DOI: 10.2478/bhee-2015-0004

Pregledni naučni rad/Review scientific paper

USPOREDBA METODA PARTICIRANJA ZA IDENTIFIKACIJU DVOSTRUKIH ZAPISA U BAZAMA PODATAKA

COMPARATION OF PARTITIONING METHODS FOR IDENTIFICATION OF DUPLICATE RECORDS IN DATABASES

Đulaga Hadžić[1], Nermin Sarajlić[2], Jasmin Malkić[3]

Sažetak: Postoje različite metode za identifikaciju dvostrukih zapisa koje se koriste u procesima čišćenja podataka za skladišta podataka i data mining. U radu su testirane i analizirane dvije vrste metoda za identifikaciju dvostrukih zapisa u relacionim bazama podataka. Za testiranje metoda korišten je sistem razvijen u okviru naučno-istraživačkog rada autora ovog članka. Na osnovu izvršenih eksperimenata i postignutih rezultata izvučeni su određeni zaključci s aspekta kvaliteta testiranih metoda nad odabranim skupovima podataka. Na kraju su date određene preporuke za daljna istraživanja u ovoj oblasti.

Ključne riječi: čišćenje podataka, dvostruki zapisi, kvalitet, metoda blokiranja, metoda prozora.

Abstract: There are different methods for identification of duplicate records, and they are used in cleaning processes of data for a data warehouse and data mining. The paper tested and analyzed two types of methods for identification of duplicate records in relational databases. The system, developed within the framework of scientific and research work of the authors of this article, is used to test the given methods. Based on the experiments and the results obtained, certain conclusions have been drawn in terms of quality of the tested methods for the selected data sets. The paper concludes with recommendations for further research in this area.

Keywords: data cleansing, duplicate records, quality, blocking method, windowing method.

UVOD

Povećanjem količine podataka, kao i svakim povećanjem mogućnosti informacionih sistema u prikupljanju podataka iz distribuiranih i raznovrsnih izvora, povećavaju se i problemi vezani za kvalitet podataka. Informacija i kvalitet podataka su predmet širokih i aktivnih istraživanja, kako iz ugla informacionih sistema tako i iz perspektive baza podataka. Kvalitet podataka su definisali Tayi i Ballou [1] kao „upotrebljivost“ (prikladnost za korištenje). Jedan od najzanimljivijih problema kvaliteta podataka je višestruko, i pri tome, u realnom svijetu, različito predstavljanje istih objekata u podacima.

Ovakvi dvostruki ili višestruko isti zapisi podataka, u literaturi takozvani duplikati, su vrlo teški za identifikovati, posebno u velikim količinama podataka. Istovremeno, oni smanjuju upotrebljivost podatka, nepotrebno opterećuju računarske resurse, daju netačne indikatore perfomansi, te sprečavaju kvalitetno razumijevanje podataka i njihovih vrijednosti. Dvostruki zapisi podataka u kontekstu ovog rada ne predstavljaju svoje tačne kopije, već predstavljaju neznatne, a ponekad i veće razlike u pojedinim vrijednostima podataka. Tako se ovakvi dvostruki zapisi podataka nazivaju i fuzzy dvostruki zapisi podataka [2]. U kontekstu tradicionalnih sistema za upravljanje bazama podataka, dvostruki zapisi podataka predstavljaju potpuno iste kopije zapisa. Oni su jednostavni za identifikaciju i ovakvi zapisi nisu tema ovog rada. Tema ovog rada su dvostruki zapisi koji se odnose na fuzzy dvostruke zapise.

Postoje mnoge kompanije koje su izgradile servise za čišćenje podataka, koji obuhvataju i identifikaciju dvostrukih zapisa, i među njima su DataFlux, Hart-Hanks Data Technologies i Innovative Systems Inc. Nažalost, niko od njih nije spreman otkriti svoje algoritme koje koriste za čišćenje podataka.

1. METODE ZA IDENTIFIKACIJU DUPLIH ZAPISA

U novije vrijeme, čišćenje podataka se posmatra korakom pretprocesiranja u procesu izdvajanja znanja iz baza podataka [3], [4]. Razni sistemi izdvajanja znanja iz baza podataka i data mininga obavljaju aktivnosti čišćenja podataka na način koji je veoma povezan s oblasti (domenom) u kojoj se ova aktivnost odvija. U [5], proces čišćenja podataka se smatra procesom ispitivanja baze podataka, otkrivanjem netačnih i nedostajućih podataka i njihovo ispravljanje.

[1] Pedagoški zavod Tuzlanskog kantona, Bosna i Hercegovina djulaga.hadzic@pztz.ba
[2] Fakultet elektrotehnike Univerziteta u Tuzli, Bosna i Hercegovina
[3] OpenLink International GmbH, Berlin, Germany
Rad dostavljen: septembar 2015. Rad prihvaćen: oktobar 2015.

Identifikacija dvostrukih zapisa mora riješiti dvije inherentne poteškoće: brzo otkriti sve podudarne zapise u velikim skupovima podataka (efikasnost – učinkovitost) i pravilno identifikovati podudarne zapise i one koji to nisu (efektivnost).

Prva poteškoća je kvadratna kompleksnost problema. Konceptualno, svaki zapis kandidat mora biti upoređivan sa svim drugim zapisima. Predloženi su mnogi pristupi u rješavanju ovog problema i povećanju efikasnosti u identifikaciji dvostrukih zapisa kroz smanjenje potrebnog broja upoređivanja zapisa. Opšta ideja je da se izbjegne upoređivanje znatno različitih objekata i da se koncentriše samo na upoređivanje objekata koji imaju, u određenim granicama, brzo identifikovane sličnosti.

Druga poteškoća identificiranja dvostrukih zapisa je definicija odgovarajuće mjere sličnosti na osnovu koje će se odlučiti da li su dva zapisa ista ili ne. Osim toga prag sličnosti mora biti tako izabran da klasificira parove zapisa kao dvostruke (ako je sličnost veća ili jednaka od praga sličnosti) ili kao različite (ako je sličnost manja od praga sličnosti).

U dosadašnjim istraživanjima kao najpopularnije metode za identificiranje dvostrukih zapisa izdvojile su se metode particiranja, a među njima dvije vrste metoda: metode blokiranja i metode prozora.

Metode blokiranja vrše grupisanje skupa zapisa u disjunktne (razdvojene) particije ili blokove a zatim upoređuje sve parove zapisa samo unutar pojedinačnih blokova [6-8]. Bitan faktor u metodama blokiranja je kvalitetan izbor prediktivnog particiranja kojim se određuje broj i veličina blokova.

Metode prozora sortiraju podatke po ključu sortiranja i koristeći klizni prozor, prolaze kroz sortirane podatke i upoređuje samo one zapise koji se nalaze u prozoru.

Najpoznatija metoda od metoda blokiranja je Standardna metoda blokiranja. (*eng.* Standard Blocking Method (SBM)), a od metoda prozora Metoda sortiranih susjeda (*eng.* Sorted Neighborhood Method (SNM)). Obje metode su opisane u nastavku.

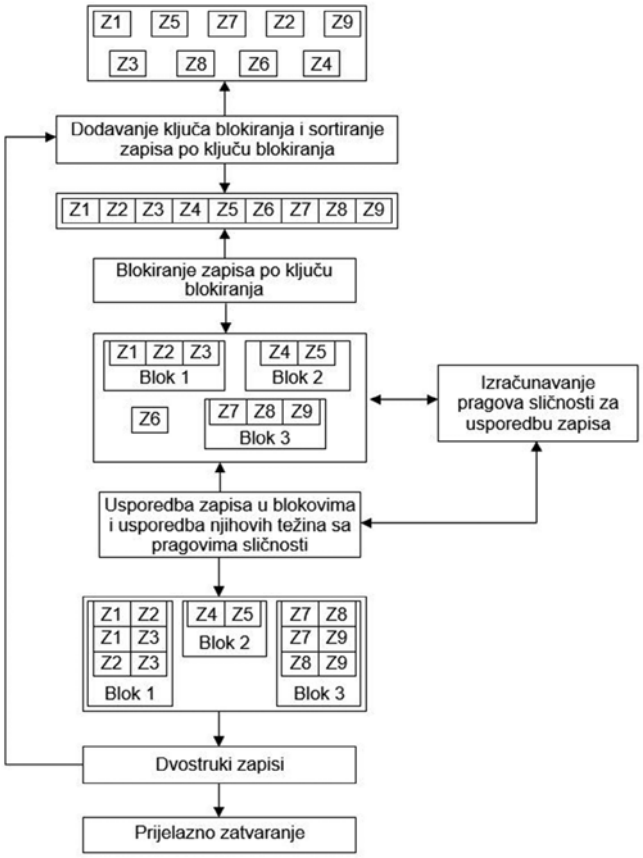
2. STANDARDNA METODA BLOKIRANJA

Na Slici 1 prikazani su osnovni koraci u implementaciji Standardne metode blokiranja. Ova metoda je zasnovana je na probalističkibaziranim tehnikama.

Njena implementacija kreće od ideja identifikacije dvostrukih zapisa od Newcombe i drugi [9] koje su formalizirane u matematički model od strane Fellegi i Suntera [11]. Za dvije tablice podataka A i B, sa n polja (atributa), na osnovu notacije od Fellegi i Suntera, sve parove zapisa iz A i B moguće je predstaviti kao:

$$A \; X \; B = \{(a,b);\; a \in A,\; b \in B\} \tag{1}$$

Svaki od parova zapisa pripada ili klasi M ili klasi U. Parovi zapisa koji pripadaju klasi M identificirani su kao da se podudaraju, a parovi zapisa koji pripadaju klasi U identificirani su kao da su različiti.



Slika 1: Standardna metoda blokiranja

Svaki par zapisa (a,b) može se predstaviti vektorom usporedbe zapisa $x = [x_1,...,x_n]$ sa n komponenti koje predstavljaju n polja (atributa) od tablica A i B. Pravilo odlučivanja u kome se na osnovu vektora podudarnosti određuje da li par zapisa pripada klasi M ili klasi U je:

$$(a,b) \in M;\; ako\; p(M|x) \geq p(U|x) \tag{2}$$
$$(a,b) \in U;\; u\; svim\; drugim\; slučajevima$$

Newcombe i drugi [9] prvi su prepoznali identifikaciju podudarnih zapisa kao Bayesian problem zaključivanja. Bayesovo pravilo odlučivanje (s minimalnom greškom) je:

$$M,\; if\; l(x) = p(x|M) \geq p(U)\; (a,b) \in p(x|U)\; p(M) \tag{3}$$
$$U,\; u\; svim\; drugim\; slučajevima$$

Odnos $l(x) = p(x|M)/p(x|U)$ naziva se odnos između vjerovatnoće da je par zapisa podudaran i vjerovatnoće da par zapisa nije podudaran. Odnos $p(U)/p(M)$ označava vrijednost praga za odnos vjerovatnoće (l) i koristi se za donošenje odluka da li je par zapisa podudaran ili ne.

Standardna metoda blokiranja je bazirana na EM algoritmu [9] (*eng.* Expectation Maximization), kojeg je Jaro [10] predložio, za izračun vjerovatnoće da se određeni par zapisa podudara, odnosno izračun vjerovatnoće da

par zapisa pripada klasi M, $P(xi=1|M)$. Vjerovatnoća da se određeni par zapisa ne podudara, odnosno da ima veliku vjerovatnoću da pripada klasi U, $P(x_i = 1|U)$, može se procijeniti uzimajući određeni broj slučajno izabranih parova zapisa.

EM algoritam generira skup parametara (m, u, p) putem iterativnog izračuna E-koraka (Expectation – Očekivanja) i M-koraka (Maximation – Maksimizacija). Parametri m, u i p predstavljaju vjerovatnoće na osnovu kojih će se formirati određeni vektor podudarnosti za dva zapisa koji se uspoređuju. Parametri m i u vezani su za atribute zapisa, dok p predstavlja ukupnu vjerovatnoću modela. Osnovni koraci u implementaciji Standardne metode blokiranja su:

1. Izrada ključa blokiranja i sortiranje zapisa po ključu blokiranja.

2. Izrada blokova: Zapisi čiji ključevi blokiranja imaju međusobnu sličnost vrijednosti 1 smještaju se u isti blok.

3. Usporedba zapisa u blokovima: Unutar svakog od formiranih blokova vrši se usporedba svih zapisa jednih s drugim. Uspoređivanje zapisa u blokovima se vrši tako što se uzme prvi zapis u bloku i uspoređuje s ostatkom zapisa u tom bloku. Zatim se uzima drugi zapis i uspoređuje se sa svim ostalim zapisima koji imaju poziciju veću od pozicije drugog zapisa i tako dalje s ostalim zapisima, dok se svi zapisi unutar jednog bloka međusobno ne usporede.

4. Za svaki par zapisa koji se uspoređuju unutar istih blokova vrši se izračunavanje vektora sličnosti preko binarnog modela [10] za izračunavanje vrijednosti vektora sličnosti γ tako da je γ_i^j=1 ako je sličnost između i-tih atributa para zapisa j jednaka ili iznad zadatog praga sličnosti, odnosno γ_i^j=0 ako je sličnost između i-tih atributa para zapisa j ispod zadatog praga sličnosti. Nakon što se izvrši usporedba svih parova zapisa unutar blokova, formirani vektori sličnosti se grupišu i sa svojim brojačem fekvencija (brojem pojavljivanja) $f(\gamma^j)$ pohranjuju za daljnu upotrebu.

5. Izračunavanje vrijednosti m_i i vrijednosti u_i vrši se iz vrijednosti vektora sličnosti γ^j i njihovih brojača frekvencije $f(\gamma^j)$ kroz primjenu EM algoritma koji generira skup parametara (m,u,p) iterativnim izračunavanjem kroz E-korak i M-korak.

6. Alternativni način izračuna u_i: Vrijednosti od u_i predstavljaju vjerovatnoće pripadnosti određenog para zapisa klasi U, tj da nisu podudarni. Stoga je Jaro [10] sugerirao alternativno izračunavanje vrijednosti u_i nad slučajno izabranim skupom zapisa.

7. Izračun pragova sličnosti parova zapisa: Slijedeći Gomatam i drugi [8], parovi zapisa (a,b) se označavaju kao:

– *podudarni (dvostruki)* (A_1),
– *moguće podudarni ili nepoznato* (A_2) i
– *nepodudarni (različiti)* (A_3).

Za fiksne vrijednosti stope pogrešno podudarnih (μ) i stope pogrešno nepodudarnih (λ) parova zapisa (a,b), Fellegi i Sunter [3] definišu optimalno pravilo povezivanja zapisa u Γ (skup svih mogući vektora podudarnosti) na nivoima μ i λ, označenog sa $L(\mu, \lambda, \Gamma)$ kao pravilo za koje je $P(A_1|U) = \mu$, $P(A_3|M) = \lambda$, i $P(A_2|L) \leq P(A_2|L')$ za sva ostala pravila L'. Neka odnos $m(\gamma)/u(\gamma)$ bude sortiran u opadajućem redoslijedu i odgovarajući γ budu indeksirani sa 1,2,..., N_Γ.

Ako je $\lambda = \sum_{i=n'}^{N_\Gamma} m(\gamma_i)$ i $\mu = \sum_{i=1}^{n} u(\gamma_i)$, $n < n'$, onda optimalno pravilo predstavlja funkciju omjera vjerovatnoća $m(\gamma)/u(\gamma)$ i dato je sa sljedećim izrazima:

$$(a,b) \in A_1 \quad \text{ako je} \quad T_\mu \leq \frac{m(\gamma)}{u(\gamma)} \tag{4}$$

$$(a,b) \in A_2 \quad \text{ako je} \quad T_\lambda < \frac{m(\gamma)}{u(\gamma)} < T_\mu \tag{5}$$

$$(a,b) \in A_3 \quad \text{ako je} \quad \frac{m(\gamma)}{u(\gamma)} \leq T_\lambda \tag{6}$$

gdje je:

$$T_\mu = \frac{m(\gamma_n)}{u(\gamma_n)} \tag{7}$$

$$T_\lambda = \frac{m(\gamma_{n'})}{u(\gamma_{n'})} \tag{8}$$

T_μ predstavlja gornji prag sličnosti, a T_λ predstavlja donji prag sličnosti za klasifikaciju parova zapisa.

8. Klasifikacija: Izračunavanje ukupne težine parova zapisa dobija se sumiranjem svih težina njihovih n atributa (i=1,...,n). Ako se i-ti atribut od para zapisa podudara, težina ovog atributa je data sa w_i=log_2(m_i/u_i), a ako se i-ti atribut od para zapisa ne podudara, onda je njegova težina data sa w_i=log_2((1-m_i)/(1-u_i)). Atributi koji se podudaraju daju pozitivan doprinos ovoj sumi, dok atributi koji se ne podudaraju daju negativan doprinos ovoj sumi. Njihova klasifikacija se vrši prema izrazima (4), (5) i (6). Za parove zapisa klasificirani kao "moguće podudarni" ili "nepoznato" (A_2) vrši se dodatna ekspertiza od strane čovjeka - stručnjaka.

9. Višestruki prolazi. Za određivanje sličnih zapisa koji se razlikuju po ključu blokiranja koristi se metoda višestrukog prolaza, gdje se za svaki prolaz definiše novi ključ blokiranja. Za svaki novi ključ blokiranja ponavljaju se koraci od 1 do 8.

10. Prijelazno zatvaranje. Na kraju ove metode vrši se tzv. prijelazno zatvaranje nad već identifikovanim dvostrukim zapisima. Naprimjer, ako zapisi a i b predstavljaju dvostruke zapise, a zapisi b i c također predstavljaju dvostruke zapise, onda su i zapisi a i c također dvostruki zapisi.

3. METODA SORTIRANIH SUSJEDA

Na Slici 2 prikazani su osnovni koraci u identifikaciji dvostrukih zapisa u Metodi sortiranih susjeda. Metoda sorti-

ranih susjeda koristi pristup na pravilima usporedbe zapisa. Pravila usporedbe zapisa grade se nad atributima (ili kombinacijom atributa) i na osnovu njih se vrši klasifikacija zapisa. U ovoj metodi korištena je varijanta na pravilimabazirane usporedbe zapisa koja je građena na teoriji jednakosti [12] .

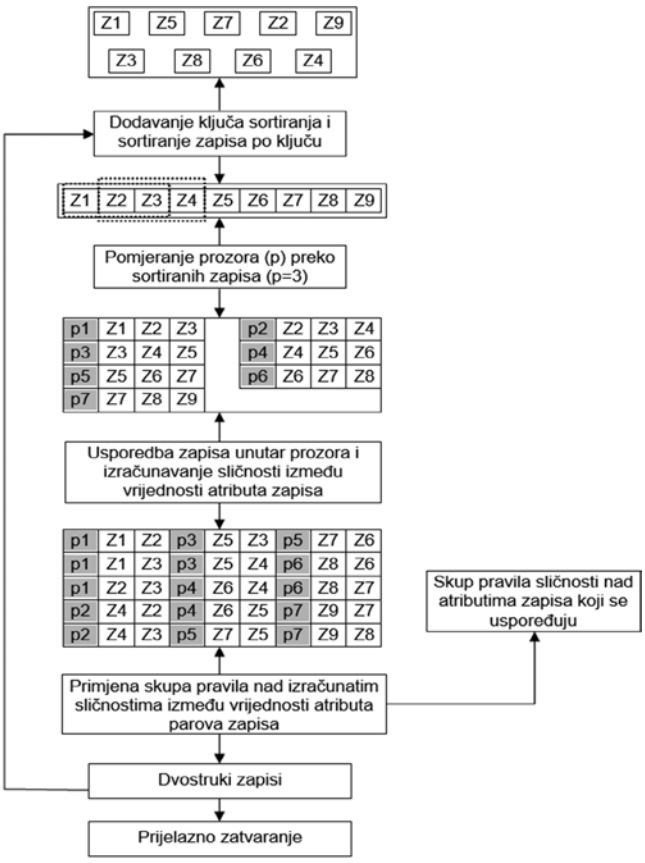
Pristup koji uspoređuje dva zapisa c_1 i c_2 upotrebom implikacije može se zapisati u sljedećem obliku:

$$P \Rightarrow c_2 \equiv c_2 \quad (9)$$

gdje je P složeni predikat nad atributima u opisima objekta od zapisa, odnosno, $OD(c_1)\ U\ OD(c_2)$, i ekvivalentni odnos između c_1 i c_2 označava da se ti kandidati smatraju dvostrukim zapisima. Skup takvih pravila gradi domen-specifičnu teoriju jednakosti koja omogućava klasifikaciju zapisa kandidata baziranu na skupu pravila.

Predikat P: P je izraz istinitosti koji može biti napisan u konjuktivnoj normalnoj formi, tj, kao konjukcija (disjunkcija) izraza:

$$P = (term_{1,1} \vee term_{1,2} ...) \wedge ... \wedge (term_{n,1} \vee term_{n2} \vee ...) \quad (10)$$



Slika 2: Metoda sortiranih susjeda

Fokusirajući se na aktuelne izraze, teorija jednakosti omogućava gotovo svaku usporedbu između vrijednosti atributa. Međutim, da bi se napravile smislene usporedbe, ove usporedbe se primijenjuju na zajedničke atribute od c_1 i c_2.

Osnovni koraci u implementaciji metode sortiranih susjeda:

1. Izrada skupa pravila za klasifikaciju zapisa.
2. Određivanje veličine kliznog prozora za usporedbu zapisa.
3. Formirati skup ključeva sortiranja.
4. Sortirati zapise po odabranom ključu sortiranja.
5. Provlačenje kliznog prozora preko zapisa uspoređujući sve parove unutar prozora upotrebom skupa pravila definisanih u koraku 1.
6. Višestruki prolazi. Za određivanje sličnih zapisa koji se razlikuju po ključu sortiranja, kao i kod Standardne metode blokiranja, koristi se metoda višestrukog prolaza, gdje se za svaki prolaz definiše novi ključ sortiranja. U koraku 3 ove metode kreiran je skup ključeva sortiranja i za svakog od njih se ponavljaju koraci 4 i 5.
7. Prijelazno zatvaranje. Na kraju ove metode, kao i kod Standardne metode blokiranja, vrši se prijelazno zatvaranje, nad već identifikovanim dvostrukim zapisama.

4. TESTIRANJE METODA

Obje opisane metode identifikacije dvostrukih zapisa (Standardna metoda blokiranja i Metoda sortiranih susjeda) implementirane su u web aplikaciji koja je razvijena u okviru naučno-istraživaškog rada autora ovog članka. Radni naziv navedene aplikacije je Sistem za identifikaciju višestrukih zapisa (*eng.* The System for Identification of Multiple Records – SIMR). Navedeni softverski alat za testiranje implementiran je u NET tehnologiji i programskom jeziku c# kao web aplikacija, s odgovarajućim konekcijama na testne baze podataka. Testiranje metoda je izvršeno na istoj računarskoj platformi sa Intel(R) Core(TM) i7, 64-bitnim CPU (2.2GHz), 16 GB RAM memorije i Windows 8.1 operativnim sistemom.

Za testiranje kvaliteta metoda identifikacije dvostrukih zapisa koje su opisane u ovom radu korištene su dvije testne baze: FEBRL i CEETISERX. Baza podataka FEBRL je sintetička, dok baza podataka CEETISERX predstavlja realne podatke. Iz baze podataka FEBRL za testiranje je korištena tablica (skup podatka) Dataset1[1]. Ova tablica sadrži sadrži 1000 zapisa od kojih je 500 dvostrukih, odnosno svaki zapis ima svoga duplikata. Iz baze podataka CEETISERX za testiranje je korištena tablica (skup podatka) Articles[2]. Ova tablica sadrži sadrži 2000 zapisa od kojih su 24 dvostruka zapisa.

Analiza kvaliteta rezultata vrši se upotrebom standardnih mjera: *Tačnost* (*eng*. Recall), *Preciznost* (*eng*. Precision) i *F-skor* (*eng.* F-score) [11]. Formule za izračunavanje ovih mjera su:

[1] http://bih5.net/nir/Febrl-Dataset1.zip
[2] http://bih5.net/nir/Citeeserx-Articles.zip

$$Preciznost = \frac{tp}{tp + fp} \quad (11)$$

$$Tačnost = \frac{tp}{tp + fn} \quad (12)$$

$$F - skor = 2 \cdot \frac{Preciznost \cdot Tačnost}{Preciznost + Tačnost} \quad (13)$$

gdje je tp = *true positives* (pravilno identifikovani dvostruki zapisi); fp = *false positives* (pogrešno identifikovani kao dvostruki zapisi) i fn = *false negatives* (pogrešno identifikaovani da nisu dvostruki zapisi). Pored navedenih opcija identifikacije sličnih zapisa postoji i opcija tn – *true negative* (pravilno identifikovani da nisu dvostruki zapisi). F-*skor* predstavlja parametar koji daje veću težinu ili *Preciznosti* ili *Tačnosti*. U ovom radu je korištena neutralna parametrizacija, gdje *Preciznost* i *Tačnost* imaju istu težinu što ima za posljedicu da F-*skor* predstavlja njihovu ponderisanu harmonijsku sredinu.

Za dodatnu analizu efikasnosti i efektivnosti primjene prezentiranih metoda nad odabranim tablicama podataka kroz izvršenje metoda evidentira se vrijeme izvršenja pojedinih faza, a također evidentira se i vrsta faze izvršenja. Vrsta faze izvršenja metode može imati dvije vrijednosti:

– Automatic i
– Manual

Automatic znači da je određena faza metode izvršena potpuno automatski od strane računara, dok *Manual* znači da je određena faza metode potpuno izvršena od strane čovjeka (stručnjaka). Tako se vrijeme izvršenja metode izračunava kako ukupno, tako i po vrstama faza izvršenja metode.

4.1. Standardna metoda blokiranja – testiranje

Testiranje počinje odabirom broja prolaza kao i definisanjem ključeva blokiranja za svaki od prolaza. Također neophodno je postaviti pragove sličnosti za ključeve blokiranja i određivanje parametara m_i i u_i.

Prag sličnosti za ključeve blokiranja u Standardnoj metodi blokiranja je vrijednosti 1 (jedan), što znači da će se u istom bloku za usporedbu nalaziti samo zapisi sa istim ključem blokiranja.

Pragovi sličnosti, kao ulazne veličine, za izračun m_i i u_i su dobiveni empirijskim putem za koje je Standardna metoda blokiranja, testirana u sistemu SIMR, davala najbolje rezultate. Također mjera sličnosti *Jaro,* koja se koristi kroz obje metode, empirijski je određena kao najkvalitetnija za odabrane skupove podataka.

Standardna metoda blokiranja kroz svoje izvršavanje ima određene faze koje se mogu izvršavati više puta u zavisnosti koliki je broj definisanih ključeva blokiranja, odnosno broja prolaza. Tako da se za svaki od prolaza izvršavaju sljedeće faze ove metode:

1. **Formiranje ključeva:** Formiranje, dodavanje i sortiranje analizirane tablice po ključu blokiranja definisanog za dati prolaz.
2. **Formiranje blokova:** Na osnovu definisane mjere sličnosti i veličine praga sličnosti za usporedbu ključeva blokiranja vrši se formiranje blokova unutar kojih će se svi zapisi međusobno usporediti.
3. **Izračunavanje $\mathbf{m_i}$:** Na osnovu izračunatih vrijednosti vektora sličnosti između zapisa u kreiranim blokovima izračunava se frekvencija njihovog pojavljivanja na koje se primjenjuje EM algoritam. Uzastopnim iteracijama se izračunavaju m_i koeficijenti za svaki od atributa zapisa.
4. **Izračunavanje $\mathbf{u_i}$:** Na osnovu izračunatih vrijednosti vektora sličnosti između slučajno odabranih parova zapisa izračunava se frekvencija njihovog pojavljivanja na koje se primjenjuje EM algoritam koji izračunava u_i koeficijente za svaki od atributa.
5. **Izračunavanje pragova sličnosti:** Na osnovu koeficijenata m_i i u_i izračunavaju se donji (7) i gornji prag sličnosti (8).
6. **Izračunavanje dvostrukih zapisa:** Vrši se usporedba zapisa unutar blokova. Kao rezultat usporedbe zapisa izračunava se njihov vektor sličnosti, a na osnovu njega se određuje težina vektora. Usporedbom težine vektora sličnosti, s izračunatim donjim i gornjim pragom sličnosti određuje se da li je uspoređivani par zapisa dvostruki, nepoznat ili različit.

4.1.1. SBM – FEBRL(Dataset1) - testiranje

Tablica *Dataset1* sadrži 1000 zapisa od kojih je 500 dvostrukih zapisa.

1. Broj prolaza: 3

   Prolaz 1 - Ključ blokiranja 1:
   surname(1,5)& given_name(1,5)& street_number;
   Prolaz 2 - Ključ blokiranja 2:
   address_1(1,5)& suburb& postcode;
   Prolaz 3 - Ključ blokiranja 3:
   soc_sec_id(1,5) &date_of_birth(1,4)& state;

Notacija: Ukoliko se iza atributa nalaze male zagrade, onda vrijednosti unutar njih određuju koji se dio atributa uključuje u ključ blokiranja.

2. Definisanje ulaznih vrijednosti:

   Mjera sličnosti: Jaro
   Prag sličnosti za ključeve blokiranja: 1;
   Prag sličnosti za mi: 0.75;
   Prag sličnosti za ui: 0.60;

3. Prolazi

Tabela I: SBM – FEBRL(Dataset1) - Prolaz 1

Faza	Rezultat	Vrijeme(ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Dataset1 po ključu blokiranja 1	39.8504 Automatic
2.Formiranje blokova	Formirani blokovi za usporedbu zapisa	78.4236 Automatic
3.Izračun mi	Broj iteracija izračuna mi:6 *m0=0.961 *m1=0.985 *m2=0.976*m3=0.861 *m4=0.779 *m5=0.918 *m6=0.980 *m7=0.961 *m8=0.899 *m9=0.947	98.5725 Automatic
4.Izračun ui	*u[0]=0.168 *u[1]=0.172 *u[2]=0.170 *u[3]=0.148 *u[4]=0.134 *u[5]=0.160 *u[6]=0.171 *u[7]=0.168 *u[8]=0.155 *u[9]=0.165	128.4715 Automatic
5. Pragovi sličnosti	$T\lambda$ -Donji Prag:0.685 $T\mu$ -Gornji Prag:12.786	0.5724 Automatic
6.Izračun dvostrukih zapisa	Broj parova zapisa: 212 Broj dvostrukih zapisa:197 Broj nepoznatih:15	49.9634 Automatic

Tabela II: SBM – FEBRL(Dataset1) - Prolaz 2

Faza	Rezultat	Vrijeme(ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Dataset1 po ključu blokiranja 2	63.8441 Automatic
2.Formiranje blokova	Formirani blokovi za usporedbu zapisa	66.9703 Automatic
3.Izračun mi	Broj iteracija izračuna mi:15 *m0=0.857 *m1=0.919 *m2=0.786 *m3=0.982 *m4=0.782 *m5=0.991 *m6=1 *m7=0.977 *m8=0.928 *m9=0.955	954.7368 Automatic
4.Izračun ui	*u[0]=0.152 *u[1]=0.170 *u[2]=0.171 *u[3]=0.208 *u[4]=0.170 *u[5]=0.208 *u[6]=0.211 *u[7]=0.204 *u[8]=0.193 *u[9]=0.204	128.2555 Automatic
5. Pragovi sličnosti	$T\lambda$ -Donji Prag:-Infinity $T\mu$ -Gornji Prag:8.759	0.4753 Automatic
6.Izračun dvostrukih zapisa	Broj parova zapisa:269 Broj dvostrukih zpisa:253(166[3]) Broj nepoznatih:16(15)	213.4395 Automatic

[3] U malim zagradama je naveden broj novih dvostrukih parova zapisa u odnosu na predhodne prolaze.

Tabela III: SBM – FEBRL(Dataset1) - Prolaz 3

Faza	Rezultat	Vrijeme(ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Dataset1 po ključu blokiranja 3	75.9178 Automatic
2.Formiranje blokova	Formirani blokovi za usporedbu zapisa	94.3961 Automatic
3.Izračun mi	Broj iteracija izračuna mi:7 *m0=0.761 *m1=0.841 *m2=0.816 *m3=0.912 *m4=0.791 *m5=0.919 *m6=0.992 *m7=0.987 *m8=0.967 *m9=1	241.4029 Automatic
4.Izračun ui	*u[0]=0.218 *u[1]=0.238 *u[2]=0.233 *u[3]=0.258 *u[4]=0.224 *u[5]=0.263 *u[6]=0.284 *u[7]=0.283 *u[8]=0.277 *u[9]=0.286	126.2928 Automatic
5.Pragovi sličnosti	$T\lambda$ -Donji Prag:-Infinity $T\mu$ -Gornji Prag:3.788	1.1961 Automatic
6.Izračun dvostrukih zapisa	Broj parova zapisa:402 Broj dvostrukih zapisa: 397(82) Broj nepoznatih:5(3)	274.5708 Automatic

Ukupan broj dvostrukih zapisa nakon tri prolaza je 458. Broj parova zapisa za dodatnu ekspertizu: 20.

4. Nakon što su izvršena tri prolaza pristupa se analizi identifikovanih parova zapisa za dodatnu ekspertizu. Ekspertiza se vrši manuelno od strane čovjeka. Rezultat:

   Broj dvostrukih zapisa kroz dodatnu ekspertizu: 20.

   Vrijeme: 278836.5994 ms Vrsta: Manual

5. Zadnja faza u identifikaciji novih dvostrukih zapisa je prijelazno zatvaranje.

   Rezultat prijelaznog zatvaranja: 0.

   Vrijeme: 710.4796 ms Vrsta: Automatic

6. Konačan rezultat primjene Standardne metode blokiranja nad tablicom Dataset1 je 478 dvostrukih zapisa.

7. Standardne i dodatne mjere kvaliteta

   tp=478 fp=0 fn=22
   Preciznost=1 Tačnost=0.956
   F-skor=0.977505112474438
   Ukupno vrijeme izvršenja metode (ms): 282184.430
   Ukupno vrijeme za Automatic faze (ms): 3347.831
   Ukupno vrijeme za Manual faze (ms): 278836.60

4.1.2. SBM – CEETISERX(Articles) - testiranje

Tablica Articles sadrži 2000 zapisa od kojih je 24 dvostrukih zapisa.

1. Broj prolaza: 2

   Prolaz 1 - Ključ blokiranja 1:
   title(1,5)& author(1,5);
   Prolaz 2 - Ključ blokiranja 2:
   journal(1,10)& volume& title(1,5);

2. Definisanje ulaznih vrijednosti:

   Mjera sličnosti: Jaro
   Prag sličnosti za ključeve blokiranja: 1;
   Prag sličnosti za mi: 0.75;
   Prag sličnosti za ui: 0.60;

3. Prolazi

Tabela IV: SBM – CEETISERX(Articles) - Prolaz 1

Faza	Rezultat	Vrijeme(ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Articles po ključu blokiranja 1	158.739 Automatic
2.Formiranje blokova	Formirani blokovi za usporedbu zapisa	186.8684 Automatic
3.Izračun mi	Broj iteracija izračuna mi:3 *m0=1 *m1=1 *m2=1 *m3=1 *m4=1 *m5=0.904 *m6=0.0952 *m7=0.952 *m8=1 *m9=1	121.997 Automatic
4.Izračun ui	*u[0]=0.0366 *u[1]=0.0221 *u[2]=0.0202 *u[3]=0.0211 *u[4]=0.0221 *u[5]=0.0211 *u[6]=0.0019 *u[7]=0.0211 *u[8]=0.0346 *u[9]=0.0366	241.4379 Automatic
5. Pragovi sličnosti	$T\lambda$ -Donji Prag:-Infinity $T\mu$ -Gornji Prag:-Infinity	9.499 Automatic
6.Izračun dvostrukih zapisa	Broj parova zapisa:38 Broj dvostrukih zapisa:21 Broj nepoznatih:0	3.3627 Automatic

Tabela V: SBM – CEETISERX(Articles) - Prolaz 2

Faza	Rezultat	Vrijeme(ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Articles po ključu blokiranja 2	262.6803 Automatic
2.Formiranje blokova	Formirani blokovi za usporedbu zapisa	153.3202 Automatic
3.Izračun mi	Broj iteracija izračuna mi:10 *m0=1 *m1=1 *m2=1 *m3=1 *m4=1 *m5=0.904 *m6=0.095 *m7=0.952 *m8=1 *m9=1	96.0719 Automatic
4.Izračun ui	*u[0]=0.0328 *u[1]=0.0203 *u[2]=0.0212 *u[3]=0.0309 *u[4]=0.0328 *u[5]=0.0183 *u[6]=0.0019 *u[7]=0.0203 *u[8]=0.0328 *u[9]=0.0328	131.8085 Automatic
5. Pragovi sličnosti	$T\lambda$ -Donji Prag:-Infinity $T\mu$ -Gornji Prag:-Infinity	0.2034 Automatic
6.Izračun dvostrukih zapisa	Broj parova zapisa:34 Broj dvostrukih zpisa:21(0) Broj nepoznatih:0(0)	213.4395 Automatic

Ukupan broj dvostrukih zapisa nakon dva prolaza je 21. Broj parova zapisa za dodatnu ekspertizu: 0.

4. Kroz prethodno opisana dva prolaza nije identificiran niti jedan par zapisa kao moguće podudaran ili nepoznat. Rezultat:

   Broj dvostrukih zapisa kroz dodatnu ekspertizu: 0.
   Vrijeme: 0 ms Vrsta: Manual

5. Rezultat broja prijelaznog zatvaranje je: 0

   Vrijeme: 1.3617 ms Vrsta: Automatic

6. Konačan rezultat primjene Standardne metode blokiranja nad tablicom Articles je 21 identifikovani dvostruki zapis.

7. Standardne i dodatne mjere kvaliteta

   tp=21 fp=0 fn=3
   Preciznost=1 Tačnost=0.875
   F-skor= 0.933333333333333
   Ukupno vrijeme izvršenja metode (ms): 1580.789
   Ukupno vrijeme za Automatic faze (ms): 1580.789
   Ukupno vrijeme za Manual faze (ms): 0.00

4.2. Metoda sortiranih susjeda – testiranje

Testiranje počinje određivanjem veličine pragova operatora sličnosti i izradom skupa pravila za klasifikaciju zapisa korišćenjem predhodno definisanih operatora. Nakon toga se određuje veličina kliznog prozora za usporedbu zapisa i broj prolaza kao i definisanjem ključeva sortiranja za svaki od prolaza.

Metoda sortiranih susjeda kroz svoje izvršavanje ima određene faze koje se mogu izvršavati više puta u zavisnosti koliki je broj definisanih ključeva sortiranja, odnosno broja prolaza. Tako se za svaki od prolaza izvršavaju sljedeće faze ove metode:

1. **Formiranje ključeva:** Formiranje, dodavanje i sortiranje analizirane tablice po ključu sortiranja definisanog za dati prolaz.
2. **Izračunavanje dvostrukih zapisa:** Povlačenjem kliznog prozora preko sortiranih zapisa tablice koja se analizira po ključu sortiranja vrši se usporedba parova zapisa unutar kliznog prozora, primjenom definisanih pravila na osnovu kojih se donosi odluka da li je analizirani par zapisa isti ili ne.

4.2.1. SNM – FEBRL(Dataset1) - testiranje

Tablica Dataset1 sadrži 1000 zapisa od kojih je 500 dvostrukih zapisa.

1. Definisanja operatora sličnosti i pravila sličnosti.

   Operatori sličnosti sa svojim donjim vrijednostima:
   Isto = 1;
   Veoma slično = 0.85;
   Slično = 0.70;
   Broj pravila: 3

   Pravilo 1:
   Slično(given_name) AND *Veoma slično*(suburb) AND *Isto*(postcode)

   Pravilo 2:
   Veoma slično(given_name) AND *Isto*(surname) AND *Isto*(state) AND *Isto*(soc_sec_id)

Pravilo 3:
Veoma slično(adress_1) AND *Isto*(soc_sec_id) AND *Isto*(suburb)
Vrijeme (ms): 378611.2653 Vrsta: Manual

2. Definisanje ulaznih vrijednosti:

   Mjera sličnosti: Jaro
   Veličina kliznog prozora = 5;

3. Definisanje ključeva sortiranja po prolazima.

   Broj prolaza: 3
   Prolaz 1 - Ključ sortiranja 1:
   surname(1,5) & given_name(1,5);
   Prolaz 2 - Ključ sortiranja 2:
   suburb & state;
   Prolaz 3 - Ključ sortiranja 3:
   soc_sec_id(1,5) & postcode;

4. Prolazi

Tabela VI: SNM – FEBRL(Dataset1) - Prolaz 1

Faza	Rezultat	Vrijeme (ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Dataset1 po ključu sortiranja 1	72.626 Automatic
2. Izračun dvostrukih zapisa	Broj parova zapisa: 3990 Broj dvostrukih zapisa: 396	739.2351 Automatic

Tabela VII: SNM – FEBRL(Dataset1) - Prolaz 2

Faza	Rezultat	Vrijeme (ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Dataset1 po ključu sortiranja 2	56.2037 Automatic
2. Izračun dvostrukih zapisa	Broj parova zapisa: 3990 Broj dvostrukih zapisa: 437 (76)	661.9261 Automatic

Tabela VIII: SNM – FEBRL(Dataset1) - Prolaz 3

Faza	Rezultat	Vrijeme (ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Dataset1 po ključu sortiranja 3	85.0539 Automatic
2. Izračun dvostrukih zapisa	Broj parova zapisa: 3990 Broj dvostrukih zapisa: 416 (2)	1177.7864 Automatic

Nakon tri prolaza ukupan broj dvostrukih zapisa je: 474.

5. Prijelazno zatvaranje

   Broj prijelaznih zatvaranja: 0
   Vrijeme: 733.7054 ms Vrsta: Automatic

6. Konačan rezultat primjene Metode sortiranih susjeda nad tablicom Dataset1 je 474 dvostrukih zapisa.

7. Standardne i dodatne mjere kvaliteta

   tp=474 fp=0 fn=26
   Preciznost=1 Tačnost=0.947895791583166
   F-skor=0.973251028806584
   Ukupno vrijeme izvršenja metode (ms): 382137.80
   Ukupno vrijeme za Automatic faze (ms): 3526.54
   Ukupno vrijeme za Manual faze (ms): 378611.27

4.2.2. SNM – CEETISERX(Articles) - testiranje

Tablica Articles sadrži 2000 zapisa od kojih je 24 dvostrukih zapisa.

1. Definisanja operatora sličnosti i pravila sličnosti

   Operatora sličnosti sa svojim donjim vrijednostima:
   Isto = 0.95;
   Veoma slično = 0.80;
   Slično = 0.70;
   Broj pravila: 2
   Pravilo 1:
   Veoma slično(title) AND *Veoma slično*(author)
   Pravilo 2:
   Slično(journal) AND *Veoma slično* (year) AND *Slično*(title) AND *Slično*(author)
   Vrijeme (ms): 232391.0516 Vrsta: Manual

2. Definisanje ulaznih vrijednosti:

   Mjera sličnosti: Jaro
   Veličina kliznog prozora = 5;

3. Definisanje ključeva sortiranja po prolazima

   Broj prolaza: 2
   Prolaz 1 - Ključ sortiranja 1:
   title(1,5) & author(1,5);
   Prolaz 2 - Ključ sortiranja 2:
   author(1,5) & title(1,5);

4. Prolazi

Tabela IX: SNM – CEETISERX(Articles) - Prolaz 1

Faza	Rezultat	Vrijeme (ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Articles po ključu sortiranja 1	137.189 Automatic
2. Izračun dvostrukih zapisa	Broj parova zapisa: 7990 Broj dvostrukih zapisa: 21	1066.7281 Automatic

Tabela X: SNM – CEETISERX(Articles) - Prolaz 2

Faza	Rezultat	Vrijeme (ms) i Vrsta faze
1.Formiranje ključeva	Sortirana tablica Articles po ključu sortiranja 2	162.8971 Automatic
2. Izračun dvostrukih zapisa	Broj parova zapisa: 7990 Broj dvostrukih zapisa: 22(1)	880.0137 Automatic

Nakon dva prolaza ukupan broj dvostrukih zapisa je: 22.

5. Prijelazno zatvaranje

   Broj prijelaznih zatvaranja: 0
   Vrijeme: 1.2988 ms Vrsta: Automatic

6. Konačan rezultat primjene Metode sortiranih susjeda nad tablicom Articles su 22 dvostruka zapisa.

7. Standardne i dodatne mjere kvaliteta

tp=22 fp=0 fn=2
Preciznost=1 Tačnost=0.91304347826087
F-skor=0.954545454545454
Ukupno vrijeme izvršenja metode (ms): 234639.18
Ukupno vrijeme za Automatic faze (ms): 2248.13
Ukupno vrijeme za Manual faze (ms): 232391.05

5. ANALIZA REZULTATA

Analiza rezultata testiranih metoda za identifikaciju dvostrukih zapisa prezentiranih u ovom radu izvršena je usporedbom rezultata po bazama podataka, odnosno tablicama podataka nad kojima je testiranje metoda izvršeno. U tabelama XI i XII prikazani su usporedni rezultati testiranja metoda. Strogo se vodilo računa da su obje metode prilikom testiranja imale iste uslove u smislu broja prolaza, kvaliteta ključeva blokiranja ili sortiranja.

Identifikacija dvostrukih zapisa vrši se u zavisnosti od metode do metode kroz nekoliko faza testiranja. Zajedničko za obje metode je da se na kraju svakog od prolaza identificiraju dvostruki zapisi. Također, zajednička je mogućnost da se kroz fazu prijelaznog zatvaranja identificiraju novi dvostruki zapisi. Karakteristično za Standardnu metodu blokiranja je da se kroz prolaze identifikuju parovi zapisa kao moguće podudarni ili nepoznati. Za ovakve parove zapisa je potrebna dodatna ekspertiza na osnovu koje je moguće identifikovati nove dvostruke zapise. Faze dodatne ekspertize u Metodi sortiranih susjeda nema (u Tabelama XI i XII polja sivo osjenčena).

Podaci na osnovu kojih se mjeri efektivnost metoda su *Preciznost*, *Tačnost* i *F-skor*. Na osnovu ovih parametara, kao i parametra *tp*, prikazanih u Tabeli XII, može se uočiti da Standardna metoda blokiranja ima veću efektivnost nad Febrl-Dataset1 skupom podataka. Također, na osnovu rezultata prikazanih u Tabeli XII može se uočiti da nad skupom podataka Citeeserx – Atricles veću efektivnost ima Metoda sortiranih susjeda.

Pored indikatora efektivnosti testiranih metoda, mjereni su i indikatori efikasnosti. Efikasnost testiranih metoda je mjerena preko ukupno utrošenog vremena za izvršenje testiranih metoda, ali i preko utrošenog vremena po vrstama faza izvršenja. Tako, pored ukupnog vremena evidentirana su i ukupna vremena izvršenja po vrstama faza: *Automatic* i *Manual*. Na osnovu ovih vremena, pored same dužine vremena izvršenja pojedine metode, može se validirati i uloga čovjeka-eksperta (*Manual*) u izvršenju pojedinih metoda.

Tako u testiranju metoda nad Febrl-Dataset1 skupom podataka, obje metode imaju približan odnos učešća utrošenog vremena faza izvršenja označenih kao *Automatic* i *Manual*. Razlog ovome je relativno velik broj parova zapisa koji su analizirani kroz dodatnu ekspertizu u Standardnoj metodi blokiranja što je gotovo u potpunosti odgovoralo vremenu potrebnom za definisanje pravila za klasifikaciju parova zapisa kod Metode sortiranih susjeda.

Tabela XI: Usporedba rezultata testiranih metoda za bazu podataka FEBRL(Dataset1)

Indikatori\Metode	Standardna metoda blokiranja	Metoda sortiranih susjeda
Broj prolaza	3	3
Broj dvostrukih zapisa kroz prolaze	458	474
Broj dvostrukih zapisa za dodatnu ekspertizu	20	
Broj dvostrukih zapisa kroz dodatnu ekspertizu	20	
Broj dvostrukih zapisa nakon dodatne ekspertite	478	
Broj dvostrukih zapisa kroz prijelazno zatvaranje	0	0
Ukupan broj dvostrukih zapisa	478	474
tp (true positive)	478	474
fp (false pozitive)	0	0
fn (false negative)	22	26
Preciznost	1	1
Tačnost	0.956	0.947
F-skor	0.977	0.973
Vrijeme – Automatic ms)	3347.831 **1.19%**	3526.54 **0.92%**
Vrijeme – Manual (ms)	278836.60 **98.81%**	378611.27 **99.08%**
Ukupno vrijeme (ms)	282184.430	382137.80
Vrijeme po tp (ms)	590.344	806.197

Tabela XII: Usporedba rezultata testiranih metoda za bazu podataka CEETISERX (Articles)

Indikatori\Metode	Standardna metoda blokiranja	Metoda sortiranih susjeda
Broj prolaza	2	2
Broj dvostrukih zapisa kroz prolaze	21	22
Broj dvostrukih zapisa za dodatnu ekspertizu	0	
Broj dvostrukih zapisa kroz dodatnu ekspertizu	0	
Broj dvostrukih zapisa nakon dodatne ekspertite	21	
Broj dvostrukih zapisa kroz prijelazno zatvaranje	0	0
Ukupan broj dvostrukih zapisa	21	22
tp (true positive)	21	22
fp (false pozitive)	0	0
fn (false negative)	3	2
Preciznost	1	1
Tačnost	0.875	0.913
F-skor	0.933	0.954
Vrijeme – Automatic (ms)	1580.789 **100 %**	2248.13 **0.96 %**
Vrijeme – Manual (ms)	0.00 **0 %**	232391.05 **99.14 %**
Ukupno vrijeme (ms)	1580.789	234639.18
Vrijeme po tp (ms)	75.27569	10665.417

Kod testiranja metoda nad Citeeserx-Articles skupom podataka uočljiva je velika razlika po metodama u učešću utrošenog vremena faza izvršenja označenih kao *Automatic* i *Manual*. Kod Standardne metode blokiranja nema učešće čovjeka-eksperta u izvršenju ove metode s obzirom da nisu identifikovani parovi zapisa za dodatnu ekspertizu. S druge strane, učešće čovjeka-eksperta kod Metode sortiranih susjeda je potpuno preovladajuće kao i kod primjene ove metode nad Febrl-Dataset1 skupom podataka.

S obzirom da identifikacija većeg broja *tp* parova zapisa zahtijeva i veće vrijeme izvršenja cjelokupne metode, uveden je još jedan indikator koji daje relativno utrošeno vrijeme izvršenja metode po tačno identifikovanom duplom zapisu (*Vrijeme po tp*). Što je ovaj podatak manji, efikasnost metode s aspekta utrošenog vremena i utrošenih resursa je veća. U oba slučaju testiranja metoda nad odabranim skupovima podataka Standardna metoda blokiranja ima manje vrijeme izvršenja po *tp*, iz čega proizilazi da ima i veću efikasnost.

6. ZAKLJUČAK

Na osnovu prikazanih usporednih rezultata testiranja Standardne metode blokiranja i Metode sortiranih susjeda može se zaključiti da je efektivnost obiju metoda na približnom nivou. Međutim, kroz evidentirane razlike efektivnosti ovih metoda po odabranim skupovima podataka može se potvrditi da Standardna metoda blokiranja identifikuje veći broj *tp* (tačno identifikovanih) dvostrukih zapisa u skupovima podataka koji imaju veći broj dvostrukih zapisa, a Metoda sortiranih susjeda daje bolje rezultate kod skupova podataka s relativno manjim brojem dvostrukih zapisa.

Što se tiče efikasnosti metoda, može se zaključiti da na nju najviše utiče stepen učešća čovjeka u njihovom izvršenju. Tako, za prvi testirani skup podataka učešće čovjeka, u kontekstu utrošenog vremena, je približno isto iz čega proističe da je i sama efikasnost metoda približno ista. Kod drugog skupa podataka razlika utrošenog vremena od strane čovjeka po metodama je značajna iz čega je proistekla veća efikasnost Standardne metode blokiranja. Međutim, ovdje je potrebno biti oprezan iz razloga što je učešće čovjeka s aspekta utrošenog vremena predvidivo (fiksno) kod Metode sortiranih susjeda. Takođe kod ove metode procenat utrošenog vremena od strane čovjeka se smanjuje povećanjem skupa podataka nad kojim se metoda izvršava. Kod Standardne metode blokiranja utrošeno vrijeme od strane čovjeka je nepredvidivo i izvršenje metode može biti bez ikakvog učešća čovjeka, pa sve do toga da je učešće utrošenog vremena od strane čovjeka gotovo stoprocentno u zavisnosti od toga koliki je broj parova zapisa koji su identifikovani kao moguće podudarni ili nepoznati.

Stoga bi u cilju eventualnog poboljšanja opisanih metoda, posebno s aspekta efikasnosti, trebalo raditi na razvoju i ugradnji određenih znanja u opisane metode, koje bi smanjilo ili potpuno isključilo ulogu čovjeka u njihovom izvršavanju.

LITERATURA

[1] GK Tayi and DP Ballou. Examining data quality. In Communications of the ACM, Volume 41 Issue 2, pages 54-57, 1998.

[2] F.Naumann and M.Herschel. An Introduction to Duplicate Detection, Morgan&Claypool Publishers, 2010.

[3] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In Advances in Knowledge Discovery and Data Mining, pages 1-36, 1996.

[4] R.J. Branchman and T. Anand. The process of knowledge discovery in databases: A human-centered approach. In Advances in Knowledge Discovery and Data Mining, pages 97-158, 1996.

[5] F. Simoudis, B. Levesey, and R. Kerber. Using recon for data cleansing. In Proceedings KDD, pages 282-287, 1995.

[6] M. Lee, T. Ling. Cleansing data for mining and warehousing. In Proceedings DEXA, pages 751-760, 1999.

[7] J.I. Maletic and A. Marcus. Data cleansing: Beyond integrity checking. In Proceedings of Information Quality (IQ), 2000.

[8] A. Maydanchik. Challenges of e_cient data cleansing. In DM Review, 1999.

[9] L. Moss. Data cleansing: A dichotomy of data warehousing. In DM Review, 1998.

[10] Matthew A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. Journal of the American Statistical Association Vol. 84, No. 406 , pages 414-420, 1989.

[11] R.J. Branchman and T. Anand. The process of knowledge discovery in databases: A human-centered approach. In Advances in Knowledge Discovery and Data Mining, pages 97-158, 1996.

[12] Mauricio A. Hernández and Salvatore J. Stolfo. Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. In Data Mining and Knowledge Discovery Volume 2 Issue 1, pages 9 – 37, 1998.

BIOGRAFIJA

Đulaga Hadžić diplomirao je na Elektrotehničkom fakultetu Univerziteta u Sarajevu 1997. godine. Naučni stepen magistra tehničkih nauka stekao je na Fakultetu elektrotehnike Univerziteta u Tuzli, 2010. godine. Trenutno radi na svojoj PhD tezi iz oblasti kvaliteta podataka. Njegova polja interesovanja su: softverski inženjering, informacioni sistemi, kvalitet podataka i vještačka inteligencija.

Nermin Sarajlić diplomirao je na Elektrotehničkom fakultetu Univerziteta u Tuzli 1987. godine. Naučni stepen magistra nauka stekao je na Fakultetu elektrotehnike i mašinstva Univerziteta u Tuzli 1997. godine, a naučni stepen doktora nauka stekao je na Fakultetu elektrotehnike Univerziteta u Tuzli 2002. godine. Uža oblast interesovanja je proračun: spregnutih elektromagnetnih-temperaturnih polja; kriptografija, kriptoanaliza.

Jasmin Malkić je 1998. godine diplomirao na Fakultetu elektrotehnike i računarstva u Zagrebu, a magistrirao 2003. na Fakulteti za elektrotehniko, računalništvo in informatiko u Mariboru. Trenutno radi za OpenLink International GmbH u Berlinu, Njemačka. Uža područja interesovanja su mu aplikacijski serveri, sigurnost u informacijskim sistemima, medicinska informatika, te informatička podrška tržištima energenata.