International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-7, Issue-7, July 2021
ISSN: 2395-3470
www.ijseas.com

Application of Clustering In Matchmaking

Solomon Sarpong

University of Environment and Sustainable Development, Somanya, Ghana

Abstract

Some persons make friends with others they have some attributes in common with. This type of friendship between two persons to a large extent depends on the attributes they have in common. In some cases, individual join a group as the individual thinks s/he identifies with that group. What then happens if a group of persons want to know how compactible they are with each other? From the conventional application of the existing matchmaking protocols, each member has to undergo a matchmaking protocol with each member in the group. This will be tedious and time consuming. It is in light of this, that this research was undertaking. This research enables a group of persons to know the number of attributes they have with each other and hence how compatible they are. The usage of k-means clustering and Floyd-Warshall's Algorithm were employed to achieve this. At the end of the protocol, each member in the group will know the number and the extent of compatibility they are with each other. Any malicious person, will know nothing.

Keywords: Clustering, *k*-means, privacy-preserving, asymmetry, Centroid.

1. Introduction

Personal attributes most often than not are private information about an individual. The personal attributes to a large extent define who an individual really is. Some of these attributes can be the individual's sexual orientation, circle of friends, religious inclination, type or nationality of their loved ones, schools attended, type of jobs, to mention just a few. Some persons do not want others to know their personal attributes as that defines them. Most often than not, people to a large extent make friends with people who have similar traits or similar orientation to attributes they have. How can individuals who do not want their attributes to be known by anyone make friends with people with similar characteristics? This has necessitated the need for some matchmaking protocols such as the; (i). use of central authority [1-3], (ii). distributed system [4-10] and (iii). hybrid system [11-19].

These protocols help an initiator find a match pair without each knowing the personal attributes of each other. Some of these protocols allow the people in the matchmaking know their attributes only when they are match paired. Furthermore, in order to prevent information asymmetry some of the protocols make the common attributes known to the users mutually. Others do not. In all these protocols, there should always be an initiator

(the person who wants to have a friend(s)). How will a group of people find how compatible (how many attributes they have in common) they are with each other? In such a case, using the conventional matchmaking protocols, it will take a lot of effort to achieve that. This is because, the people need to perform $(n-1)!$ permutations if there are *n* persons in the group.

Clustering techniques have been used in a number of studies. These include; i). identifying fake news, ii). Spam filtering, iii). Marketing and sales, iv). Classifying network traffic, v). identifying fraudulent and criminal activities, vi). Document analysis, vii). Fantasy football and sports.

It is in lieu of this problem that this paper proposes a protocol for the matchmaking among a group of persons. Hence, this protocol seeks to help a group of persons securely compute the intersection of their attributes. This will further help them know how compatible they are with each other. All of the persons in the protocol will know how compatible they are with each other simultaneously. As a results, there is no information asymmetry in this protocol.

2. Related Studies

In [20-21] oblivious transfer was used to construct private set intersection which was applied in matchmaking protocols. Kissner and Song [22] used threshold cryptosystem to solve set matching problems. Freedman *et al*. () [23] implemented oblivious pseudo-random function-based protocols in private set intersection and private cardinality of set intersection in attribute matchmaking protocol. Threshold cryptography was applied in PSI to propose a matchmaking protocol in which the intersection (number of common attributes) is satisfied if it is greater than a threshold agreed on by both parties [24]. Agrawal et al (2003) [25], used a commutative encryption function in matchmaking protocol. Sarpong and Xu (2015) [26] applied privacy-preserving scalar computation in secure attribute-based matchmaking protocol. Some researchers formulate different secured and privacy-preserving matchmaking protocols to help persons look for a match-pair.

3. Methodology

3.1 Introduction to *K*-means Clustering

K-means method has the ability to efficiently cluster huge data including outliers. It also maintains a basic framework for developing numerical or conceptual clustering systems because various possibilities of distance and prototype choice can be used, Salem *et al*., (2017). However, Cheung, (2003) [32] observed that *k*-means algorithm is very sensitive in the initial starting point hence it may lead to incorrect clustering results. As the initial clustering points are generated randomly hence, *k*-means does not guarantee a unique clustering results, Shehroz and Ahmad, (2004) [33] as it has the ability to reach a local minimum but lacks the ability to reach a global optimum (Kövesi *et al*., 2001) [34] observed. *K*-means clustering is a type of unsupervised learning,

whose goal is to cluster the data in order to find any groups (the number of groups represented by the variable *K*) in the data based on similarities between the constituents of the data. Hence, *K*-means clustering groups the data into its organic composition instead of allowing the researcher to define the groups.

K-means algorithm can be described as follows:

1. An initial cluster centers, c_k is randomly generated.
2. The distance $d(x,c)$ between vectors x_i to cluster center is calculated. In this research, the centroid method will be used.
3. Separate x_i into s_k which has minimum $d(x,c)$.
4. The new cluster centers are defined by $c_i = \frac{1}{p}\sum_{j=1}^{p} m(s_i,j)$ where $p = n(s_i)$.

3.1.1 Assigning data points to a cluster

Once the initial centroids are selected, the next step is to assign each data point to a cluster. Mathematically speaking, this can be done by using the equation:

$$S_i^{(t)} = \left\{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j\|^2\right\} \forall_j, 1 \leq j \leq k$$

where;

$S_i, 1 \leq i \leq n$, are the clusters, and m_i, m_j are the centroid values.

3.1.2 Calculating new centroid values

After each data point is assigned to a cluster, a new centroid value for each cluster is calculated (the mean of all the data points in the cluster). Hence using equation (1), new centroids are recomputed by reassigning new data points to the cluster until an optimal number is achieved.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|}\sum_{x_j \in S_i^{(t)}} x_j \qquad \text{Eqn. (1).}$$

3.1.3 *K*-Means Algorithm

Let $A = \{a_i | \ i = 1, \ldots, n\}$ be the number of people in the group and $a_i = \{x_{ij}, \ j = 1, \ldots r\}$ the attributes each member in *A* possesses. Using *K*-means, the centroid, s_i, is computed for each $x_i \in X$. Hence, there will be $\{s_1, s_2, \ldots, s_n\}$ centroids for the *n* persons in the group A. At this point, the attributes of each member in the group is represented by the centroid.

3.1.4 Distance Measures

Distance measures will be used to ascertain how compatible the members in the group are with each other. The

shorter the distance between two centroids, the more attributes they have in common hence, the more compatible they are with each other. From Figure 1, it can be observed that the distance between centroids 3 and

1 is shorter than that between centroids 2 and 3 or 1 and 2. Hence, the clusters with centroids 3 and 1 have more attributes in common than between the rest. In this protocol, Floyd-Warshall shortest distance approach will be used to measure the distance between the centroids of the attributes of the individuals.

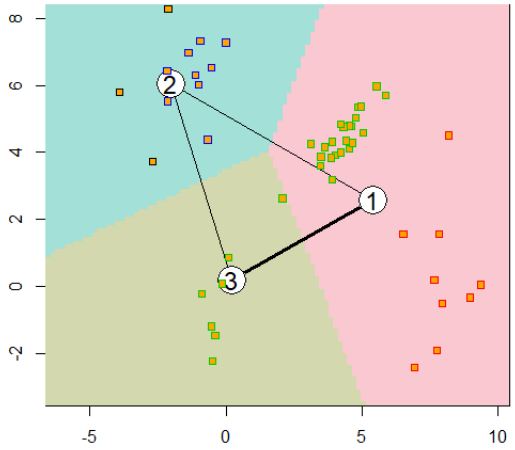

Figure 1: A typical clustering and distance between the centroids.

Floyd-Warshall's Algorithm

The Floyd-Warshall's algorithm as a variant of dynamic programming, solves problems by looking at the solution to be obtained as an interrelated decision. The solutions are formed from solutions that come from the previous stage and hence, there is the possibility of more than one solution [5]. Given a weighted digraph $G = (V, E)$ with weight function $w: E \rightarrow R$, where R is the set of real numbers.

Considering non-zero and negative cost, the shortest distance between vertices can be found using the Floyd-Warshall's algorithm. That is, let $n \times n$ matrix be a representation of a graph with weights at the edges,

$$W_{ij} = \begin{cases} 0, & \text{if } i = j \\ w(i,j), & \text{if } i = j \text{ and } (i,j) \in E \\ \infty, & \text{if } i \neq j \text{ and } (i,j) \notin E. \end{cases}$$

In the $n \times n$ matrix, let the distance $D = [d_{ij}]$ where d_{ij} is the distance from vertex i to j. Let $v_2, v_3, \ldots, v_{i-1}$ be the intermediate vertices of the path $p = \langle v_1, v_2, \ldots, v_n \rangle$. Let $d_{ij}^{(k)}$ be the length of the shortest path from i to j such that, all intermediate vertices on the path (if any) are in the set $\{1, 2, \ldots, k\}$. Set $d_{ij}^{(0)}$ to be w_{ij}, i.e. no intermediate vertex; $D^{(k)}$ be the $n \times n$ matrix $\left[d_{ij}^{(k)}\right]$. In this algorithm $d_{ij}^{(n)}$ is the distance from i to j hence, $D_{ij}^{(k)}, k = 0, 1, \ldots, n$ will be computed. However, it must be noted that; (i). if a shortest path does not contain the same vertex twice; (ii). For the shortest path from i to j such that any intermediate vertices on the path are chosen from the set $\{1, 2, \ldots, k\}$, there are two possibilities; (a). *k* is not a vertex on the path – the shortest of such paths has length $d_{ij}^{(k-1)}$; (b). if *k* is a vertex on the path – the shortest of such paths has length $d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$ i.e. there is a subpath from i to k and from k to j.

Each such subpath can only contain intermediate vertices in $\{1, 2, \ldots k-1\}$ and must be as short as possible. Hence, from i to j through an intermediate *k* has a length $d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$. The shortest path from i to j is the minimum distance from the points i to j and i to j through an intermediate step *k*. Mathematically, this shortest distance is given by $d_{ij}^{(k)} = min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$. The Floyd-Warshall algorithm has time complexity of $O(n^3)$, however when only one matrix is considered the time complexity reduces to $O(n^2)$. Assuming, there are $A = \{a_i | \, i = 1, \ldots, n\}$ number of people in the group and $a_i = \{x_{ij}, \, j = 1, \ldots r\}$ attributes of each member in the group. Hence an individual a_1 has

$\{x_{1j},\ j = 1,\ \ldots\ r\}$ attributes. Supposing there are a_n individuals in the group, there will be a total of $\{x_{nj},\ j = 1,\ \ldots\ r\}$ attributes.

Matchmaking Protocol

This protocol consists a certification authority, CA, that cannot be compromised and the group of individuals. Each of the persons in the protocol has a portable device or a Smartphone equipped with Bluetooth or WiFi. The

CA generates RSA key pair (e_{CA}, d_{CA}) and $N = pq$ where p and q are large primes. The CA publishes N and e_{CA}. Likewise, each of the persons in the protocol also generates RSA key pair (e_{a_i}, d_{a_i}) and $N_i = p_i q_i$ where $i = 1, 2, \ldots, n$; p_i and q_i are large primes. Furthermore, each of them chooses a username, $Username_i$, an identity, ID_i and a random number $R_{a_i} \leftarrow Z_{N/2}$. Each *User* sends the attributes to the CA for certification by sending $E_{e_{a_i}}[Attributes_{ij} \| ID_i \| R_{a_i} \| Username_i \| RSA\ public\ Key, e_{a_i}]$. The CA certifies the attributes by computing $\delta_i = sign_{d_{CA}}(ID_i \| Attributes_{ij})$ and returns the signed attributes to the *Users*. After the certification of the attributes by the CA, the *Users*' attributes becomes $\{(x_{ij}, \delta_{ij})\}$. Thus for the *User* one, the certified attributes become $\{(x_{11}, \delta_{11}), (x_{12}, \delta_{12}), \ldots, (x_{1r}, \delta_{1r})\}$. The certification prevents the users from modifying their attributes enabling them to gain additional information from the others in the protocol. In this protocol, attributes are the same if they are semantically the same. Each *User* exponentiates the attributes with the random number. The exponentiated attributes for the $\{a_i | i = 1, \ldots, n\}$ persons become $\left\{x_{i1}^{R_{a_i}}, x_{i2}^{R_{a_i}}, \ldots, x_{nr}^{R_{a_i}}\right\}$. Using *K*-means algorithm, the centroid is computed for the variables. The centroid is calculated for the variables belonging to each user. Hence, there will be C_n centroids from the $\{a_i | i = 1, \ldots, n\}$ users. Thus the number of centroids formed will be the same as the number of

users. Furthermore, the distance between the centroids formed are also calculated. Hence, the distance between *Users* one and two is given by d_{12}.

An $n \times n$ matrix as shown below is formed from the distances between the centroids of the attributes of the *Users* in the protocol.

$$\begin{array}{c|cccccc} & a_1 & a_2 & \cdot & \cdot & \cdot & a_n \\ a_1 & 0 & d_{12} & & & & d_{1n} \\ a_2 & d_{21} & 0 & & & & d_{2n} \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ a_n & d_{n1} & d_{n2} & \cdot & \cdot & \cdot & 0 \end{array}$$

Using the Floyd-Warshall's algorithm *n* times, the shortest paths from a centroid to the other are calculated by the CA. The matrix formed after the *n* iterations of the Floyd-Warshall's algorithm gives the shortest distance between the centroids of the individual attributes. From the matrix, the smaller the distance between the centroids, the closer the attributes are in similarity. After the computation of the shortest distance the CA furnishes the persons in the protocol with the matrix of the shortest distances. From this matrix, the *Users* will know how similar or otherwise their attributes are with each other. This knowledge will help the persons in the protocol to form match-pair if they want to.

For simplicity, let us assume the rest of the protocol is between only two *Users*. *User* one has a random number R_{a_1} and *User Two* also a random number R_{a_n}. At this point, the persons know only how close their attributes are related to each other from the matrix of the shortest distance. They also do not know how many or the type of attributes are common to them. *User One* sends $\left\{x_1^{R_{a_1}}, x_2^{R_{a_1}}, \ldots, x_r^{R_{a_1}}\right\}$ to *User Two. User Two* also sends $\left\{x_1^{R_{a_2}}, x_2^{R_{a_2}}, \ldots, x_r^{R_{a_2}}\right\}$ to *User One*. Furthermore, *User One* sends the random number, R_{a_1}, to *User Two; User Two* sends the random number, R_{a_n}, to *User One*. With the knowledge of the random numbers, *User One* computes $\left\{x_1^{R_{a_1}R_{a_2}}, x_2^{R_{a_1}R_{a_2}}, \ldots, x_r^{R_{a_1}R_{a_2}}\right\}$ and *User Two* also computes $\left\{x_1^{R_{a_2}R_{a_1}}, x_2^{R_{a_2}R_{a_1}}, \ldots, x_r^{R_{a_2}R_{a_1}}\right\}$. The intersection between $\left\{x_1^{R_{a_1}R_{a_2}}, x_2^{R_{a_1}R_{a_2}}, \ldots, x_r^{R_{a_1}R_{a_2}}\right\}$ and $\left\{x_1^{R_{a_2}R_{a_1}}, x_2^{R_{a_2}R_{a_1}}, \ldots, x_r^{R_{a_2}R_{a_1}}\right\}$ is computed. This intersection gives the *Users* the number of attributes common to them. Let $I_{12} = \left\{x_1^{R_{a_1}R_{a_2}}, x_2^{R_{a_1}R_{a_2}}, \ldots, x_r^{R_{a_1}R_{a_2}}\right\} \cap \left\{x_1^{R_{a_2}R_{a_1}}, x_2^{R_{a_2}R_{a_1}}, \ldots, x_r^{R_{a_2}R_{a_1}}\right\}$.

In order to know the type of attributes common to the *Users*, each *User* contacts the CA. *User One* sends his/her Username, identity and the identity of the pair together with the number of the attributes common to them to the CA. Thus, s/he sends $Sign_{a_1}\{ID_{a_1}, Username_{a_1} \| ID_{a_n}, Username_{a_n} \| I_{12}\}$ to the CA. Also, *User two* sends his/her Username, identity and the Username and identity of the pair together with the number of attributes common to them to the CA. Thus, s/he sends $Sign_{a_n}\{ID_{a_n}, Username_{a_n} \| ID_{a_1}, Username_{a_1} \| I_{12}\}$ to the CA. If I_{12} from *User One* is the same as that of *User Two*, then the protocol is correct hence, no one cheated. Upon receiving these and making sure that no one has cheated, the CA then sends the actual attributes common to them to each *User*.

Security Analysis

The protocol entails a CA that cannot be compromised. Also, only persons who have registered with the CA can take part in the protocol. The attributes of the persons in the protocol are certified to prevent their modification. A User can modify the attributes enabling him/her gain extra information on the other *User(s)*. Hence, the modification binds the attributes to the user. The *Users* get to know the number of and actual attributes common to them mutually. Hence, there is no information asymmetry in this protocol.

The degree of compatibility between the attributes of the *Users* are computed securely and privately. As a result, any aggressive malicious person can know only the degree of compatibility but nothing else. The matrix of the shortest distance gives the users the degree of compatibility they have with each other. The number of attributes common to them cannot be known at this point. The number of attributes common to them will be known only when they exchange their random numbers. When the intersection computed is an empty set, then the *Users* will know that there is something wrong with the protocol. This can be reported to the CA for the necessary action to be taken against the malicious *User*.

Conclusion

This research has brought to the fore the ability for a group of persons to be match-paired simultaneously without needing to do several match-pairing with each member in the group. The protocol in this research unlike in the conventional matchmaking protocols makes the matchmaking easier and less cumbersome.

References

[1] Peitilainen A. K., Oliver E., LeBrunm J., Varghese G. and Diot C. “Mobliclique: middleware for mobile social networking”. 2nd ACM Workshop on online social networks, WOSN, 2009, pg. no: 49-54.

[2] Kjeldskov J. and Paay J. “Just-for-Us: A context-aware mobile information system facilitating sociality”. In Proc. 7th International Conference on Human Computer Interaction with Mobile Devices and Services, 2005, pg. 23-30.

[3] Yang Z. Dai J., Champion A., Xuan D. and Li D. “Esmalltalker: A distributed mobile system for social networking in physical proximity” in IEEE ICDCS, 2010, pg. no: 468-477.

[4] Eagle N. and Pentland A. “Social serendipity: mobilizing social software”. In IEEE Pervasive Computing, Special Issue: The Smartphone, 2005, pg. no: 28-34.

[5] Li K., Sohn T., Huang S. and Griswold W. “PeopleTones: A system for the detection of notification of buddy proximity on mobile phones”. In Proc. 6th International Conference on Mobile Systems (Mobisys), 2008, pg. no: 160-173.

[6] Liu M. and Lou W. FindU: Privacy-preserving personal profile matching in mobile and social networks. Proc. of Infocom, 2011.

[7] Zhang L and Li X. “Message in a sealed Bottle: privacy-preserving friending in social networks”. 33rd IEEE Conference on distributed Computing Systems, 2013, IEEE Computer Society, pg. no: 327-336.

[8] Wang B., Li B. and Li H. “Gmatch: secure and privacy-preserving group matching in social network”. IEEE Global Communications Conference, GLOBECOM Anaheim, CA, USA: IEEE 2014, pg. no. 726-731.

[9] DeCristofaro E., Durussel A. and Aad I. “Reclaiming privacy for Smartphone applications”. In Proc. of Pervasive Computing and Communications (Per-Com), IEEE International, 2011, pg. no. 84-92.

[10] Lu R., Lin X. and Shen (Sherman) X. “SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-health emergency”. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(3), pg. no: 614-624.

[11] Wang Y., Wang H., Li T., Zhang T., and Jie H., “A privacy-preserving matchmaking scheme for multiple mobile social networks, algorithms and architecture for parallel processing”. 13th International Conference, ICA3PP, December 18-20, 2013, Springer pg. no: 233-240.

[12] Sarpong S., Xu C-H. and Zhang X., “An authenticated privacy-preserving attribute matchmaking protocol for mobile social networks”. International Journal of Network Security, 2015, 17(3) pg. no: 357-364.

[13] Sarpong S. and Xu C-H., “Privacy-preserving attribute matchmaking in proximity-based mobile social networks”. International Journal of Network Security and its Applications, 2015, 9(5) pg. no: 217-230.

[14] Sarpong S. and Xu C-H., “A collision-resistant privacy-preserving attribute matchmaking for mobile social networks”. International Journal of Innovative Science, Engineering and Technology, 2, pg. no: 485-495.

[15] Sarpong S. and Xu C-H., “A secure and efficient privacy-preserving attribute matchmaking protocol for proximity-based mobile social networks”. Advanced Data Mining and Applications (ADMA). 10th International Conference, Springer-Verlag, Springer LNCS 8933, pg. no: 305-318.

[16] Sarpong S. and Xu C-H., "A secure and efficient privacy-preserving matchmaking for mobile social networking". 1st International Conference on Computer, Network Security and Communication Engineering (CNSCE), DeStech Publications pg. no: 362-366.

[17] Liao X., Uluagnac A. S., Beyah R. A. "S-MATCH: Verifiable privacy-preserving profile matching for mobile social services". 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN, 2014, pg. no: 287-298.

[18] Kolesnikov V, Kamaresan R., Rosulek M. and Trieu N. Efficient batch oblivious PRF with applications to private set intersection. In CCS, 2016.

[19] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung, "Efficient robust private set intersection," International Journal of Applied Cryptology, vol. 2, no. 4, pp. 289–303, 2012.

[20] L. Kissner and D. X. Song, "Privacy-preserving set operations," in Advances in Cryptology (CRYPTO'05), pp. 241–257, 2005.

[21] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in Advances in Cryptology (EUROCRYPT'04), LNCS 2267, pp. 1–9, Springer, 2004.

[22] J. Camenisch and G. M. Zaverucha, "Private intersection of certified sets," in Financial Cryptography and Data Security, LNCS 5628, pp. 108–127, Springer, 2009.

[23] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," in Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 86–97, New York, USA, 2003

[24] Salem B. S., Naoualli S. and Sallami M., (2017). Clustering categorical data using the *k*-means algorithm and the attribute's relative frequency. World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering 11(6).

[25] Cheung Y-M., (2003). *K*-means: A new generalized *k*-means clustering algorithm. Pattern Recognition letters 24, pp. 2883-2893

[26] Khan S. S and Ahmad A (2004). Cluster center initialization algorithm for *K*-means clustering. Pattern Recognition Letters 25(11), pp. 1293-1302

[27] Kövesi B., Boucher J.-M and Saoudi S., (2001). Stochastic K-means Algorithm for vector quantization. Pattern Recognition Letters. 22, pp. 603-610. Doi:10.1016/S0167-8655(01)00021-6.

[28] R. Ramadiani, D. Bukhori, A. Azainil and N. Dengen (2018). Floyd-warshall algorithm to determine the shortest path based on android. IOP Conference Series Earth and Environmental Science 144(1): 012019.

[29] Ying Zhao, and George Karypis (2002). Evaluation of hierarchical clustering algorithms for document datasets. CIKM 2002 Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 515-524, https:// doi.org/10.1145/584792.584877.

[30] Tian Zhang, Raghu Ramakrishnan, Miron Livny (1996). BIRCH: An efficient data clustering method for very large databases. AGM SIGMOD Record Vol. 25, No. 2.

[31] P. Krishna Prasad and C. P. Rangan (2006). Privacy-preserving BIRCH Algorithm for Clustering over vertically partitioned databases. SDM, 2006, Proceedings of the Third VLDB International Conference on Secure data Management, pp. 84-99, *https://doi.org/10.100711844662_7***.**

[32] Sabastian Zander, Thuy Nguyen and Grenville Armitage (2005). Automated traffic classification and Application Identification using Machine Learning. Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN, 2005).

[33] Yong Ge, Hui Xiong, Chuanren Liu, Zhi-Hua Zhou (2011). A taxi driving fraud detection system. 11th IEEE International Conference on Data Mining Vancouver, BC, Canada.

[34] Zhou Xusheng, and Zhou Yu (2009). Application of clustering Algorithms in Ip Traffic classification. GCIS, Proceedings of 2009 WRI Global Congress on Intelligent Systems Vol. 2, pp. 399-403, *https://doi.org//10.1109/GCIS.2009.139***.**

[35] Jeffery Erman, Martin Arlitt, Anirban Mahanti, (2006). Traffic classification using clustering algorithms. MineNet 2006: proceedings of the 2006 SIGCOMM Workshop on Mining Network data, pp. 281-286, *https://doi.org/10.1145/1162678.1162679***.**