

Proqram Məhsullarında Xətaların Aşkarlanması Metodu

Fərqanə Abdullayeva

AMEA İnformasiya Texnologiyaları İnstitutu, Bakı, Azərbaycan
farqana@iit.ab.az

Xülasə— Proqram təminatının keyfiyyətinin qiymətləndirilməsində mühüm amillərdən biri proqram modullarında xətaların aşkarlanmasıdır. Təqdim olunan işdə xətaların aşkarlanması üçün klassifikatorlar ansamblına əsaslanan model təklif olunur. Xətaların aşkarlanması kriteriyalara qoyulmuş sərhəd qiymətləri əsasında təmin edilir. Burada xətaları aşkarlamaq üçün PROMISE proqram mühəndisliyi reyestrindən əldə edilmiş KC2 verilənlər bazasından istifadə edilmişdir. Eksperimentlər Matlab və Weka proqramları vasitəsi ilə aparılmışdır. Metodun eksperimental tədqiqi nəticəsində 0.88% aşkarlama dəqiqliyi əldə edilmişdir.

Keywords— *proqram mühəndisliyi, xəta, maşın təlimi, Isolation Forest, Random Forest*

I. GİRİŞ

Proqram sistemlərinin mürəkkəb quruluşa malik olması bu sistemlərdə yol verilmiş xətaların aşkarlanması prosesini olduqca çətinləşdirir. Layihə menecerləri proqram məhsullarında xətaları proqnozlaşdırmaq metodlarından istifadə etməklə adətən bu xətaları sistem test edilməzdən əvvəl aşkarlaya bilər. Proqram məhsullarında xətaların aşkarlanması üçün qərar ağacı, loqarifmik reqresiya, qeyri-səlis məntiq, neyron şəbəkələri, genetik alqoritm kimi məşhur aparatlara əsaslanan vasitələr təklif edilmişdir [1]. Bu yanaşmaların test edilməsi prosesində verilənlər bazasında kənarçıxmaların (outlier) olması faktı nəzərə alınmamışdır. Bu səbəbdən verilənlər bazasında kənarçıxmaya olduqca mövcud metodların aşkarlama dəqiqliyi olduqca azalır.

Kənar element (outlier) şablonun real elementlərindən fərqli qiymətlər alan elementlərdir [2]. Yəni verilənlər obyektinin qiyməti əlamət üçün təyin edilmiş sərhəd qiymətlərini aşarsa lakin verilənlər obyektinin sinifi xətasız kimi təsnifatlaşdırılırsa, həmin element kənar element kimi qəbul olunur. Və əksinə əgər verilənlər obyektinin qiyməti əlamət üçün təyin edilmiş sərhəd qiymətlərindən aşağıdırsa lakin verilənlər obyektinin sinifi xətalı kimi təsnifatlaşdırılırsa, həmin element kənar element kimi qəbul olunur.

Sistemin proqram məhsullarında yol verilmiş xətaların aşkarlanmasında dəqiqliyi yüksəltmək üçün klassifikatorların birləşdirilməsi zəruri hesab olunur [3]. Təqdim olunan məqalədə proqram məhsullarında xətaların aşkarlama dəqiqliyini artırmaq üçün klassifikatorlar ansamblına əsaslanan yanaşma təklif edilir. Xəta verilənləri - proqram məhsullarının qiymətləndirilməsi üçün yaradılmış verilənlər bazalarında (məsələn, KC2) *xətalı* və *xətasız* kimi iki sinifdə təsvir edilmişdir [4].

Təklif edilmiş metodun eksperimental yoxlanması WEKA proqramında və KC2 verilənlər bazasının üzərində aparılmışdır.

II. XƏTALARIN AŞKARLANMASI ÜÇÜN ƏLAMƏTLƏR VEKTORU

Proqram məhsullarında xətaları aşkarlamaq üçün aşağıdakı əlamətlər və uyğun sərhəd qiymətləri istifadə edilir [4,5]:

- Kodun uzunluğu (lines of code – LoC). Kodun sətirlərinin ümumi sayı nəzərdə tutulur;
- Dövrü mürəkkəblik (cyclomatic complexity – CC). Proqramın struktur cəhətdən mürəkkəbliyinin ölçülməsi meyarıdır. Proqramın axın qrafındakı (flow graph) asılı olmayan yolların ümumi sayı ilə ölçülür;
- Əsas mürəkkəblik (essential complexity – EsC). Proqramda “if-then-else” şərt və “while” dövr prosesinin yaxşı strukturlaşdırılmasını tələb edir. Proqramda “goto” operatorundan istifadə edilməsi EsC qiymətini artıran amildir;
- Unikal operator (unique operator – UOpr). Proqramda unikal operatorların ümumi sayıdır. Əgər eyni bir operator bir neçə dəfə istifadə edilibsə, bu operator bir dəfə nəzərə alınır;
- Unikal operand (unique operand – UOpnd). Unikal operandların ümumi sayıdır. Əgər eyni bir operand bir neçə dəfə istifadə edilibsə, bu operand bir dəfə nəzərə alınır;
- Ümumi operator (total operator – TOpr). Modulda rast gəlinən operatorların ümumi sayı. Operatorlara misal olaraq məntiqi operatorları, riyazi əməliyyatları və hər bir açar sözü (return, sizeof və s.) göstərmək olar;
- Ümumi operand (total operand – TOpnd). Modulda rast gəlinən operandların ümumi sayı nəzərdə tutulur.

İntegrasiya olunmuş Proqram Metrikaları Korporasiyası (Integrated Software Metrics, Inc.) xətaları aşkarlamaq üçün metrikalara sərhəd qiymətləri müəyyən etmişdir (cədvəl 1) [6].

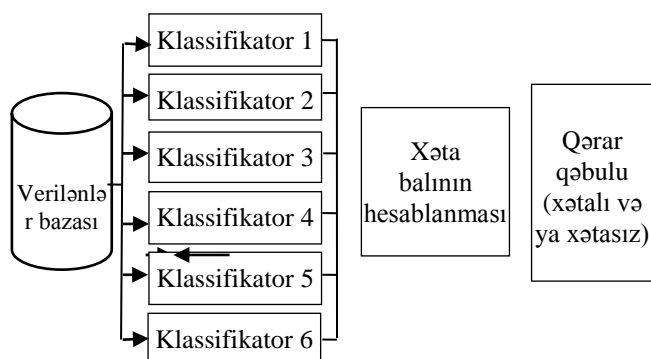
CƏDVƏL 1. METRİKALAR VƏ SƏRHƏD QIYMƏTLƏRİ

Metrika	Qısaltma	Sərhəd qiyməti
Kodun uzunluğu	LoC	65
Dövrü mürəkkəblik	CC	10
Əsas mürəkkəblik	EsC	4
Unikal operator	UOpr	40
Unikal operand	UOpnd	25
Ümumi operator	TOpr	125
Ümumi operand	TOpnd	70

Təqdim olunan məqalədə xətalara aşkarlanması üçün cədvəl 1-də verilən sərhəd qiymətləri istifadə edilir.

III. XƏTALARIN AŞKARLANMASI METODU

Proqram məhsullarında xətalara aşkarlanması üçün təklif edilmiş yanaşmanın qərar qəbulu sisteminin arxitekturu şəkil 1-də təsvir edilmişdir.



Şəkil 1. Proqram məhsullarında xətalara aşkarlanması modeli

Şəkil 1-dən görüldüyü kimi qərar qəbulu sistemi altı klassifikatordan ibarətdir. Burada bu klassifikatorların hər birinin fərdi qərarı kollaborativ qərar vermək üçün birləşdirilir.

Bu klassifikatorları birləşdirməkdə məqsəd sistemin proqram məhsullarında yol verilmiş xətalara aşkarlama keyfiyyətini yüksəltməkdir. Bu prosesi həyata keçirmək üçün hər bir klassifikator təsnif edilmiş verilənlər vasitəsi ilə öyrədilir.

Fərz edək ki, $C = \{c_i | 1 \leq i \leq n\}$ öyrədilən klassifikatorlardır və $D = \{d_i | 1 \leq i \leq n\}$ sayda qərar qəbulu modeli qurur. Hər bir i -ci qərarqəbuletmə d_i modeli test edilən x_i elementinə binar qiymət (0 və ya 1) verir, $d_i = (x_i) : v_i$. v_i -nin binar qiyməti 1 olduqda x_i elementi xətalı kimi qiymətləndirilir, əks halda xətasız kimi qiymətləndirilir. Belə olduqda xəta balı aşağıdakı düsturla hesablanır:

$$x_i = \frac{\sum_{j=1}^n d_j(x_i)}{n} \quad (1)$$

burada n xəta balının hesablanmasında iştirak edən qərar qəbuletmə modellərinin sayıdır. Verilənlər bazasının elementlərinin sinifi, xətalı və ya xətasız, aşağıdakı düsturla hesablanır:

$$Class(x_i) = \begin{cases} x_i \geq \rho & qüsurlu = 1 \\ \text{əks halda} & qüsursuz = 0 \end{cases} \quad (2)$$

burada ρ - qərar qəbuletmə modellərinin test edilən x_i elementini xətasız kimi qiymətləndirmək üçün verdikləri balın faiz qiymətidir. Məsələn, əgər ρ sərhəd qiyməti 1 götürüldükdə, və əgər iştirak edən qərar qəbuletmə

modellərinin hamısı eyniliklə elementin xətalı olduğunu qəbul etməsə, test edilən element xətalı kimi qəbul olunmayacaqdır.

Misal. Fərz edək ki, öyrədilən beş klassifikator götürülmüşdür $C = \{c_1, c_2, c_3, c_4, c_5\}$, bu klassifikatorlar qərar qəbulu modellərini $D = \{d_1, d_2, d_3, d_4, d_5\}$ qurmaq üçün təsnif edilmiş verilənlərlə öyrədilmişdir, və fərz edək ki, test edilən x_i elementinə qərar qəbuletmə modelləri tərəfindən uyğun olaraq 1, 1, 0, 1, 1 balları verilmişdir. Onda x_i elementinin xəta balı aşağıdakı kimi hesablanır:

$$x_i = \frac{1+1+0+1+1}{5} = \frac{4}{5} = 0.80 \quad (3)$$

burada sərhəd qiymətinin $\rho = 0.6$ qiymətində x_i elementinin xəta olduğu fərz edilərsə, onda ən azı üç qərar qəbuletmə modelinin x_i elementini xəta kimi qiymətləndirməsi kifayət edər ki, element xəta hesab olunsun.

IV. TƏKLİF OLUNAN METODUN EKSPERİMENTAL TƏDQIQI

Eksperimentlərin aparılması üçün NASA KC2 verilənlər bazasından istifadə edilmişdir [4].

Proqram məhsullarında qüsurlu elementlərin aşkarlanması üçün BFTree (Best First Tree), PART, IBk, NNg, IsolationForest, RandomForest alqoritmləri istifadə edilmişdir.

Təklif edilmiş yanaşmanın effektivliyini qiymətləndirmək üçün doğru müsbət aşkarlama əmsalı (true positive rate - TP), yanlış müsbət aşkarlama əmsalı (false positive rate - FP), F-ölçü (F-measure), doğruluq (accuracy), dəqiqlik (precision), tamlıq (recall) parametrləri istifadə edilmişdir. Klassifikasiya prosesində istifadə edilən xətalara matrisi (confusion matrix) cədvəl 2-də təsvir edilmişdir.

CƏDVƏL 2. XƏTALAR MATRİSİ

	Real təsnif olunmuş verilənlər	
	Hə	Yox
Proqnoz edilən təsnif olunmuş verilənlər	TP	FP
	FN	TN

Təklif edilmiş modelin effektivlik parametrləri aşağıdakı kimi hesablanır:

8) Dəqiqlik (*precision*). Çeşidlənmiş davranış sırasında doğru aşkarlanmış normal davranışın faiz dərəcəsi.

$$Precision = TP / (TP + FP) \quad (4)$$

9) Tamlıq (*recall*). Bütün normal davranışlar sırasında aşkarlanmış normal davranışın faiz dərəcəsi.

$$Recall = TP / (TP + FN) \quad (5)$$

10) Yalnız müsbət hallar (*false positive rate - FPR*).

$$FPR = FP / (FP + TN) \quad (6)$$

11) F-ölçü (*F-measure*).

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (7)$$

12) Doğru müsbət hallar (*True Positives – TP*).

$$TPR = TP / positive \quad (8)$$

13) Doğru mənfi hallar (*True Negatives – TN*).

$$TNR = TN / negative \quad (9)$$

14) Yalnış mənfi hallar (*False Negatives – FN*).

$$FNR = FN / (FN + TP) \quad (10)$$

15) Yalnış aşkarlanma əmsalı (*False discovery rate – FDR*).

$$FDR = FP / (FP + TP) \quad (11)$$

16) Doğruluq (*accuracy*).

$$Accuracy = (TP + TN) / (P + N) \quad (12)$$

KC2 verilənlər bazası üzərində aparılmış eksperimentlərin nəticəsi cədvəl 3-də təsvir edilmişdir. Burada müxtəlif metodların KC2 verilənlər bazası üzərində test nəticələri daxil edilmişdir.

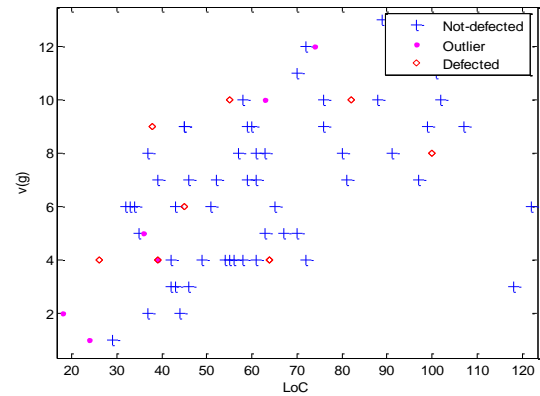
CƏDVƏL 3. TƏKLİF EDİLMİŞ YANAŞMANIN MÖVCUD METODLARLA MÜQAYİSƏSİ (KC2 VERİLƏNLƏR BAZASI) (FAİZLƏ)

	BFTree	J48	Multilayer Perseptron	Klassifikatorlar ansamblı
TP	0.816	0.803	0.711	0.882
FP	0.243	0.243	0.206	0.150
F-measure	0.756	0.748	0.711	0.843
False discovery rate	0.1842	0.197	0.289	0.1184
Acuracy	0.7814	0.7760	0.7596	0.8634
Precision	0.705	0.701	0.711	0.8816
Recall	0.816	0.803	0.711	0.882

Cədvəl 3-dən görüldüyü kimi ayrı-ayrı klassifikatorların nöqsanları aşkarlama dəqiqliyi bütün göstəricilər üzrə klassifikatorlar ansamblının aşkarlama dəqiqliyindən olduqca aşağıdır. Belə ki, TP göstəricisi BFTree, J48, Multilayer Perseptron alqoritmləri üçün uyğun olaraq 0.816, 0.803, 0.711 təşkil etdiyi halda klassifikatorlar ansamblı üçün bu qiymət 0.882 % təkil etmişdir. Bundan əlavə təklif edilmiş metodun yanlıq aşkarlama dəqiqliyi də məqsədəuyğun alınmışdır. Belə ki, klassifikatorlar ansamblında FP-nin qiyməti 0.150%, digər alqoritmlərdə isə bu qiymət olduqca böyük alınmışdır. Bundan əlavə metodun aşkarlama dəqiqliyində də yaxşı nəticə alınmışdır. Belə ki, bu göstərici üzrə BFTree 0.705%, J48 0.701%, Multilayer Perseptron 0.711, klassifikatorlar ansamblı 0.8816 faiz təşkil etmişdir.

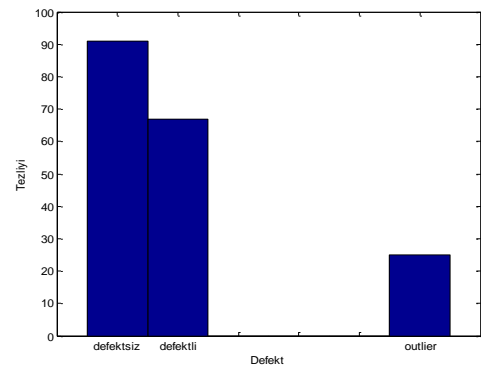
Şəkil 2-də klassifikatorlar ansamblının KC 2 verilənlər bazasının LoC və V(g) parametrlərinə nəzərən nəticəsi vizual

olaraq təsvir olunmuşdur. Şəkilə təsvir olunmuş göy rəngli “+” işarəsi xətasız (not-defected), qırmızı rəngli romb işarəsi xətalı (defected), çəhrayı rəngli nöqtələr kənarçıxmaları göstərir.



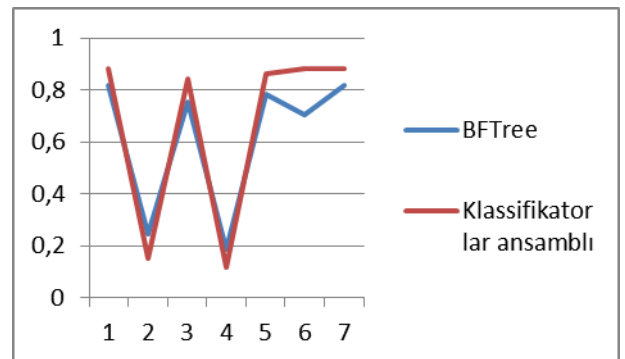
Şəkil 2. KC2 verilənlər bazasının təsnifatlaşdırılması

Şəkil 3-də nöqsanların aşkarlanması prosesinin vizual təsviri verilmişdir. Şəkiləndən görüldüyü kimi təklif edilmiş metodda aşkarlama zamanı çox az sayda kənarçıxma alınmışdır. Bu səbəbdən təklif edilmiş metodun aşkarlama dəqiqliyi digər metodlarla müqayisədə daha yüksək alınmışdır.



Şəkil 3. Xətalın aşkarlanması prosesinin diaqramı

Şəkil 4-də klassifikatorlar ansamblının BFTree alqritmi ilə effektivliyinin müqayisəsi təsvir edilmişdir. Buradan aydın şəkildə görünür ki, klassifikatorlar ansamblı BFTree alqritminə nəzərən yüksək effektivliyə malikdir.



Şəkil 4. Alqoritmlərin effektivliyinin müqayisəsi

Proqram məhsullarında defektlərin aşkarlanması sahəsində eksperimentlərin aparılması üçün CM1, PC!-1107, JM1-10876, KC1-2107 verilənlər bazaları da istifadə oluna bilər.

NƏTİCƏ

Məqalədə açıq verilənlər bazasından istifadə edərək proqram məhsullarında xətalərin aşkarlanması üçün üsul təklif olunmuşdur. Eksperimentlərin nəticəsi təklif edilmiş metodun bir sıra qiymətləndirmə metrikalar üzrə aşkarlama dəqiqliyinə görə məşhur BFTree, J48 və MultilayerPerseptron alqoritmlərindən üstünlük təşkil etdiyini göstərmişdir. Burada istifadə olunan verilənlər bazasının sinifləri əvvəlcədən məlum olduğundan təklif edilmiş modeldə öyrədilən (supervised) maşın təlimi alqoritmləri istifadə edilmişdir.

ƏDƏBİYYAT

- [1] C. Catal, B. Diri, “Investigating the effect of dataset size, metrics sets and feature selection techniques on software fault prediction problem,” *Information Sciences*, 2009, vol. 179, no. 8, pp. 1040-1058.
- [2] J.V. Hodge, J. Austin, “A survey of outlier detection methodologies,” 2004, vol. 22, no. 2, pp 85-126.
- [3] Р.М. Алыгулиев, Я.Н. Имамвердиев, Ф.Д. Абдуллаева, “Обнаружение аномалий в облачных BIG DATA данных,” XIII-ая международная конференция «Опτικο-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации», 2017, в печати.
- [4] KC2, NASA metrics data program, PROMISE software engineering repository, 2004, <http://promise.site.uottawa.ca/SERepository/datasets-page.html>.
- [5] P. Hea, B. Lic, X. Liua, J. Chenb, Y. Mab, “An empirical study on software defect prediction with a simplified metric set,” *Information and Software Technology*, 2015, vol. 59, pp. 170-190.
- [6] C. Cagatay, D. Banu, “A fault detection strategy for software projects,” *Technical Gazette*, 2013, vol. 20, no. 1, 7 p.