Bulud İnfrastrukturunun Keyfiyyət Göstəricilərində Anomaliyaların Real Zamanda Aşkarlanması Metodu

Rasim Əliquliyev[1], Ramiz Alıquliyev[2], Fərqanə Abdullayeva[3]
[1,2,3]AMEA İnformasiya Texnologiyaları İnstitutu, Bakı, Azərbaycan
[1]*rasim@science.az*, [2]*r.aliguliyev@gmail.com*, [3]*farqana@iit.ab.az*

***Xülasə*— Bulud istifadəçilərə sorğu əsasında miqyaslanan resurslar təqdim edən texnologiyadır. Bulud infrastrukturuna istənilən qurğu vasitəsi ilə daxil olmağın mümkünlüyü və açıq şəbəkələrdən istifadə etməsi bu mühiti müxtəlif tipli kiber-hücumların təsirinə məruz qoymuşdur. Burada informasiya təhlükəsizliyi hücumlarının reallaşması bulud infrastrukturunun serverlərinin yaddaş, CPU resurslarında anomal davranışın yaranmasına səbəb olur. Bu sistemlərin generasiya etdiyi böyük həcmdə verilənlərin təsnif edilməsi prosesi böyük xərc və vaxt tələb etdiyi üçün burada bu məsələnin həllində anomaliyaların yarım-öyrədilən (semi-supervised) metodların köməyilə aşkarlanması qənaətbəxş hesab olunur. Məqalədə bulud infrastrukturunun keyfiyyət göstəricilərində anomaliyaların aşkarlanması üçün yarım-öyrədilən metod təklif edilir. Burada anomal davranışı aşkarlamaq üçün keyfiyyət göstəriciləri üzrə Google və Yahoo! şirkətlərinin açıq verilənləri, Python 2.7, Matlab, Weka və Google Cloud SDK Shell proqramları istifadə edilmişdir. Modelin eksperimental tədqiqi nəticəsində 0.99% aşkarlama dəqiqliyi əldə edilmişdir.**

Açar sözlər*— *keyfiyyət göstəricilərində anomaliya; CPU sərfiyyatı; yaddaş sərfiyyatı; Isolation forest; Naive Bayes; J48 qərar ağacı, yarım-öyrədilən alqoritmlər; klassifikatorlar ansamblı

I. GİRİŞ

Son zamanlar bulud texnologiyalarının tətbiq sahəsinin çox yayılması bu infrastrukturun miqyasının və mürəkkəbliyinin böyük sürətlə artmasına səbəb olmuşdur. Bu iri miqyaslı sistemlərin keyfiyyət göstəricilərində deqradasiya və fasilələrin (downtimes) yaranması olduqca böyük problem hesab olunur [1]. Bu problemlərin aradan qaldırılmasında başlıca məsələ bulud infrastrukturunun aparat təminatında, sistemlərinin vəziyyətində və proqram təminatlarının yerinə yetirilməsində baş verə bilən anomaliyaların aşkarlanmasıdır.

Bulud infrastrukturunda anomal davranışlar sistemin resurslarının yüklənməsi, imtinalar, konfiqurasiya səhvləri, DDoS hücumlar səbəbindən baş verə bilir. Bu anomallıq gözlənilməz vəziyyətin yaranmasına səbəb olur və verilənlər mərkəzinin effektivliyinin aşağı düşməsi və işinin dayanması ilə nəticələnir [2].

Bulud infrastrukturunda anomal davranışı aşkarlamaq üçün resurs sərfiyyatının vəziyyətini qiymətləndirmək lazımdır. Bu qiymətləndirməni zaman sıraları ilə təsvir edilmiş server kriteriyalarını (ləngimə, CPU, yaddaş) izləməklə təmin edirlər [3].

Bulud infrastrukturunda anomaliyaların maşın təlimi üsulları ilə aşkarlanması üçün olduqca çox sayda tədqiqatlar aparılmışdır [1,4]. [5]-də öyrədilməyən aşkarlama metodlarının effektivliyini artırmaq məqsədi ilə klasterizasiya ansamblı yanaşmasından istifadə edirlər. [6]-da müdaxilələrin aşkarlanması məsələsində klassifikatorlar ansamblı istifadə edilir.

[7]-də bizim yanamaya oxşar yanaşma təklif edilmişdir. Belə ki, təsnif olunmamış verilənlərə ballandırma üsulu tətbiq olunaraq verilənlərin iki sinfə bölünməsi həyata keçirilmişdir. Burada yüksək anomaliya balına malik olan verilənlər anomal, qalanları isə normal davranış kimi təsnif edilmişdir. Daha sonra təsadüfi olaraq normal verilənlər çoxluğu seçilmişdir və anomal verilənlərlə birləşdirilərək təsnif edilmiş verilənlər bazası formalaşdırılmışdır. Növbəti mərhələdə öyrədilən alqoritm bu təsnif edilmiş verilənlərlə öyrədilmiş və nəticədə anomal davranış normal davranışdan fərqləndirilmişdir. Təqdim olunan məqalədə təklif edilən metodun mövcud metodlardan fərqi odur ki, burada təbiəti əvvəlcədən məlum olmayan verilənlərdən təsnif edilmiş verilənlər formalaşdırılır və bir neçə öyrədilən klassifikatorlar istifadə olunaraq robast aşkarlama sistemi qurulur. Burada klassifikatorlar çoxluğundan istifadə edilməsinin məqsədi, hər bir klassifikatorun mövcud anomaliyanı fərqli aşkarlamasıdır. Bundan başqa mövcud üsullarda aşkarlama prosesində kənaraçıxmaların (outlier) meydana çıxması halı nəzərə alınmır. Bu vəziyyət isə öz növbəsində anomaliyaların aşkarlanması dəqiqliyini azaldır. Burada verilənlər obyektinin qiyməti əlamət üçün təyin edilmiş sərhəd qiymətlərini aşarsa və verilənlər obyektinin sinfi normal kimi təsnif olunarsa, həmin element kənar element hesab olunur. Və əksinə əgər verilənlər obyektinin qiyməti əlamət üçün təyin edilmiş sərhəd qiymətlərindən aşağıdırsa və verilənlər obyektinin sinfi anomal kimi təsnif olunarsa, həmin element kənar element hesab olunur.

Təqdim olunan məqalədə bulud infrastrukturunda anomal davranışları aşkarlamaq üçün yanaşma təklif edilir. Təklif edilmiş metod vasitəsi ilə loq fayllar analiz edilir, qəbul edilmiş sərhəd qiyməti vasitəsi ilə resurs sərfiyyatı proqnozlaşdırılaraq təsnif olunmuş anomaliyalar formalaşdırılır. Metodda sərhəd qiymətinin istifadə edilməsi anomaliyaları yüksək dəqiqliklə aşkarlamağa imkan verir.

Təklif edilmiş yanaşma dörd mərhələdən ibarətdir. 1) Verilənlər bazasının normal və anomal siniflərə bölünməsinin təşkili. 2) Çoxsaylı klassifikatorların təsnif edilmiş verilənlər əsasında hər birinin fərdi qərarlarının formalaşdırılması. 3) Kollaborativ qərarın qəbul olunması üçün klassifikatorların irəli sürdüyü fərdi qərarları birləşdirən yekun anomaliya balının hesablanması. 4) Qərarın qəbulu.

DOI: 10.25045/NCSoftEng.2017.05

Anomaliyaların aşkarlanması metodunun əsasını üç klassifikator, Isolation forest, Naive Bayes və J48 qərar ağacı təşkil edir.

Təklif edilən metodun effektivliyinin qiymətləndirilməsi "Google cluster trace" və Yahoo!S5 açıq verilənlər bazaları üzərində aparılmışdır. Metodun aşkarlama dəqiqliyi *precision, recall, FP, f-measure, TP, TN, FN, FD, acuracy* metrikaları əsasında hesablanmışdır. Təklif edilən metodun mövcud verilənlər bazaları üzərində test edilən zaman metodun anomaliyaları aşkarlama dəqiqliyi 0.99% faiz, kənaraçıxma isə ümumi verilənlərin 0.70 faizini təşkil etmişdir.

II. KEYFİYYƏT GÖSTƏRİCİLƏRİNİN ANALİZİ KRİTERİYALARI

Mərkəzi prosessor (CPU) bütün proqram təminatını idarə edir və adətən sistemin keyfiyyət göstəricilərinin analizinin həyata keçirilməsində əsas kriteriya kimi istifadə edilir [8]. Müasir sitemlərdə CPU çox sayda olur və onlar mərkəzi paylayıcı (kernel scheduler) vasitəsi ilə yerinə yetirilən proqram təminatları arasında paylaşdırılır. CPU resurslarına normaldan artıq tələbat yarandıqda tapşırıqlar növbə yaradır və emal olunmaq üçün gözləmə rejimində olur. Gözləmə sistemin tapşırıqların yerinə yetirilmə vaxtında ləngimələr yaradır, bu isə keyfiyyət göstəricilərinin azalmasına səbəb olur.

Keyfiyyət göstəricilərini yaxşılaşdırmaq üçün CPU sərfiyyatını (CPU usage) analiz etmək lazım gəlir. Əksər hallarda CPU sərfiyyatı prosesə, axına və ya tapşırığa nəzərən analiz edilir.

Keyfiyyət göstəricisini yaxşılaşdırmaq üçün analiz edilən digər kriteriya məşğulluq nisbətidir (CPU utilization). Məşğulluq nisbəti CPU-nun işi yerinə yetirmək üçün məşğul olduğu zaman intervalındakı vaxtla ölçülür və faizlə göstərilir.

Yaddaşa giriş-çıxış cəhdləri (memory I/O (reads, writes,)) də yüksək CPU sərfiyyatına səbəb olur. Yaddaşa giriş-çıxış cəhdləri edildikdə CPU işini dayandırır və bu prosesin başa çatmasını gözləyir.

"Google cluster trace" verilənlər bazasında resurs sərfiyyatını analiz etmək üçün aşağıdakı parametrlər vardır: *cpu_usage; disk_io_time; disk_space; mem_usage; number_of_running_task; time*.

Təqdim olunan məqalədə resurs sərfiyyatının analizi *cpu_usage*, *mem_usage* və *time* parametrlərinin əsasında həyata keçirilmişdir.

III. KEYFİYYƏT GÖSTƏRİCİLƏRİNİN ANALİZİ ÜSULLARI

Bulud infrastrukturunun keyfiyyət göstəricilərinin diaqnostikasının aparılması üsulları iki böyük sinfə bölünür: keyfiyyət göstəriciləri anomaliyalarının aşkarlanması və əsas səbəblərin analizi (root cause analysis) [9].

Anomaliyaların aşkarlanması. Anomaliya ümumi qaydadan, düzümdən (arrangement) və ya formadan kənaraçıxma kimi təyin olunur [10] .

Bulud infrastrukturunun keyfiyyət göstəricilərində anomaliyaları aşkarlamaq üçün iki məşhur yanaşma istifadə edilir [9]:

*1) **Anomaliyaların keyfiyyət göstəriciləri tələblərinə görə aşkarlanması.*** Bu metoddan sistem ona qoyulmuş tələbləri ödəmədiyi hallarda istifadə edilir. Burada keyfiyyət göstəriciləri tələbləri açıq şəkildə servis səviyyəsi müqaviləsində (Service Level Agreement, SLA) göstərilir. Məsələn, məlumatların növbəsinin uzunluğu, tələb olunan sayda CPU.

*2) **Anomaliyaların normal keyfiyyət göstəricisindən kənaraçıxmalar əsasında aşkarlanması***. Bu yanaşma keyfiyyət göstəricisinə xüsusi tələblər olmadığı, lakin keyfiyyət göstəricisinin nəzərdə tutulmuş normadan kənara çıxdığı hallarda istifadə edilir. Məsələn, proqram təminatının versiyasının əvvəlki versiyadan əhəmiyyətli dərəcədə pis olmasının aşkarlanması, bir serverin keyfiyyət göstəriciləri kriteriyalarının digər identik serverdən kənara çıxmasının aşkarlanması. Bu tip məsələlərdə anomal keyfiyyət göstəriciləri verilənləri uyğun normal keyfiyyət göstəriciləri verilənləri ilə müqayisə edilir və kənaraçıxma sərhəd qiymətinə əsasən müəyyən olunur [11].

Təqdim olunan məqalədə anomaliyaların aşkarlanması 2-ci yanaşma əsasında həyata keçirilir.

Anomaliyaların baş vermə səbəblərinin analizi. Anomaliyaların baş vermə səbəblərinin analizi – bulud infrastrukturunda olan anomaliyanın baş vermə səbəblərinin müəyyən olunması prosesidir [2]. Keyfiyyət göstəriciləri kontekstində bu keyfiyyət göstəriciləri anomaliyalarının mənbəyinin identifikasiyası prosesidir. Proqramın yerinə yetirilməsi prosesində durğunluq (deadlocks), səhv konfiqurasiyalar, DDoS hücumlar və s. keyfiyyət göstəriciləri anomaliyalarının mənbələrinə misal göstərilə bilər.

IV. PROBLEMİN VƏZİYYƏTİ

Anomaliyaların maşın təlimi üsulları vasitəsi ilə aşkarlanması üçün olduqca çox sayda yanaşmalar təklif edilmişdir [12]. Böyük həcmdə təsnif olunmamış verilənlərin təsnifatlaşdırılması prosesi çox vaxt və xərclər tələb etdiyi üçün anomaliyaların yarım-öyrədilən aşkarlanması yanaşmalarından istifadə edirlər [13]. Yarım-öyrədilən metodların tətbiqi zamanı iki şərt nəzərə alınmalıdır: 1) verilənlər bazasında normal nümunələrin sayı anomal nümunələrin sayından kəskin artıq olmalıdır; 2) anomal nümunələr normal nümunələrdən statistik olaraq fərqlənməlidir. Bu səbəbdən aşkarlama modellərinin effektivliyi birbaşa normal və anomal davranışı fərqləndirmə üsulunun dəqiqliyindən asılı olur.

Məqalədə anomaliyaların yarım-öyrədilən aşkarlanması üsulu təklif edilir. Burada təbiəti əvvəlcədən məlum olmayan verilənlərdən təsnif edilmiş verilənlər formalaşdırılır və bir neçə öyrədilən klassifikatorlar istifadə olunaraq anomaliyaların robast aşkarlanması sistemi qurulur. Təklif edilmiş yanaşmanın əsas fəlsəfəsi aşağıdakı kimidir:

Addım 1. "Google cluster trace" verilənlər bazasının "task_usage" loq yazılarından resurs sərfiyyatı parametrlərinin əldə olunması;

DOI: 10.25045/NCSoftEng.2017.05

Addım 2. Hostdakı CPU-nun sərfiyyatına görə sərhəd qiymətinin təyin edilməsi (bizim halda CPU sərfiyyatı normalda 0.35-dur);

Addım 3. "Google cluster trace" verilənlər bazasından götürülmüş verilənlərin 0.35 sərhəd qiymətinə nəzərən iki sinfə ayrılması (1 - normal CPU sərfiyyatı, 0 – anomal CPU sərfiyyatı);

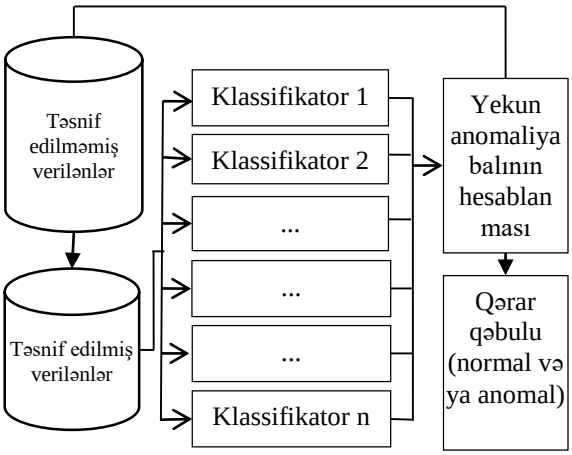
Addım 4. Bu verilənlər üzərində müxtəlif maşın təlimi üsullarının test edilməsi;

Addım 5. Əgər CPU sərfiyyatı təyin edilmiş sərhəd qiymətindən böyükdürsə CPU sərfiyyatı anomal sinfinə daxil edilsin;

Addım 6. Əgər CPU sərfiyyatı təyin edilmiş sərhəd qiymətindən kiçikdirsə CPU sərfiyyatı normal sinfinə daxil edilsin.

V. TƏKLİF EDİLƏN YANAŞMA

Məqalədə bulud texnologiyaları mühitində anomaliyaları aşkarlamağa imkan verən yanaşma təklif edilir (şəkil 1). Bu bölmədə yanaşmanın aşkarlamanı həyata keçirmək üçün nəzərdə tutulmuş bloklarının şərhi verilir.



Şəkil 1. Bulud infrastrukturunda anomaliyaların aşkarlanması modeli

1. Verilənlər bazalarının yaradılması

Bu mərhələdə təsnif edilməmiş verilənlər əsasında təsnif edilmiş verilənlər bazası yaradılır. Bu prosesi həyata keçirmək üçün ümumi verilənlər bazasının verilənləri qəbul edilmiş sərhəd qiyməti (0.35 prosessor sərfiyyatı üçün, 0.39 yaddaş sərfiyyatı üçün) əsasında normal və anomal kimi iki sinfə bölünür.

2. Qərarların qəbulu modulu

Qərarların qəbulu modulu üç öyrədilən klassifikatordan (Naive Bayes, Isolation forest, J48) ibarətdir. Bu klassifikatorların hər birinin irəli sürdüyü fərdi qərarları birləşdirilərək kollaborativ qərar qəbul olunur. Burada hər bir klassifikator qərar modelini qurmaq üçün təsnif edilmiş verilənlər çoxluğu ilə öyrədilir.

Fərz edək ki, $C = \{c_i \mid 1 \le i \le n\}$ seçilmiş klassifikatorlardır və $D = \{d_i \mid 1 \le i \le n\}$ sayda qərar qəbulu modeli qurur. Hər bir i -ci qərarqəbuletmə modeli d_i test edilən x_j elementinə binar qiymət (0 və ya 1) verir, $d_i(x_j) = v_{ij}, \quad j = 1,...,m.$ Burada v_{ij} -un binar qiyməti 1 olduqda x_j elementi anomaliya kimi qiymətləndirilir, əks halda normal kimi qiymətləndirilir. Belə olduqda anomaliya balı aşağıdakı düsturla hesablanır:

$$S_j = \frac{\sum_{i=1}^{n} v_{ij}}{n} \tag{1}$$

burada n anomaliya balının hesablanmasında iştirak edən qərar qəbuletmə modellərinin sayıdır. Verilənlər bazasının elementlərinin sinfi, anomal və ya normal, aşağıdakı düsturla hesablanır:

$$Class(x_j) = \begin{cases} S_j \ge \rho & Anomal = 1 \\ S_j < \rho & Normal = 0 \end{cases} \tag{2}$$

Burada ρ – qərar qəbuletmə modellərinin test edilən x_j elementini anomaliya kimi qiymətləndirmək üçün verdikləri balın faiz qiymətidir. Məsələn, əgər ρ sərhəd qiyməti 1 götürüldükdə, və əgər iştirak edən qərar qəbuletmə modellərinin hamısı eyniliklə elementin anomaliya olduğunu qəbul etməsə, test edilən element anomaliya kimi qəbul olunmayacaqdır.

Misal. Fərz edək ki, öyrədilməyən 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. Fərz edək ki, test edilən x_j elementinə qərar qəbuletmə modelləri tərəfindən uyğun olaraq 1, 1, 0, 1, 1 balları verilmişdir. Onda x_j elementinin anomaliya balı aşağıdakı kimi hesablanar:

$$x_j = \frac{1+1+0+1+1}{5} = \frac{4}{5} = 0.80 \tag{3}$$

Burada $\rho = 0.6$ qiymətində x_j elementi anomal kimi qiymətləndirilir.

VI. İSTİFADƏ EDİLƏN ALQORİTMLƏR

Maşın təlimi nəzəriyyəsinin əsasını çoxsaylı klassifikasiya alqoritmləri təşkil edir. Təqdim olunan məqalədə aşağıdakı alqoritmlər istifadə edilmişdir:

1. Sadəlövh Bayes alqoritmi (Naive Bayes);

Naive Bayes üsulunda dəyişənlərin bir-birindən asılı olmaması əsas şərt kimi qəbul olunur. Obyektin bu və ya digər sinfə aid olması ehtimalını Bayes düsturu vasitəsi ilə tapır. Metodun adında sadəlövh (naive) sözü dəyişənlərin bir-birindən asılı olmadığına görə istifadə edilmişdir.

DOI: 10.25045/NCSoftEng.2017.05

Fərz edək ki, $P(y = c_r)$ hər hansı bir y obyektinin $c_r(y = c_r)$ sinfinə aid olması ehtimalıdır. Və fərz edək ki, $E-$ asılı olmayan dəyişənlərin müəyyən qiymət almasını göstərən hadisə, $P(E)$ isə həmin hadisənin baş vermə ehtimalıdır.

Metodun ideyası obyektin c_r sinifinə aid olmasını müəyyən edən şərti ehtimalı hesablamaqdır.

Ehtimal nəzəriyyəsindən məlumdur ki, hər hansı E hadisəsinin baş vermə ehtimalı aşağıdakı düsturla hesablanır:

$$P(y = c_r \mid E) = \frac{P(E \mid y = c_r) * P(y = c_r)}{P(E)} \qquad (4)$$

Naive Bayes metodunda qaydalar generasiya olunur, bu qaydaların şərt hisssində bütün asılı olmayan dəyişənlər uyğun mümkün qiymətlərlə müqayisə edilir. Yekun hissədə asılı dəyişənin bütün mümkün qiymətləri göstərilir. Yəni $x_1 = c_1^k, \ldots, x_n = c_n^k$, $y = c_r$ və s. Beləliklə bu qaydalar bütün elementlər üçün qurulur. Burada y – öyrədilən seçim daxilindəki bütün obyektlər, c_r – hər hansı r -ci sinif, $x_1, \ldots, x_n$ - asılı olmayan dəyişənlər, c_i^k - dəyişənlərin aldığı bütün mümkün qiymətlərdir.

Bütün bu qaydaların hər biri üçün Bayes düsturuna əsasən onların ehtimalı aşağıdakı düsturla hesablanır:

$$P(E'' y = c_r) = P(x_1 = c_1^k \mid y = c_r) * \ldots * P(x_n = c_n^k \mid y = c_r) \qquad (5)$$

2. İzolyasiya olunmuş meşə alqoritmi (Isolation forest);

İsolation Forest alqoritmi verilmiş verilənlər bazası üçün izolyasiya ağacları ansamblı qurur. Ağacda yolun ortalama uzunluğu qısa olan nümunələr anomaliya kimi qiymətləndirilir. Burada nümunələrin anomaliya balı aşağıdakı düsturla hesablanır [14]:

$$c(n) = 2H(n-1) - (2(n-1)/n) \qquad (5)$$

$$S(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \qquad (6)$$

burada $h(x)$ x obyektinin izolyasiya ağacını (ITree) dövr edən zaman keçdiyi tillərin sayıdır, $E(h(x))$ – h(x) tillərinin ortalama sayıdır. $H(i)$ harmonik ədəddir və Eyler sabiti ilə $\ln(i) + 0.5772156649$ təyin edilir.

3. Qərar Ağacı (J48).

J48 qərar ağacı alqoritmi C4.5 alqoritminin Java proqramında reallaşdırılmış versiyasıdır. C4.5 isə ID3 alqoritminin modifikasiya olunmuş versiyasıdır. J48 alqoritmi bütün təlim nümunələri üzərində əməliyyatları ağacın kökündən başlayaraq həyata keçirir. Ağacın kökündəki nümunələr çoxluğunu ayırmaq (bölmək) üçün atributlardan biri seçilir və bu atributun aldığı hər bir qiymət üçün budaq qurulur və bununla da altqovşaq yaradılır. Daha sonra atributun qiymətinə uyğun olaraq bütün nümunələr bu altqovşaqlara paylanır. Alqoritm bölünməni bir növ sinfə aid nümunələr alınana qədər rekursiv olaraq davam etdirir. Bu şərt ödəndikdə qovşaqlar yarpaq kimi elan edilir və bölünmə dayandırılır. Bu alqoritmin ən mürəkkəb mərhələlərindən biri bölünməni həyata keçirmək üçün istifadə olunacaq atributun seçilməsi hesab olunur.

Bölünmə atributunu seçmək üçün informasiya artımı (information gain) və entropiya azalması (entropy reduction) kriteriyalarından istifadə edirlər.

$$Gain(S) = Entropy(T) - Entropy_s(T) \qquad (7)$$

Burada $Entropy(T)$ – T çoxluğunun bölünməyə qədər olan entropiyasıdır; $Entropy_s(T)$ – S bölünməsindən sonrakı entropiyadır.

İnformasiya artımı ($Gain(S)$) yüksək olan atribut S bölünməsi prosesində istifadə olunmaq üçün ən yaxşı atribut hesab edilir.

VII. EKSPERİMENT

Eksperimentlərin aparılması üçün "Google cluster trace" və "Yahoo! S5" real verilənlər bazaları istifadə edilmişdir [15, 16]. Məqalədə təklif edilmiş metodun tətbiqi Google verilənlər mərkəzinin topladığı açıq keyfiyyət göstəriciləri verilənləri üzərində aparılmışdır. Bundan əlavə Yahoo! verilənlər bazası və Google verilənlərindən istifadə etməklə təklif edilmiş metodun aşkarlama dəqiqliyi yoxlanmışdır. Burada "Google cluster trace" verilənlər bazasının "machine usage" loq yazılarının əlamətlər vektorunu 'cpu_usage', 'disk_io_time', 'disk_space', 'mem_usage' kimi xarakteristikalar təşkil edir.

"Google cluster trace" loq yazıları csv (comma separated values) formatında olan çox sayda fayldan ibarətdir. Bu faylların kontenti xronoloji ardıcıllıqda düzülmüşdür və hadisələrin qeydə alınması o-cı zamandan (timestamp 0) başlayaraq həyata keçirilir. Burada növbəti zaman anında baş verən hadisə sonrakı csv faylında qeydə alınmışdır. Bu csv fayllarını birləşmiş şəkildə istifadə etmək üçün "python data analysis library"istifadə edilmişdir.

Anomaliyaları aşkarlamaq üçün bu csv faylları çox sayda əlamət bölmələrindən ibarətdir: 'time', 'missing info', 'job ID', 'task index', 'machine_id', 'event type', 'user', 'scheduling class', 'priority', 'cpu request', 'memory request', 'disk space request', 'different machines restriction'. Təqdim olunan məqalədə anomaliyaların aşkarlanması modeli "cpu_usage", "mem_usage" və "time" əlamətlər vektoru əsasında test edilmişdir.

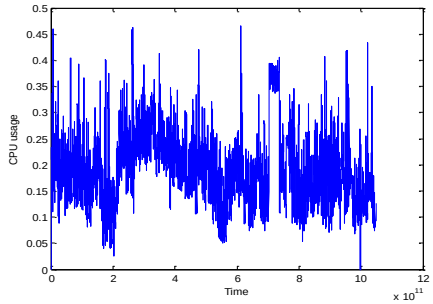
A. Anomaliyanın aşkarlanması

Anomaliyanı aşkarlamaq üçün sistemin davranışı bulud verilənlər mərkəzinin iş yükünə görə analiz olunur (modelləşdirilir). Bu prosesi həyata keçirmək üçün bulud verilənlər mərkəzindən iş yükünün prosessor sərfiyyatı (CPU usage), yaddaş sərfiyyatı (memory usage) və zaman (time) xarakteristikaları götürülmüşdür. Burada anomaliyaların aşkarlanması sərhəd metoduna əsaslanır.

Şəkil 2-də "Google cluster trace" verilənlər bazasından götürülmüş təsnif olunmamış verilənlər əsasında zamana görə

DOI: 10.25045/NCSoftEng.2017.05

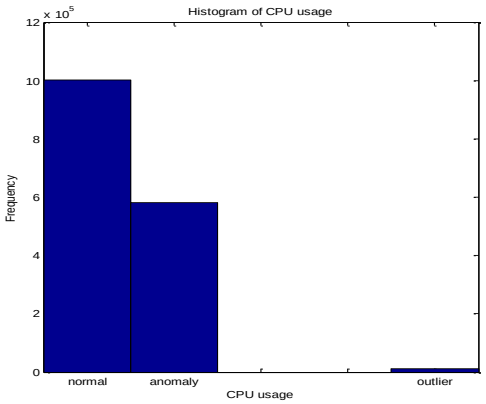
CPU sərfiyyatının zaman sıralarını göstərən diaqram təsvir edilmişdir. Təklif edilmiş yanaşmaya əsasən anomalıiyanı aşkarlamaq üçün ilk öncə bu təsnif edilməmiş verilənlərin təyin edilmiş sərhəd qiymətinə əsasən təsnifatlaşdırılması aparılır. Burada sərhəd qiyməti 0.35 götürülərək verilənlər bazasının verilənləri normal və anomal kimi iki sinfə bölünür. CPU sərfiyyatının 0.35 qiymətindən kiçik qiymətləri normal, böyük qiymətləri isə anomal davranış kimi qiymətləndirilir.



Şəkil 2. "Google cluster trace" bazasında CPU sərfiyyatının zaman sıraları

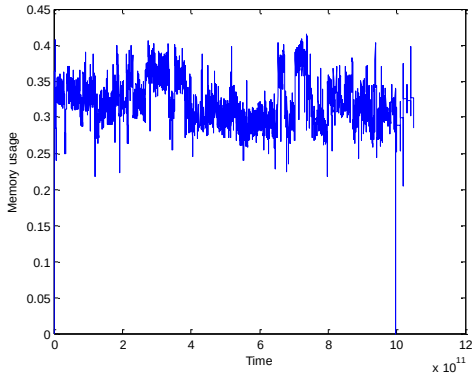
Anomaliyanı aşkarlamaq üçün qəbul edilmiş sərhəd qiyməti provayder tərəfindən təyin olunur.

Şəkil 3-də təsvir edilmiş diaqramda kiçik qiymətlər metodun aşkarlanma prosesində yol verdiyi kənaraçıxmalardır (outlier). Bu kənaraçıxmalar verilənlər bazasının bəzi vektorlarının (elementlərinin, təsvirlərinin) yalnış identifikasiya olunduğunu göstərir. Burada təklif edilmiş metodun tətbiqi ilə ümumi (1594877) verilənlərin 1001009 nümunəsi normal, 582601 anomal, 11267 isə kənaraçıxma kimi identifikasiya edilmişdir. Ümumiyyətlə anomaliyaların aşkarlanması metodlarında kənaraçıxmaların sayının minimum olmasına çalışırlar. Bir çox tədqiqat işlərində bu qiymət ümumi verilənlərin 1%-dən çox hissəsini təşkil edir [1]. Məqalədə təklif edilmiş metodun üstünlüklərindən biri odur ki, burada aşkarlama prosesində olduqca az kənaraçıxmanın olmasına yol verilmişdir və bu qiymət ümumi verilənlərin 0.71 faizini təşkil edir.



Şəkil 3. Aşkarlanmanın CPU sərfiyyatı üzrə nəticələrinin histoqramı

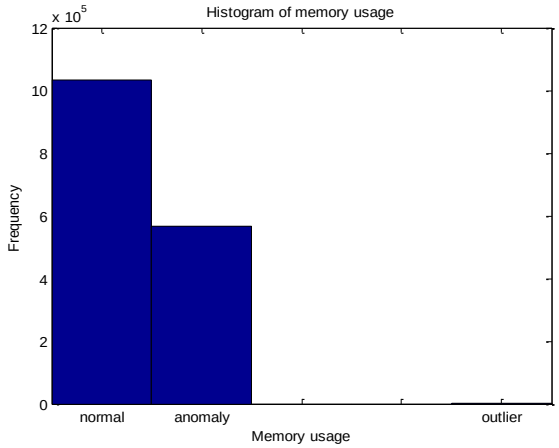
Şəkil 4-də "Google cluster trace" verilənlər bazasından götürülmüş təsnif edilməmiş verilənlər əsasında zamana görə yaddaş sərfiyyatının zaman sıralarını göstərən diaqram təsvir edilmişdir.



Şəkil 4. "Google cluster trace" bazasında yaddaş sərfiyyatının zaman sıraları

Yaddaş resurslarında anomal davranış sistemdə yaddaş sızdırma (memory leakage) yaradan informasiya təhlükəsizliyi hücumlarının reallaşması hesabına baş verə bilər. CPU sərfiyyatında olduğu kimi anomaliyaların yaddaş sərfiyyatı üzərində aşkarlanması prosesi də sərhəd yanaşmasına əsaslanır. Burada yaddaş sərfiyyatı üçün maksimum diapazon sərhəd qiyməti vasitəsi ilə təyin edilir. Sərhəd qiyməti 0.39 götürülərək vrilənlər bazasının verilənləri normal və anomal kimi iki sinfə bölünür (şəkil 4). Yaddaş sərfiyyatının 0.39 qiymətindən kiçik qiymətləri normal, böyük qiymətləri isə anomal davranış kimi qiymətləndirilir. Burada yaddaş sərfiyyatı üçün yol verilən ranq 0-dan sərhəd qiymətinə kimi interval sayılır. Sistemin boş dayandığı və ya yuxu rejimində olduğu hallarda yaddaş sərfiyyatı 0 ola bilər. Bu vəziyyət anomal hesab edilə bilməz.

Yaddaş sərfiyyatının test edilməsində də metodun aşkarlanma prosesində yol verdiyi kənaraçıxmalar azlıq təşkil edir (şəkil 5).



Şəkil 5. Aşkarlanmanın yaddaş sərfiyyatı üzrə nəticələrinin histoqramı

B. Dəqiqliyin yoxlanılması

Təklif edilmiş modelin anomaliyanı aşkarlama dəqiqliyini qiymətləndirmək üçün aşağıdakı kriteriyalar istifadə edilmişdir:

DOI: 10.25045/NCSoftEng.2017.05

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

$$\text{Precision} = TP/(TP + FP) \quad (10)$$

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

$$\text{Recall} = TP/(TP + FN) \quad (11)$$

3) Yalnış müsbət hallar (false positive rate, FPR).

$$FPR = FP/(FP + TN) \quad (12)$$

4) F-ölçü (F-measure).

$$\text{F-measure} = \frac{2*(\text{precision} * \text{recall})}{\text{precision} + \text{recall}} \quad (13)$$

5) Doğru müsbət hallar (True Positive, TP).

$$TPR = TP/\text{positive} \quad (14)$$

6) Doğru mənfi hallar (True Negative, TN).

$$TNR = TN/\text{negative} \quad (15)$$

7) Yalnış mənfi hallar (False Negative, FN).

$$FNR = FN/(FN + TP) \quad (16)$$

8) Yalnış aşkarlanma əmsalı (False discovery rate, FDR).

$$FDR = FP / (FP + TP) \quad (17)$$

Doğruluq (acuracy).

$$\text{Acuracy} = (TP + TN) / (P + N) \quad (18)$$

Precision göstəricisinin yüksək olması FPR səhvlərinin az olduğunu, recall göstəricisinin yüksək olması FNR səhvlərinin az olduğunu göstərir. Precision və recall göstəriciləri yüksək olan modellər ideal aşkarlama modelləri hesab olunur, məsələn, precision=recall=1. Lakin Precision və recall göstəricilərinin hər ikisində eyni zamanda yüksək qiymət əldə etmək olduqca mürəkkəbdir. Onlar adətən bir-birinə tərs mütənasib olur. Əksər hallarda recall göstəricisini artırdıqda precision aşağı qiymət alır və əksinə. F-measure modelin dəqiq olub-olmadığını göstərir. Bu göstərici precision və recall göstəricilərinin kifayət qədər yüksək olduğunu göstərir.

Anomaliyaların aşkarlanması modeli təsnif edilməsi Yahoo! S5 və təsnif edilməmiş "Google cluster trace"verilənlər bazaları üzərində test edilmişdir. Testetmə zamanı modelin aşkarlama dəqiqliyi (accuracy) hər iki baza üzrə 0.99% təşkil etmişdir (cədvəl 1).

CƏDVƏL 1. TƏKLİF EDİLMİŞ METODUN AŞKARLAMA DƏQİQLİYİ (ACURACY)

	Yahoo! S5	**Google cluster trace**
Naive Bayes	0.56 %	0.97%
Isolation Forest	0.51%	0.48%
Klassifikatorlar ansamblı	0.99%	0.99%

Bundan başqa təklif edilmiş yanaşmanın Naive Bayes və Isolation Forest kimi digər mövcud metodlarla müqayisəli analizi aparılmışdır (cədvəl 2,3).

CƏDVƏL 2. YAHOO! S5 VERİLƏNLƏR BAZASI

Yahoo! S5	**Naive Bayes**	**Isolation Forest**	**J48**	**Klassifikatorlar ansamblı**
TP	0.547	0.777	0.966	0.996
FP	0.447	0.761	0.007	0.004
TN	0.579	0.239	0.960	1.000
FN	0.425	0.486	0.080	0.000
FDR	0.453	0.223	0.014	0.004
F-measure	0.561	0.619	0.974	0.998
Recall	0.547	0.777	0.973	0.996
Acuracy	0.563	0.513	0.973	0.997
Precision	0.547	0.777	0.973	0.997

CƏDVƏL 3. "GOOGLE CLUSTER TRACE" VERİLƏNLƏR BAZASI

"Google cluster trace"	**Naive Bayes**	**Isolation forest**	**J48**	**Klassifikatorlar ansamblı**
TP	0.961	0.742	0.920	0.990
FP	0.022	0.313	0.013	0.006
TN	0.978	0.330	0.923	0.995
FN	0.036	0.606	0.011	0.008
FDR	0.039	0.258	0.015	0.010
F-measure	0.962	0.514	0.921	0.990
Recall	0.961	0.742	0.924	0.990
Acuracy	0.971	0.482	0.925	0.992
Precision	0.960	0.741	0.915	0.989

Cədvəldən göründüyü kimi təklif edilən yanaşmanın bütün kriteriyalar üzrə aşkarlama dəqiqliyi digər metodlarla müqayisədə daha yüksəkdir.

VIII. NƏTİCƏ

İri miqyasa və mürəkkəb konstruksiyaya malik bulud texnologiyaları müxtəlif kiber-hücumların təsirinə asanlıqla məruz qalır. Bu vəziyyət bulud texnologiyalarının etibarlılığının azalmasında və keyfiyyət göstəricilərinin deqradasiyasının yaranmasında əhəmiyyətli dərəcədə özünü göstərir və keyfiyyət göstəriciləri anomaliyalarının yaranmasına səbəb olur. Məqalədə bulud texnologiyalarının müxtəlif kiber-hücumlara qarşı dözümlülüyunu artırmaq məqsədi ilə bu infrastrukturda kiber-hücumların təsirindən baş verən keyfiyyət göstəriciləri anomaliyalarının real zamanda aşkarlanması üsulu təklif edilir.

Bu prosesi həyata keçirmək üçün əvvəlcə təsnif edilməmiş verilənlər iki sinfdə, anomal və normal, qruplaşdırılmaq üçün təsnifatlaşdırılır. Sonra bir neçə öyrədilən (supervised)

DOI: 10.25045/NCSoftEng.2017.05

klassifikatorlar həmin təsnif edilmiş verilənlərlə öyrədilir və anomal davranışı aşkarlamaq üçün bu klassifikatorların verdiyi ballar aqreqasiya olunaraq kollaborativ qərar qəbul olunur.

Təklif edilmiş üsulun eksperimental tədqiqi zamanı anomaliyaların klassifikatorlar ansamblı əsasında aşkarlanmasının nəticəsi ayrı-ayrı klassifikatorların verdiyi nəticələrdən üstün olmuşdur.

TƏŞƏKKÜRNAMƏ

Bu iş Azərbaycan Respublikasının Prezidenti yanında Elmin İnkişafı Fondunun maliyyə yardımı ilə yerinə yetirilmişdir– **Qrant № EİF-KETPL-2-2015-1(25)-56/05/1**

ƏDƏBİYYAT

[1] B. Agrawal, T. Wiktorski, R. Chunming, “Adaptive Anomaly Detection in Cloud Using Robust and Scalable Principal Component Analysis,” In Proc. of IEEE 15th International Symposium on Parallel and Distributed Computing., 2016, pp. 1-8.

[2] S. Hangal, M. S. Lam, “Tracking Down Software Bugs using Automatic Anomaly Detection,” Proc. of the 24th ACM International Conference on Software Engineering, 2002, pp. 291-301.

[3] Q. Guan, Z. Zhang, S. Fu, “Ensemble of Bayesian Predictors and Decision Trees for Proactive Failure Management in Cloud Computing Systems,” Journal of communications, 2012, vol. 7, no. 1, pp. 1-10.

[4] O. Ibidunmoye, T. Metsch, E. Elmroth, “Real-time Detection of Performance Anomalies for Cloud Services,” IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), 2016, pp. 1-2.

[5] F. Weng, Q. Jiang, L. Shi, and N. Wu, “An intrusion detection system based on the clustering ensemble,” Proc. of the IEEE International Workshop on Anticounterfeiting, Security, Identification, 2007, pp. 121-124.

[6] J. Kittler, M. Hatef, R. P. Duin, J. Matas, “On combining classifiers,” Proc. of the IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, vol. 20, no. 3, pp. 226-239.

[7] K. Yamanishi, J. Takeuchi, “Discovering outlier filtering rules from unlabeled data,” Proc. of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 2001, pp. 389-394.

[8] B. Gregg Systems Performance: Enterprise and the Cloud, 1st edition, 2013, 792 p.

[9] M. Peiris, J.H. Hill, J. Thelin, S. Bykov, G. Kliot, C. Konig, PAD: Performance Anomaly Detection in Multi-server Distributed Systems, Proc. of the IEEE International Conference on Cloud Computing, 2014, pp. 769-776.

[10] P. Barford, N. Duffield, A. Ron, and J. Sommers, “Network Performance Anomaly Detection and Localization,” IEEE INFOCOM, 2009, pp. 1377-1385.

[11] H. Malik, H. Hemmati, and A. E. Hassan, “Automatic Detection of Performance Deviations in the Load Testing of Large Scale Systems,”Proc. of the IEEE International Conference on Software Engineering, 2013, pp. 1012-1021.

[12] V. Chandola, A. Banerjee, V. Kumar, “Anomaly detection:A survey,” Journal of ACM Computing Surveys, 2009, vol. 41, no. 3, 72 p.

[13] L. Portnoy, E. Eskin, S. Stolfo, “Intrusion detection with unlabeled data using clustering,” Proc. of ACM CSS Workshop on Data Mining Applied to Security (DMSA), 2001, pp. 5-8.

[14] W. Chen, Y. Yun, M. Wen, H. Lu, Z. Zhang, Y. Liang, “Representative subset selection and outlier detection via isolation forest,” Analitical Models, 2016, Vol. 8, No.39, pp.7225- 7231.

[15] Google Cluster Data, “ClusterData2011_2 traces,” https://github.com/google/cluster-data

[16] Yahoo! Webscope, "S5 - A Labeled Anomaly Detection Dataset, version 1.0 (16M),"
https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

DOI: 10.25045/NCSoftEng.2017.05