        String content = contents.toLowerCase();
        String[] words = content.split("\\r\\n");
        for (String word : words) {
            if (!word.trim().equals("")) {
                String hash;

                try {
                    hash = HashGeneratorUtils.generateSHA256(word);
                    System.out.println(hash);

                } catch (HashGenerationException ex) {

                }

            }

        }
    }
}

```

7.1.4. Prototipas Nr.4

```

package hashgeneratorutils;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.Arrays;

public class StringHashGeneratorExample2 {

    public static void main(String[] args) throws IOException {
        String contents = new
String(Files.readAllBytes(Paths.get("C:/Users/Jurate/Desktop/mag/mazas 3/test
3.20.txt")));
        String content = contents.toLowerCase();
        String[] words = content.split("\\s+|\\.|\\,|\\r\\n|\\-|\\?|\\:|\\;|\\!|");
        words = Arrays.stream(words)
            .filter(s -> (s.length() > 0))
            .toArray(String[]::new);

        String[] threeWords = new String[3];

        for (int i = 0; i < words.length; i++) {
            String word = words[i];
            if (!word.trim().equals("")) {
                for (int j = 0; j < 3; j++) {
                    if (words.length > i + j) {
                        threeWords[j] = words[i + j];
                    } else {
                        threeWords[j] = "";
                    }
                }
            }
        }
    }
}

```



```

where test07 in (select test1 from WORK.rez03)
UNION
select COUNT(test08), 'test08' from WORK.rez03
where test08 in (select test1 from WORK.rez03)
UNION
select COUNT(test09), 'test09' from WORK.rez03
where test09 in (select test1 from WORK.rez03)
UNION
select COUNT(test010), 'test010' from WORK.rez03
where test010 in (select test1 from WORK.rez03)
UNION
select COUNT(test011), 'test011' from WORK.rez03
where test011 in (select test1 from WORK.rez03)
UNION
select COUNT(test012), 'test012' from WORK.rez03
where test012 in (select test1 from WORK.rez03)
UNION
select COUNT(test013), 'test013' from WORK.rez03
where test013 in (select test1 from WORK.rez03)
UNION
select COUNT(test014), 'test014' from WORK.rez03
where test014 in (select test1 from WORK.rez03)
UNION
select COUNT(test015), 'test015' from WORK.rez03
where test015 in (select test1 from WORK.rez03)
UNION
select COUNT(test016), 'test016' from WORK.rez03
where test016 in (select test1 from WORK.rez03)
UNION
select COUNT(test017), 'test017' from WORK.rez03
where test017 in (select test1 from WORK.rez03)
UNION
select COUNT(test018), 'test018' from WORK.rez03
where test018 in (select test1 from WORK.rez03)
UNION
select COUNT(test019), 'test019' from WORK.rez03
where test019 in (select test1 from WORK.rez03)
UNION
select COUNT(test020), 'test020' from WORK.rez03
where test020 in (select test1 from WORK.rez03);

```