ПРИЛОЖЕНИЕ НА ТЕХНИКИ ОТ КЛЪСТЕРНИЯ АНАЛИЗ В СИСТЕМИТЕ ЗА ОТКРИВАНЕ НА НАРУШЕНИЯ

Веселина Жечева, Евгения Николова

Резюме: Настоящата статия представя приложения на техники от клъстърния анализ за реализиране на система за откриване на нарушения, извършваща поведенчески анализ. Чрез прилагане на K-значно клъстериране се извършва разделяне и класифициране на данните за извършваните действия в наблюдаваната система, като се откриват действията, резултат от нарушения на политиката на сигурност.

Ключови думи: информационна сигурност, системи за откриване на нарушения, клъстерен анализ

1. Увод

Значението на информационната сигурност нарасна през последните десетилетия заедно с нарастването на броя на компютрите и мобилните устройства, които работят в мрежова и Интернет среда. Нарушенията на сигурността на системите представляват опити за заобикаляне или нарушаване правилата, определени от политиката на сигурност. Тези действия представляват нарушения на различни аспекти на информационната сигурност:

- Конфиденциалност (поверителност) – ресурсът трябва да бъде разкрит само на тези, които имат съответните права за достъп;
- Интегритет (цялостност) – ресурсът трябва да бъде изменян само от тези, които имат съответните права за това;
- Достъпност – ресурсът трябва да бъде достъпен и използваем за легитимните потребители когато те поискат достъп до него.

Откриване на нарушения на сигурността в информационните и мрежови системи се нарича процесът на наблюдение на процесите в тези системи и определяне дали извършените действия са злонамерени или неоторизирани. За тази цел системите за откриване на нарушения сканират действията на потребителите и ги сравняват с образци на известни средства за злонамерени действия (сигнатурен анализ) или търсят отклонения от предварително генерирани профили на нормалните потребителски действия в системата (поведенчески анализ). Основното предимство на метода, основан на поведенчески анализ е потенциалната възможност за откриване на нови, неизвестни до момента атаки. Тези системи се състоят от сензори, които събират определени данни от един или повече източника, извършват първична обработка и селекция на данните по различни признаци, прилагат определен алгоритъм, след което при необходимост изпращат сигнал за открито нарушение до мениджър, отговорен за следене, конфигуриране и анализ на данните. Получените сигнали за нарушения могат да доведат до съответни законови действия, преконфигуриране на защитна стена, поправяне на открити уязвимости, както и да помогнат при цялостен анализ на изследваната система (root cause analysis).

Системите за откриване на нарушения обработват голямо количество данни за процесите и събитията в наблюдаваната система. Един от ефективните методи за редуциране обема и изследване характеристиките на изследваните данни е клъстерният анализ [Julisch, Lieto, Ejaz, Al-Mamory]. Неговата цел е класификация и групиране на данните за разглежданата система на непресичащи се подмножества (наречени клъстери) на базата на различни признаци. Обект на разглеждане на настоящата статия е изследването на възможността за прилагане на алгоритъм от клъстерния анализ за

реализиране на поведенчески анализ в системи за откриване на нарушения. Целта е сканиране на данните за работата на наблюдаваната система и реализирането на двоична класификация, т.е. разделянето им на два непресичащи се клъстъра, съдържащи съответно данни за нормални действия и за действия, които са резултат от неоторизиран достъп.

2. Описание на методологията

2.1. К-значно клъстериране (K-means clustering)

К-значното клъстериране (*K-means clustering)* [MacQueen J.] е алгоритъм от клъстърния анализ, който групира обекти в K непресичащи се клъстъра въз основа на функция на разстоянията. K е цяло положително число, което показва броя на клъстърите и се задава предварително. Четирите стъпки на алгоритъма са следните:

1) Определя се броят на клъстърите K.

За да използваме техниката на клъстърния анализ в системите за откриване на нарушения, първо трябва да определим на колко клъстъра ще разделим нашите наблюдения. В нашия случай, ние искаме да ги разделим в два класа, единият от нормални наблюдения, а другият – от аномалии.

2) Инициализират се K клъстърни центъра, като се разделят всички n обекта $X_n = \{x_1, x_2, ..., x_n\}$ в K клъстъра

$$K = \{K_1, K_2, ..., K_k\}, \; K_i \cap K_j = 0, \; \bigcup_{i=1}^{k} K_i = X_n,$$

изчисляват се клъстърните центрове $\{\xi_1, \xi_2, ... \xi_k\}$ и се проверява дали са различни. Алтернативен подход е да се изберат K произволни, различни обекта за центрове.

В системите за откриване на нарушения ние искаме да класифицираме всяко наблюдение в една от две групи. Първоначално не знаем как изглеждат клъстърите. Избираме K-значното клъстериране, за да ги намерим, както и да определим всяко наблюдение на кой от клъстърите принадлежи. Първо, избираме по случаен начин две наблюдения за центрове на всеки един от клъстърите.

3) Итеративно за всяко едно наблюдение се изчислява разстоянието до центровете. Най-малкото разстояние от наблюдението до всеки от клъстърите определя принадлежността му към съответния клъстър. Тази класификация зависи от използваните метрики, представени в точка 2.2.

4) Когато всички наблюдения са класифицирани в техните най-близки клъстъри, се преизчисляват центровете на клъстърите. Оценяваме j нов център чрез

$$\xi_j = \arg\min_{\xi} \sum_{i:\pi_i = j} d(x_i, \xi),$$

където $\pi_i \arg\min_j d(x_i, \xi_j)$, $d(.,.)$ - мярка за разстоянието между два вектора.

5) Повтаря се стъпка 3 докато се намерят точните центрове на всеки клъстър.

Основното предимство на K-значното клъстериране е неговата простота и скорост, както и факта, че е добра възможност, ако системите за отчитане на нарушения се използват в реално време. Също така и фактът, че неговата сложност се увеличава линейно при увеличаване на броя наблюдения.

2.2. Метрики за определяне разстоянието между два вектора

За определяне разстоянието между два вектора могат да бъдат приложени множество метрики.

- *Разстояние на Вагнер – Фишер* [Wagner]. То определя минималния брой операции (вмъкване, изтриване, заместване), които трябва да се извършат с единия вектор, за да стане той идентичен с втория. Това разстояние се задава от формулата:

$$d_{WF}(i,j) = \min\begin{Bmatrix} d(i-1,j)+w(x_i,\varepsilon), d(i,j-1)+w(\varepsilon,y_j), \\ d(i-1,j-1)+w(x_i,y_j) \end{Bmatrix}. \qquad (1)$$

където $w(a,b)$ е цената на заместването на елемента *a* с елемента *b*; $w(a,\varepsilon)$ е цената на изтриването на елемента *a* и $w(\varepsilon,b)$ е цената на вмъкването на елемента *b*. Този алгоритъм изисква време от порядъка на O(*mn*) и памет $(m+1)\times(n+1)$, необходима за съхранението на двумерен масив от числа с плаваща точка, където *m* и *n* са дължините на двата вектора.

- *Разстояние на Жаро* [Jaro]. То се задава по следния начин:

$$d_J = \frac{1}{3}\left(\frac{m}{|s_1|}+\frac{m}{|s_2|}+\frac{m-t}{m}\right) \qquad (2)$$

където *m* е броят на общите елементи на двата вектора, *t* е броят на транспозициите, т.е. броят на общите елементи, разделен на 2.

- *Разстояние на Жаро-Уинклер* [Winkler]. Това разстояние е модификация на предходното и се задава по следния начин:

$$d_{JW} = \frac{1}{3}\left(\frac{m}{|s_1|}+\frac{m}{|s_2|}+\frac{m-t}{m}\right)+lp\left[1-\frac{1}{3}\left(\frac{m}{|s_1|}+\frac{m}{|s_2|}+\frac{m-t}{m}\right)\right], \qquad (3)$$

където *m* е броят на общите елементи на двата вектора, *t* е броят на транспозициите, l е дължината на общата последователност (префикс) в началото на двата вектора, а *p* е константен теглови множител, задаващ степента на близост на общите префикси на двата вектора.

- *Разстояние на Смит – Уотърман* [Smith]. Това разстояние може да се намери чрез алгоритъм от динамичното програмиране, който изчислява разстоянието по следния начин:

$$H(i,j) = \max\begin{pmatrix} 0 & \\ H(i-1,j-1)+w(a_i,b_j) & съвпадение / несъвпадение \\ H(i-1,j)+w(a_i,-) & изтриване \\ H(i,j-1)+w(-,b_j) & вмъкване \end{pmatrix}, 1\le i\le n, 1\le j\le m$$

където *a* и *b* са двата вектора, *n* и *m* са съответно дължините им. Матрицата *H* се инициализира по следния начин: $H(i,0)=0, 1\le i\le n$; $H(0,j)=0, 1\le j\le m$. Ако $a_i = b_j$, то стойността $w(a_i,b_j)$ е равна на предварително определена теглова стойност, а ако $a_i \neq b_j$, то стойността $w(a_i,b_j)$ е равна на определена теглова стойност за несъвпадение.

- Разстояние на Монге - Елкан [Monge, Monge]. Това разстояние използва и обобщава резултатите, получени от други разстояния:

$$sim(a,b) = \frac{1}{K}\sum_{i=1}^{K}\max_{j=1}^{L} sim'(a_i,b_j)$$

където *a* и *b* са двата вектора, *K* и *L* са съответно дължините им, а sim' е функция, изчисляваща друго разстояние, например това на Левенщайн, Жаро и др.

2.3. Структура на клъстърите

При анализиране на клъстерите може да се предположи, че клъстерът с повече на брой вектори е клъстер, съдържащ вектори от нормалните дейности, а другият клъстер

съдържа аномалии. Въпреки, че това звучи логично, може да се види, че широкомащабните атаки генерират повече вектори аномалии в сравнение с нормалните, които ще генерират фалшиви такива. За по-добро класифициране трябва да се анализират структурите на клъстерите. За целта се интересуваме от размера и разстоянието между клъстърите.

Вътреклъстърно разстояние (Intra-cluster distance) ще наричаме размера или компактността на всеки клъстър. То се пресмята по един от следните начини:

- Разстоянието като диаметър, който е най-голямото разстояние между два вектора в клъстъра Ki $\Delta(K_i) = \max_{x,y \in K_i} \{d(x,y)\}$;
- Разстоянието като осреднен диаметър, който е средна стойност на разстоянията между всички вектори, принадлежащи на един и същи клъстър Ki $\Delta(K_i) = \frac{1}{n+(n+1)} \sum_{\substack{x,y \in K_i \\ x \neq y}} d(x,y)$;
- Разстоянието като диаметър относно центъра. В този случай, то е два пъти средното разстояние между всеки елемент на клъстъра и клъстърния център, т.е. радиуса спрямо този център. $\Delta(K_i) = 2\left(\frac{\sum_{x \in K_i} d(x, \xi_i)}{n} \right)$, където ξ_i е център на клъстъра Ki.

Междуклъстърно разстояние (Inter-cluster distance) се нарича разстоянието между клъстърите и се пресмята по един от следните начини:

- Като единична връзка, която е най-близкото разстояние между две наблюдения, принадлежащи на два различни клъстъра K_i и K_j $\delta(K_i, K_j) = \min\left\{ d_{x \in K_i, y \in K_j}(x,y) \right\}$;
- Като цялостна връзка, която е най-отдалеченото разстояние между две наблюдения, принадлежащи на два различни клъстъра K_i и K_j $\delta(K_i, K_j) = \max\left\{ d_{x \in K_i, y \in K_j}(x,y) \right\}$;
- Като осреднена връзка, която е средно разстояние между всички наблюдения, принадлежащи на два различни клъстъра K_i и K_j $\delta(K_i, K_j) = \frac{1}{n_1 n_2} \sum_{x \in K_i, y \in K_j} d(K_i, K_j)$
- Като централна връзка, която е разстояние между центровете на два различни клъстъра K_i и K_j $\delta(K_i, K_j) = d(\xi_i, \xi_j)$.

Някои от методите на валидност, чрез които се оценява компактността на клъстърите и разстоянията между тях са:

- Индекс на валидност на Дейвис-Болдин **[**Bolshakova**,** G¨unter**]** приема ниска стойност, ако клъстърите са компактни и далеч един от друг
$$DB(K) = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \left\{ \frac{\Delta(K_i) + \Delta(K_j)}{\delta(K_i, K_j)} \right\}.$$
Ниската стойност показва добра клъстеризация.
- Профилен индекс [Rousseeuw]. Ширина на профила на *i*-тото набюдение от *j*-тия клъстър се изчислява по формулата
$$s_i^j = \frac{x_i^j - y_i^j}{\max\left\{x_i^j, y_i^j\right\}}.$$

От израза се вижда, че $-1 \le s_i^j \le 1$. Чрез него се дефинира профил на клъстър K_j

$$S_j = \frac{1}{m_j} \sum_{i=1}^{m_j} s_i^j .$$

Глобалният профилен индекс за клъстеризация се дава чрез

$$S = \frac{1}{n} \sum_{j=1}^{n} S_j .$$

Лесно се вижда, че профилът на клъстъра и глобалния профилен индекс приемат стойности между -1 и 1.

- Индекс на Дън [Dunn] се дефинира като частно на минималното междуклъстърно разстояние и максималното вътреклъстърно разстояние

$$D = \frac{\delta_{\min}}{\Delta_{\max}} .$$

Този индекс се ограничава в интервала $[0,\infty)$ и трябва да се максимизира.

- *C*-индекс [Hubert] се дефинира като

$$C = \frac{S - S_{\min}}{S_{\max} - S_{\min}} ,$$

където S е сума от разстоянията на всички двойки наблюдения от един и същи клъстър, m е броят на тези двойки, $S_{\min}$ е сума на m най-малки разстояния, ако се разгледат всички двойки наблюдения и $S_{\max}$ е сума на най-големите разстояния по всички двойки. Този индекс е ограничен в интервала $[0,1]$ и трябва да се минимизира.

Високата хомогенност в клъстерите и високата разнородност между клъстерите показват, че е постигнато добро групиране.

Ние се ограничаваме до проучване на случая с два клъстера, единият от които съответства на нормалната дейност, а другият на нарушенията. Логиката на този подход е предположението, че нормална дейност и аномалиите формират различни клъстери. Векторите на атаките често много си приличат, ако не са идентични. Очакването е, че клъстера на атаките в случай на масирана атака е изключително компактен.

3. Експерименти.

При експериментите са използвани данни за процеси (synthetic ftp, xlock, login, named, synthetic lpr), изпълнявани с администраторски права в Unix система [Forrest]. Разгледани са данни, включващи ID на изпълнявания процес, номера на системното извикване, данни за времето, през което е изпълняван процесът, изпълняваните инструкции от процесора между две извиквания на процеса, параметри, подадени на процеса, взаимодействието му с други процеси и др.

Използвана литература:

[1] [Al-Mamory] Al-Mamory S.O., H. Zhang, Intrusion detection alarms reduction using root cause analysis and clustering, *Computer Communications*, Volume 32, Issue 2, 12 February 2009, pp. 419–430.

[2] **[**Bolshakova**]** Bolshakova N. and Azuaje F., Cluster Validation Techniques for Genome Expression Data, *Signal Processing*, 83, 2003, pp. 825-833.

[3] [Davies] Davies, D.L., Bouldin, D.W., (2000) A cluster separation measure. *IEEE Trans. Pattern Anal. Machine Intell*., 1(4), 1979, 224-227.

[4] [Dunn] Dunn, 1974. Dunn, J. (1974) Well separated clusters and optimal fuzzy partitions, *Journal of Cybernetics* , 4, 95-104.

[5] [Ejaz] Ejaz A., S. Kashan, M. Waqar, Cluster-based Intrusion Detection (CBID) architecture for mobile ad hoc networks, In *5th Conference, AusCERT2006 Gold Coast, Australia, May 2006 Proceedings*, Gold Coast, Australia.

[6] [Forrest] Forrest S., S.A. Hofmeyr, A. Somayaji, T.A. Longtaff, A Sense of Self for Unix Processes. *In Proceedings of the 1996 IEEE Symposium on Security and Privacy, IEEE Computer Society Press*, Los Alamitors, CA, pp.120-128.

[7] [G¨unter] G¨unter S. and Bunke H., "Validation Indices for Graph Clustering", J. Jolion, W. Kropatsch, M. Vento (Eds.) *Proceedings of the 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, CUEN Ed., Italy, 2001, pp. 229-238.

[8] [Hubert] Hubert L, Schultz J. Quadratic assignment as a general data-analysis strategy . *British Journal of Mathematical and Statistical Psychologie*, 1976; 190-241.

[9] [Jaro] Jaro M. A., Advances in record linking methodology as applied to the 1985 census of Tampa Florida, *Journal of the American Statistical Society*, 1989, 414-420.

[10] Julisch K., Clustering Intrusion Detection Alarms to Support Root Cause Analysis, *ACM Transactions on Information and System Security*, Volume 6 Issue 4, November 2003, pp. 443 – 471.

[11] [Lieto] Lieto G., F. Orsini, G. Pagano, Cluster Analysis for Anomaly Detection, *CISIS 2008, ASC 53, Springer*, 2009, pp. 163–169.

[12] [MacQueen] MacQueen J., Some methods for classification and analysis of multivariate observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1967, pp. 281-297.

[13] [Monge 1] Monge, A., Elkan, C.. The field-matching problem: algorithm and applications. *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[14] [Monge 2] Monge, A., Elkan, C. An efficient domain-independent algorithm for detecting approximately duplicate database records. *In The proceedings of the SIGMOD 1997 workshop on data mining and knowledge discovery*, 1997.

[15] [Rousseeuw] Rousseeuw, P.J., Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics*, 20, 1987, 53-65.

[16] [Smith] Smith, Temple F.; Waterman, Michael S. Identification of Common Molecular Subsequences, *Journal of Molecular Biology* 147: 1981, 195–197.

[17] [Wagner] Wagner R. A., M. J. Fischer, "The string-to-string correction problem", *Journal of the Association for Computing Machinery 21*, pp. 168-173, 1974.

[18] [Winkler] Winkler W. E., The state of record linkage and current research problems, *Statistics of Income Division, Internal Revenue Service Publication* R99/04, 1999.