DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OUTLIER DETECTION WITH K NEAREST NEIGHBOR CLUSTERING

by
Yunus DOĞAN

August, 2009
İZMİR

OUTLIER DETECTION WITH K NEAREST NEIGHBOR CLUSTERING

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science in
Computer Engineering, Computer Engineering Program

by
Yunus DOĞAN

August, 2009
İZMİR

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitle "**OUTLIER DETECTION WITH K NEAREST NEIGHBOR CLUSTERING**" completed by **YUNUS DOĞAN** under supervision of **ASST. PROF. DR. GÖKHAN DALKILIÇ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

…….

Asst. Prof. Dr. Gökhan DALKILIÇ

_____

Supervisor

_____ _____

_____ _____

(Jury Member) (Jury Member)

_____

Prof.Dr. Cahit HELVACI

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENT

The author extends his sincere thanks to his supervisor Asst. Prof. Dr. Gökhan DALKILIÇ for his advice and guidance. This article has been produced through a master degree thesis to Graduate School of Natural and Applied Sciences, Dokuz Eylül University, and also this study has been supported as a BAP (Scientific Research Project). The project number is 2008.KB.FEN.020 and the name of this BAP Project is “Enhancing Security in Wireless Computer Networks”.

Yunus DOĞAN

OUTLIER DETECTION WITH K NEAREST NEIGHBOR CLUSTERING

ABSTRACT

A server which serves to wireless network needs strong security systems. For this goal, a new perspective to network security is won by using data mining paradigms like outlier detection, clustering and classification. This study uses k-Nearest Neighbor algorithm for clustering and classification. K- Nearest Neighbor algorithm needs data warehouse which impersonates user profiles to cluster. Therefore, requested time intervals and requested IPs with text mining are used for user profiles. Users in the network are clustered by calculating optimum k and threshold parameters of k-Nearest Neighbor algorithm with a new approach. Finally, over these clusters, new requests are separated as outlier or normal by different threshold values with different priority weight values and average similarities with different priority weight values.

Key words : outlier detection, k-Nearest Neighbor clustering, k-Nearest Neighbor classification, optimum k and threshold numbers, text mining

EN YAKIN K KOMŞU KÜMELEMESİ İLE AYKIRI DURUMLARIN TESPİTİ

ÖZ

Kablosuz ağ servisi yapan bir sunucu güçlü güvenlik sistemlerine ihtiyaç duymaktadır. Bu amaç için, ağ güvenliğine, aykırı durum tespiti, kümeleme ve sınıflandırma gibi veri madenciliği paradigmaları kullanılarak yeni bir perspektif kazandırılır. Bu çalışma hem ilk aşamadaki kümeleme hem de sonrasındaki sınıflandırma için en yakın k komşu algoritmasını kullanır. En yakın k komşu algoritması kümeleme için kullanıcı profillerini anlamlaştıran veri ambarına ihtiyaç duyar. Bu nedenle, sunucudan istekte bulunulduğu zaman aralıkları ve doküman madenciliğinden geçmiş, istek IP adresleri kullanılacaktır. Ağdaki kullanıcılar, yeni bir yaklaşımla, en yakın k komşu algoritmasının k ve eşik değer parametrelerinin uygun değerlerinin hesaplanması ile kümelenir. Sonuç olarak, oluşan bu kümeler üzerinden, öncelikli ağırlık değerleri farklı olan, farklı eşik değerlerle ve öncelikli ağırlık değerleri farklı olan ortalama benzerliklerle, yeni isteklerin bir aykırı durum mu yoksa normal mi oldukları ayırt edilebilecektir.

Anahtar sözcükler : aykırı durum tespiti, en yakın k komşu ile kümeleme, en yakın k komşu ile sınıflandırma, uygun k ve eşik değer parametreleri, dokuman madenciliği

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

1.1 Introduction

Security of wireless network connection has a dominant place in the computer science together with growing usage of Internet and also, usage of wireless network connection. Thus, this notion becomes a more important subject day by day. This important notion has been already discussed and researched scrupulously.

Portable devices like laptops and mobile phones in the target group of wireless network connection get more popular increasingly and users can surf on Internet freely everywhere. However, this freedom carries some security problems at the same time, which cut out advantage of wireless network connection. Since, wireless network connection allows forbidden listens and harmful attacks easily in spite of all inhibitive cryptographic software and personal data of an Internet surfer comes up against threat of being saisired. Being added alternative mechanisms, which can determine these hazardous behaviors and avoiding them consequently, become more necessary because of popular wireless network connections.

The main feature of this thesis is using data mining paradigm, another branch of computer science, as an alternative mechanism and by this means, determining the attacks and forbidden listens, and enhancing security in wireless network communication. As a result, it would be possible that dangerous attacks are stated in the wireless network connection by outlier detection techniques, one of four paradigms of data mining (classification, clustering, association rule mining and outlier detection), which is used for discovering exceptional situations successfully.

For outlier detection technique, clustering method would be used. One of the most successful classification algorithms, K-Nearest Neighbor is able to be used for clustering and outlier detection paradigm with some adaptations.

Owing to clustering method, the users with same profiles would be collected in the same cluster; therefore, when the system searches whether there has been an attack for a user, the system would not look at only the past of this user. Behaviors and profiles of all users in the cluster, where the user is put, are able to take into consideration by the system. Because then behaviors, which this user have not got but are similar to profile of this user, are paid attention. As a result, with this approach, the study would find out an outlier healthily and reliably.

The system has an improvement mechanism for K-Nearest Neighbor algorithm. According to the data set, the system decides that which clustering result would have the best distribution. The mechanism finds the optimum parameters of K-Nearest Neighbor according to some expectancy from the clusters and the contents of these clusters. These parameters and optimum values will be detailed on this thesis.

Ultimately, this thesis mentions that for a more secure wireless network connection, the server must have systematic mechanism which has data mining approaches and this mechanism controls user requests according to their time intervals and source IP addresses, thus the knowledge about whether it is a normal or an outlier is obtained from the data mining operations. As a result, this thesis is based on users at a wireless network and this study figures out outliers inside usages of wireless network.

1.2 Related works

At a work (Bloedorn & others, 2001), “network intrusion detection” and “data mining” notions are mentioned and by comparing the methods before using data mining approaches with the methods of using data mining for intrusion detection, the benefits of data mining approaches are pointed.

The outlier detection which is one of the data mining paradigms has five different algorithm approaches. These are distribution based, clustering based, density based, distance based and depth based. Three of these different algorithm

techniques are compared by a study of recent date (Lazarevic & others, 2002). These three algorithms are Nearest Neighbor Algorithm with k which equals to 1 for clustering based approach, Mahalanobis-distance Algorithm for distance based approach and Local Outlier Factor (LOF) algorithm for density based approach. The most successful approach according to intrusion detection rate finds clustering approach with Nearest Neighbor algorithm according to this study.

At a study of recent date about k-Nearest Neighbor classifier for intrusion detection (Liao & Vemuri, 2002), they applied the k-Nearest Neighbor classifier to the 1998 DARPA data which is collected as a large sample of computer attacks embedded in normal background traffic by the 1998 DARPA Intrusion Detection System Evaluation program (Evans, 2008). This data collection contains system calls which are treated as "words" in a document and processes which are created by sets of system calls and treated them as "documents" like the spectrum at a report about text categorization (Aas & Eikvil, 1999).

The k-Nearest Neighbor clustering algorithm has a handicap, because the question, of which k and threshold values must be taken, cannot find the answer with simple k-Nearest Neighbor algorithm. Like the research about average-case analysis of the k-Nearest Neighbor classifier for noisy domains (Nobuhiro & Okamoto, 1997), research about finding optimum k and threshold numbers are over k-Nearest Neighbor classification algorithm, not over k-Nearest Neighbor clustering algorithm.

There are two different approaches to intrusion detection as statistical anomaly detection and rule-based detection according to a study about Intrusion Detection (Porras, 1993 - referenced in Stallings, 2006). This study has statistical anomaly detection approach, because the study finds optimum parameters of k-Nearest Neighbor algorithm over time interval and requested IPs data collections like in threshold detection subject of statistical anomaly detection approach and the study clusters users according to the past behaviors like in profile-based anomaly

detection of statistical anomaly detection approach (Porras, 1993 - referenced in Stallings, 2006).

In this thesis, the optimum values of the parameters are found at k nearest neighbor clustering operation, because the changing of k number and threshold number changes the space of the clusters and the results of the outlier detection. Also, a recent study about clustering-based intrusion detection uses firstly trial and error method for clustering parameters; however, unsuccessful detection rates and false negative-positive rates are taken. Therefore, determining the ideal parameters is preferred (Wen-chao & Huan, 2004).

The one of the goals in this study is finding out the profiles of the users in the system, because the clustering operation is done according to this knowledge and the relationships. At a recent study about profiling and clustering internet hosts, the profiles are determined according to the network data like identifying host services and identifying TCP connection. Then, the users are clustered with the certain boundaries without using an algorithm. (Wei, Mirkovic & Kissel, 2006). This thesis uses the time interval and requested IP data for finding out the profiles. Also, k nearest neighbor clustering algorithm is used for clustering part.

In this thesis, outlier situations of the requests control and are detected according to the profiles and cluster space. A recent study about network traffic anomaly detection with using k-means clustering also focuses on Network Data Mining (NDM); however, the clustering operation is done with k-means algorithm independently from user profiles. The clustering operation is used over the network traffic characteristics (Münz, Li & Carle, 2007). This thesis uses k nearest neighbor clustering over the user profiles.

1.3 Methodology

Outlier detection, one of the data mining paradigms, which is used in this thesis, needs a data set in an order which is prepared by one of the other paradigms.

Therefore, new data which comes to the system is understood as outlier or normal by this processed data set. It can be said that there are two main cycles on the development duration of the project. On the first cycle, ordered data set is prepared and on the second cycle, the necessary mechanism which catches harmful attack and called “it can be outlier” is arranged. As a result, all cycles according to a study about Knowledge Discovery in Databases (Fayyad, Piatetsky-Shapiro & Smyth, 1996) are implemented in this thesis.

On the first cycle, while data set is preparing, it is decided this question as which paradigm the best choice would be. Clustering paradigm would be the best choice because this technique would prevent that limited set number. If classification method was chosen, wrong sets would be able to be created. The most important goal of this cycle is that specific and distinctive profiles of all users in our network system would be determined, and then these common or similar users would be in the same set. Therefore, attributes which are dissociated the users in the network are stated on this cycle, too. Also, algorithm and result analysis mechanism which would have been used for outlier detection cycle at the end of clustering process are thought and planed on the first cycle of the project.

At this study, for two different typed data sets, the outlier detection system has to give an answer. These data sets would have time intervals as numeric values and requested IPs in these time intervals simultaneously as textual values. For these different typed data sets, two different data warehouses have to be created. After data warehouses are collected, clustering operation has to be done with k-Nearest Neighbor algorithm on the first cycle. K-Nearest Neighbor algorithm is generally used for classification and it is very successful at classification and also, this algorithm will be used for anomaly detection by classification. Therefore, k-Nearest Neighbor algorithm is chosen.

For k-Nearest Neighbor algorithm, optimum k and threshold numbers are the most important parameters, because according to k and threshold numbers, k-Nearest Neighbor algorithm gets different clusters for the same data warehouse.

Therefore, for the best clustering result, an alternative approach is used and without “trial and error” method, the optimum k and threshold numbers are found. The thoughts, which bring us these optimum numbers, are that elements with the most similar scores have to be in the same clusters and these clusters total number have to be the minimum as possible. After the optimum k and threshold numbers are found, the clusters are created according to these values and the first cycle of the methodology is completed.

On the second cycle, k-Nearest Neighbor algorithm is used with classification paradigm and with this algorithm outlier requests are detected according to clusters and threshold value. Because over the clusters, it is calculated that whether new requests of the user are either fit or not, or similarity score is either less than threshold or not.

Briefly, the study has a double-sided outlier control mechanism. Firstly, outlier controls started according to clustering over data warehouse of time interval. If there is not an unconvincing situation, controls are terminated by the mechanism. However, if there is a problem at requires according to these time intervals for a user, the server tends to control outlier detection according to clustering over data warehouse of required IPs and the other side of the outlier detection mechanism is put into use step in. As a result, anomalies are caught by k-Nearest Neighbor classification.

CHAPTER TWO
EXPERIMENTS

The data sets in the study are collected from a business place where workers work between 9 am and 5 pm for four weeks. Data except this time interval shows that either these users have gone on working overtime or they have connected the network by the remote access. In addition, the following operations except tests subject (2.6 Outlier Detection with Different Priority Weight Values and Tests) in this chapter are implemented over a part of the real data collection as a week. The test operations are done for all 86 users in the network according to the data sets for four weeks.

2.1 Data Sets

2.1.1 Data Set of Time Intervals

Alternatively, as data set, one part of our K-Nearest Neighbor clustering algorithm uses request times and the other part of algorithm uses a data collection which contains required IPs differently from 1998 DARPA Datasets which are used in the study about k-Nearest Neighbor classifier for intrusion detection (Liao & Vemuri, 2002); however, a similar approach like this study (Liao & Vemuri, 2002) is used for requested IP data warehouse. Each required IP is related to a “term” in a document and each user, who requires the IPs, is related to a “document” according to the text processing metaphor at a book which is published of recent date (Manning, Raghavan & Schütze, 2008).

For accustomed Internet usages of the Internet surfers which means profile patterns, which attributes would be necessary had to be determined. Usage scores of all users would have to be calculated by a similarity function of clustering paradigm according to these attributes. And basically, according to these scores, similar wireless network users would be put in the same set.

As an answer to the question of which attributes would be based on, has been decided like the following ideas; surely, begin time and frequency of the Internet surfing had to be known. Of course, a wireless network user could use Internet in any time and after a certain interval, the user could leave from Internet. If that situation, these processes have to be done on a day, has been considered, being put only the begin time and frequency of the Internet surfing would not be enough. Therefore, each clock interval for a user has to be determined as an attribute and like in the following Table 2.1; vectors data for three Internet surfers of the wireless network system would be obtained.

Table 2.1 Snap frequency list with binary time interval values

Clock Intervals / **Users**	00-01 (bin.)	01-02 (bin.)	02-03 (bin.)	**	21-22 (bin.)	22-23 (bin.)	23-00 (bin.)
C	0	0	0	*	0	1	0
B	0	0	0	*	1	0	0
C	1	0	0	*	0	0	0
A	0	0	0	*	0	0	1
B	0	0	0	*	1	0	0
A	0	0	1	*	0	0	0
***	*	*	*	*	*	*	*

(*) : sequence of binary numbers, (**) : sequence of time interval from 03-04 to 20-21, (***) : sequence of users, (bin.) : binary number

This table shows all clock intervals on a day and according to these intervals, the table lists the vectors which put request time for all users who have required a web data from the central server. All vectors put a “1” and twenty three “0”. If a user requires a web data, system adds a new vector according to user and clock interval onto the table.

Time intervals are taken as only twenty four clocks in the book about intrusion detection (Frincke, 2002). However, Internet surfers behave differently during weekend days and weekdays. Therefore, if weekend days and weekdays attributes are separated as from 00-01 to 23-24 for weekend days and from 24-25 to 47-48 for weekdays, more sensitive result will be obtained at clustering cycle. The following Table 2.2 shows the list of request vectors.

Table 2.2 Snap frequency list with binary time interval values which are separated as weekend and week days

Clock Intervals / **Users**	00-01 (bin.)	01-02 (bin.)	**	23-24 (bin.)	24-25 (bin.)	25-26 (bin.)	**	47-48 (bin.)
C	0	0	*	0	0	0	*	0
B	0	0	*	0	0	0	*	1
C	1	0	*	0	0	0	*	0
A	0	0	*	0	0	0	*	0
B	1	0	*	0	0	0	*	0
A	0	0	*	0	0	0	*	1
***	*	*	*	*	*	*	*	*

(*) : sequence of binary numbers, (**) : sequences of time intervals from 02-03 to 22-23 and from 26-27 to 46-47, (***) : sequence of users, (bin.) : binary number

Finally, all data is able to be thought as 48 numbers in 48 attributes and by listening to the wireless network system is interrupted after a certain days, numbers in the vectors of the same Internet users are able to be accumulated one by one. Then the result data set which will be used by the clustering algorithm is created. For instance wireless network system which contains three Internet surfers, result vectors list is like in the following Table 2.3 and each vector gives an idea with total weight values as frequencies for profile of a user.

Table 2.3 The last situation of frequency list with total decimal time interval values which are separated as weekend and week days

Clock Intervals / **Users**	00-01 (dec.)	01-02 (dec.)	**	23-24 (dec.)	24-25 (dec.)	25-26 (dec.)	**	47-48 (dec.)
C	20	56	*	12	18	3	*	10
B	7	19	*	55	4	6	*	78
A	3	7	*	2	8	22	*	69

(*) : sequence of decimal numbers, (**) : sequences of time intervals from 03-04 to 22-23 and from 26-27 to 46-47, (dec.) : decimal number

While listening to the wireless network system at central server, 48 digits for each request are saved in this central server and one of these 48 digits is meaningful. Therefore, holding other 47 digits for each vector occupies too large place on disk during collecting more vectors. Instead of this method, the central

server must save only order number of digit and thus, especially at long listening, probable disk area problem will not accrue. Each data collection area for a user contains order numbers which shows the place in the 48 digits time sequence for a user like in the following Table 2.4.

Table 2.4 Sequences of time interval values between 0 and 47 in the data collection

Users	**Clock Digits** (between 0 and 47)														
A	22	47	47	47	47	47	47	47	12	12	12	12	22	22	*
B	24	24	24	24	25	25	25	25	36	36	7	7	7	7	*
C	25	25	47	47	47	25	1	1	13	13	13	4	4	4	*

(*) : sequence of clock digits between 0 and 47

2.1.2 Data Set of Requested IPs

The data set, which is obtained in each clock interval for a user as an attribute, is not enough satisfying to cluster users according to their profile. In other words, deciding “It is an attack” with only clock interval for a request from wireless network connection may be specious. Therefore, the central server needs alternative data set besides clock interval values like contents of requests.

If an Internet surfer in the wireless network requires a web site from the central server, this server is able to determine which web site is wanted and this data is able to be saved by the server. On this way, the central server, which serves wireless network connection, saves requested IP addresses for each user while saving clock interval data of these requests. These requested IP addresses will be used for a different clustering operation distinct from clock interval clustering.

Like in the following Table 2.5, requested IP addresses are collected for all users as a document in text mining metaphor.

Table 2.5 Sequences of requested IP addresses in the data collection

Users	IP Addresses				
A	20.*.218	19.*.12	12.*.21	22.*.149	*
B	61.*.211	12.*.222	81.*.93	24.*.105	*
C	24.*.105	19.*.5	87.*.48	19.*.12	*

(*) : sequence of requested IP addresses

All these data sets have been collected from a real system for four weeks and two data warehouses of time intervals and requested IPs have been obtained simultaneously. For clean data, "ignore the tuple" method as a data mining data cleaning technique is used and data warehouses have been purified from noisy data. As a result, all data has been passed the data preparation cycles (data integration, data selection, data preprocessing and data transformation) according to the study about Knowledge Discovery in Databases (Fayyad & others, 1996).

2.2 Vector Space Model over Requested IPs

Requested IP addresses listed in the data warehouse which is prepared to be clustered by k-Nearest Neighbor algorithm are not able to be used directly. This data set must be converted to a matrix form. Therefore, Vector Space Model (Manning & others, 2008) which is a text mining and information retrieval technique and is generally decided by search engines is selected.

This approach uses IP requests and by using IP requests, anomalies can be detected with text mining techniques. For this aim, each IP request is related to a "term" in a document and each user, who requires the IPs, is related to a "document" like at the book about information retrieval (Manning & others, 2008). The documents contain sequential IP addresses and each document symbolizes a user in the wireless network connection. Therefore, each document is able to be shown as a vector and the data warehouse is able to be shown also as a vector array like the example in the following Table 2.6.

Table 2.6 A part of the data collection which contains requested IP address

Users	A	B	C	
	19.*.12	61.*.211	21.*.69	21.*.69
	19.*.12	12.*.222	24.*.105	24.*.105
	20.*.218	81.*.93	21.*.69	21.*.69
	19.*.12	20.*.218	21.*.69	21.*.69
	12.*.21	24.*.105	21.*.69	21.*.69
	19.*.12	19.*.5	87.*.48	87.*.48
	22.*.149	93.*.9	21.*.69	21.*.69
	19.*.12	84.*.12	21.*.69	
Requested	61.*.211	61.*.211	21.*.69	
IP	61.*.211	93.*.9	19.*.5	
Addresses	24.*.105	84.*.12	87.*.48	
	20.*.218	81.*.93	81.*.93	
	19.*.12	19.*.12	21.*.69	
	20.*.218	20.*.218	21.*.69	
	19.*.12	24.*.105	84.*.12	
	12.*.21	19.*.5	21.*.69	
	19.*.12	93.*.9	84.*.12	

The vector array at the Vector Space Model, which will be used for k-Nearest Neighbor clustering, is created according to information retrieval weighting technique. This technique firstly creates a two-dimensional array or another word, a vector array. One dimension is for terms (requested IPs) and the other dimension is for documents (users in the wireless network system). This matrix declares term frequencies which mean total numbers of same IPs for each user like the following example Table 2.7;

Table 2.7 Term frequency values for IP address dictionary for user A, B and C

Users / **IPs**	A (frequencies)	B (frequencies)	C (frequencies)
19.*.12	8	1	0
20.*.218	3	2	0
12.*.21	2	0	0
22.*.149	1	0	0
61.*.211	2	2	0
24.*.105	1	2	2
12.*.222	0	1	0
81.*.93	0	1	1
19.*.5	0	2	1
93.*.9	0	3	0
84.*.12	0	2	2
21.*.69	0	0	15
87.*.48	0	0	4

Then, for results with weight values, some mathematical operations like calculating tf as term frequency, idf as inverse document frequency of the text matrix and taking normalization with the cosine similarity formula are done.

$tf_{t,d}$ is the term frequency of term t in document d and it means the number of times that t occurs in document d. The aim of the term frequency is to put numbers in smaller values. The weight of the term frequency is calculated like the following formula (Manning & others, 2008);

$$tf = \begin{cases} \text{if } tf_{t,d} > 0,\ 1 + \log_{10}(tf_{t,d}) \\ \text{else, } 0 \end{cases}$$

The situation after calculating term frequency of our example becomes like the Table 2.8.

Another weighting operation is taking document frequency. df_t is the document frequency of term t and it means the number of documents that contain term t. The weight of the document frequency is calculated with idf_t and by where N is the total document number in our collection. idf_t is calculated as in the following formula; $idf_t = \log_{10}(N/df_t)$. The situation after calculating inverse document frequency of our example table is given in the Table 2.9.

Table 2.8 The logarithmic weight of the term frequency for user A, B and C

Users / **IPs**	A (log frequencies)	B (log frequencies)	C (log frequencies)
19.*.12	1.90309	1.00000	0.00000
20.*.218	1.47712	1.30103	0.00000
12.*.21	1.30103	0.00000	0.00000
22.*.149	1.00000	0.00000	0.00000
61.*.211	1.30103	1.30103	0.00000
24.*.105	1.00000	1.30103	1.30103
12.*.222	0.00000	1.00000	0.00000
81.*.93	0.00000	1.30103	1.00000
19.*.5	0.00000	1.30103	1.00000
93.*.9	0.00000	1.47712	0.00000
84.*.12	0.00000	1.30103	1.30103
21.*.69	0.00000	0.00000	2.17609
87.*.48	0.00000	0.00000	1.47712

Table 2.9 Inverse document frequency of IP dictionary

IPs	**idf Values of IPs** (log scores)
19.*.12	0.17609
20.*.218	0.17609
12.*.21	0.47712
22.*.149	0.47712
61.*.211	0.17609
24.*.105	0.00000
12.*.222	0.47712
81.*.93	0.17609
19.*.5	0.17609
93.*.9	0.47712
84.*.12	0.17609
21.*.69	0.47712
87.*.48	0.47712

Result of weight values come with $tf_{d,t}$ x idf_t which is defined as the product of term frequency weight and inverse document frequency weight. It gives the weight of term t in document d and it is calculated like the following formula (Manning & others, 2008);

$$W_{t,d} = (1 + \log_{10} tf_{t,d}) \quad x \quad \log_{10}(N / df_t)$$

The situation after calculating weight $tf_{d,t}$ x idf_t of our example is given in the Table 2.10.

Table 2.10 $W_{t,d}$ values of user A, B and C before normalization

Users / **IPs**	A (log weights)	B (log weights)	C (log weights)
19.*.12	0.33511	0.17609	0.00000
20.*.218	0.26010	0.22910	0.00000
12.*.21	0.62074	0.00000	0.00000
22.*.149	0.47712	0.00000	0.00000
61.*.211	0.22910	0.22910	0.00000
24.*.105	0.00000	0.00000	0.00000
12.*.222	0.00000	0.47712	0.00000
81.*.93	0.00000	0.22910	0.17609
19.*.5	0.00000	0.22910	0.17609
93.*.9	0.00000	0.70476	0.00000
84.*.12	0.00000	0.22910	0.22910
21.*.69	0.00000	0.00000	1.03825
87.*.48	0.00000	0.00000	0.70476

Finally, a normalization operation must be implemented over the result matrix values, because these numeric values are independent from each other and they must be accumulated between two common numbers as 0 and 1. For the normalization operation, the following cosine similarity formula is used while m is the number of total IPs (Manning & others, 2008);

$$W_{t,d} = W_{t,d} \times \frac{1}{\sqrt{W_1^2 + W_2^2 + W_3^2 + \ldots + W_m^2}}$$

The situation which has values between 0 and 1 after calculating cosine normalization of our example can be seen in the Table 2.11.

Table 2.11 $W_{t,d}$ values of user A, B and C after normalization between 0 and 1

Users / **IPs**	A (between 0 and 1)	B (between 0 and 1)	C (between 0 and 1)
19.*.12	0.36446	0.17454	0.00000
20.*.218	0.28288	0.22708	0.00000
12.*.21	0.67511	0.00000	0.00000
22.*.149	0.51891	0.00000	0.00000
61.*.211	0.24916	0.22708	0.00000
24.*.105	0.00000	0.00000	0.00000
12.*.222	0.00000	0.47293	0.00000
81.*.93	0.00000	0.22708	0.13548
19.*.5	0.00000	0.22708	0.13548
93.*.9	0.00000	0.69858	0.00000
84.*.12	0.00000	0.22708	0.17627
21.*.69	0.00000	0.00000	0.79885
87.*.48	0.00000	0.00000	0.54225

If normalization operation is not used; by using the following Euclidean distance formula (as X and Y are the users, X_i and Y_i are weight values of their requested IPs), the similarities of the users are calculated (Manning & others, 2008).

$$Sim(X, Y) = \frac{\sum (Xi * Yi)}{\sqrt{X_1^2 + X_2^2 + \ldots + Xi^2} \times \sqrt{Y_1^2 + Y_2^2 + \ldots + Yi^2}}$$

However, the normalization is used, and by excluding the denominator of Euclidean distance formula and using only the fraction (∑ (Xi * Yi)), the similarities of the users are calculated and the similarity matrix of our example users is created like the Table 2.12 (Manning & others, 2008);

Table 2.12 The similarity matrix of user A, B and C according to requested IPs

Users	A (%)	B (%)	C (%)
A	100	18.44	0
B	18.44	100	10.15
C	0	10.15	100

If this vector space model as a method for creating the similarity matrix is not used, the matrix with only IP frequencies is normalized directly and Euclidean distance formula is applied to this normalized matrix. However, this basic method is not a healthy way, because without vector space model, there is no connection between same requested IPs by different users and these values are not heeded. Thus, similarity results in the final similarity matrix do not become successful.

2.3 Clustering with K- Nearest Neighbor

K-Nearest Neighbor (KNN) is a successful classification algorithm; therefore, an adapted situation of this algorithm is usually used at intrusion detection techniques (Liao & Vemuri, 2002). This study does not use k-Nearest Neighbor as a classifier; it is used as a clustering method over users in the wireless network system.

While doing the classification with KNN, class numbers and types are apparent before the operation. A new data must be in one of these apparent classes. However, while clustering with KNN, the number of groups is not certain and it is able to change with some parameters. These parameters are k and threshold numbers. The k is the number of the nearest neighbors which are controlled for each user. In other words, for a user, k users which have the most similar profile are looked by KNN algorithm. However, the similarities have to be greater than the threshold number, and they have to be calculated before the usage of KNN and then

they are collected in a matrix. Euclidean distance formula is used to calculate the similarities of the users in the system and a matrix which contains numbers between 0 and 1 is obtained.

This study has a double-sided control mechanism with requested IPs and their time intervals. In other words, the system has to have two separated matrices which are created by Euclidean distance formula and contain similarity values of all users for these two data sets. Thus, these two separated data sets are converted to a common data form and are able to be used by the same KNN clustering algorithm.

Searching the nearest neighbors in a large matrix is able to cause a performance problem. Therefore, sorting the similarities for each user by using quick sort algorithm before these matrices are used by KNN clustering would increase the performance of KNN clustering algorithm. Before the quick sorted matrix is used, another arrangement operation has to be done. In this operation, it is assumed that all users are in a different cluster and for this aim; a separated array from the matrix is used for putting these different clusters of the users. This array puts clusters as a number and the indexes of the array correspond to real index variable of quick sorted matrix structure. Because during quick sorting operation, users' index in the matrix changes absolutely and this information has not to be lost. As a result, KNN clustering function has to take four parameters; a quick sorted matrix, an array for putting cluster, k number and threshold number.

The function has two main loops; the outer loop repeats until there is no change for clustering. The other loop is for each user's cluster which the user belongs to and it repeats for each user. Firstly, the algorithm takes k cluster names of the most similar neighbors of the user in the inner loop by controlling with the threshold value. With the quick sorted matrix, this operation would be faster. Then, the algorithm determines the most frequent cluster name from these cluster names. Finally, this user is joined in this cluster. For all users, these operations repeat. After the inner main loop finishes, this situation controls if there is not a cluster changing. If there is a cluster changing, the inner loop starts to repeat again with

new clusters of all users. If there is not a cluster change, KNN clustering is terminated by the system. The KNN clustering function is represented as in the following pseudo code;

```
Function KNN_Clustering(parameters: quick sorted matrix, an array
for putting clusters, k number and threshold number)
     Do
           For each user:
                 For each index of quick sorted matrix from 1 to k
                       Get neighbor users to an array with their
                       cluster names
           Find the most frequented cluster in the array which puts
           the clusters of the nearest neighbors
           Update the clusters of the user and the users who are in
           the same cluster with this user
     While changing positions of users in the clusters
     Return the array which puts clusters
```

2.4 Finding Optimum K and Threshold Numbers

K-Nearest Neighbor algorithm gets different cluster results for the same data set according to k and threshold numbers. Therefore, deciding k and threshold numbers is very important for the following outlier operation. In other outlier detection studies which use KNN algorithm, k and threshold numbers are decided with trial or error method (Wen-chao & Huan, 2004). If an intelligent system is used for deciding these parameters, more healthy and reliable results can be taken from the following outlier operations over the correct clusters. For this aim, it is important to find the answer of this question: “What are the features of a successful clustering?”

Firstly, a successful clustering operation does not include a lot of clusters. Possibly, the number of clusters must be low. However, the limited number of clusters has to have unrelated users. In other words, unrelated users do not have to be in the same clusters. If there are a lot of clusters, interested users will not be in the same cluster and this situation is also an unwanted result. Therefore, optimum values of k and threshold numbers must be searched in a balanced. For this aim, an error rate formula is used by the system, then k and threshold numbers which give the least error rate are preferred. Finally, these parameters are used for the clustering.

Having low or high number of clusters changes according to the total number of users. Therefore, if having a low number is wanted, (the number of clusters / the number of users) must be directly proportional to error rate, because this situation increases error. If the number of clusters is equal to the number of users, the result of this parameter would be “1”; if all users are collected into only one cluster, the result of this parameter would be a value near “0”. However, this event is also an unwanted clustering result. Therefore, there must be another parameter based on similarities of users in the same cluster. Thus, the error rate of this situation with single cluster would increase.

If being the users with low similarities in the different clusters is wanted, geometric mean according to similarities in the matrix of the all clusters are taken one by one and sum of them is taken. After that, for taking arithmetic mean of this sum, this sum of geometric means of the clusters is divided by the number of clusters. If the result is a great value, it means the most similar users are in the same cluster; therefore total geometric means of all clusters / the number of clusters is inversely proportional to error rate calculation. If all users are in the different clusters, this value would be “1”; if all users are collected into only one cluster, the result of this parameter would be a value near “0”. As a result, these two parameters must be used together in error rate formula as the following;

Error Rate = (the number of clusters / the number of users) x (the number of clusters / the total of geometric means according to the similarities in the matrix of all clusters)

In the study, firstly, KNN clustering operation is done for all combinations with all k and threshold numbers one by one and then, error rate scores of all combinations are calculated by the error rate formula. Finally, according to the aim of smaller error rate, the biggest threshold and the smallest k number in the combinations which give the smallest error rate are chosen. Thus, with the smallest k number, the nearest neighbors are found faster and with the biggest threshold, the most similar users are put in the same cluster. In this thought, the biggest threshold

and the smallest k number in different combinations which give the equal smallest error rate are chosen as the optimum k and threshold numbers for the data set. Outlier detection operations would be done over the data set which is clustered by KNN algorithm with these optimum values of threshold and k numbers. Thus, for outlier requests, the most successful results are obtained.

The following pseudo code takes data set as a matrix, an array for cluster numbers like in KNN clustering function, total k and threshold numbers, and then returns the best clustering situation for this data set.

```
Function Find_Optimum_K_Threshold(parameters: quick sorted matrix,
an array for putting clusters, possible total k number and possible
total threshold number)
    Build a Struct array for all combinations of threshold and k
    numbers
    For each combination:
          Cluster by KNN(parameters: quick sorted matrix, an array
          for putting clusters, k number of current combination
          and threshold number of current combination)
          For each cluster:
                Calculate error rate score by product similarities
                from quick sorted matrix
                Calculate sum of these scores
          Calculate Geometric Mean of the scores of all clusters
          with cluster numbers as root.
          Calculate the total error rate by (the cluster number /
          the score which calculated by Geometric Mean) x (the
          cluster number / the user number)
    Find the index of the combination array which has the smallest
    error rate and also has the smallest k number and the biggest
    threshold in the combinations which have the same error rate.
    Return the optimum k and threshold numbers
```

The following Figure 2.1 which prepared in the MATHLAB 7.6 shows error rates according to the 120 combinations of 12 k numbers and 10 threshold numbers for a part of the data collection of the time intervals for a week and for 29 users.

For threshold from 0.0 to 0.6, error rate parabola shows the same characteristic features and the smallest error rate is caught when k is 1 and 2 as about 0.33. For threshold values 0.7, 0.8, 0.9, the smallest error rates are 0.4, 0.21 and 0.27 when k numbers are 1, 1 and 5 respectively. Thus, it is seen that the optimum k is 5 and the optimum threshold number is 0.8.

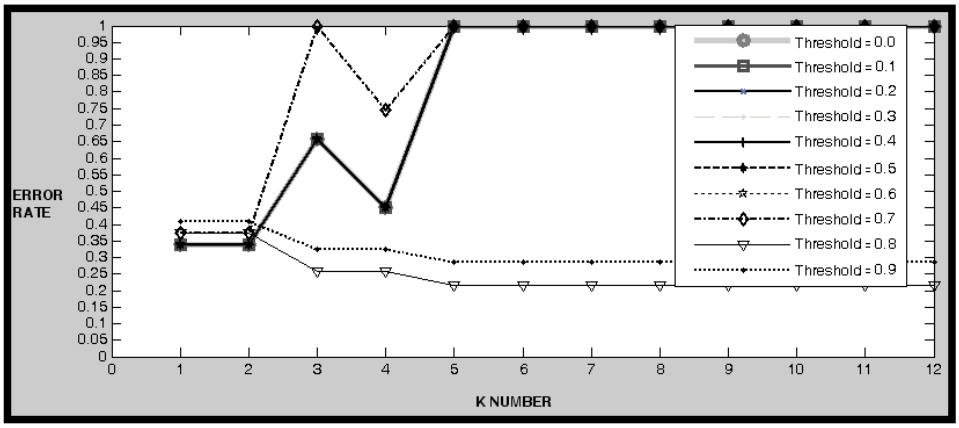

Figure 2.1 Error rates according to the 120 combinations of 12 k numbers and 10 threshold numbers for a part of the data collection of the time intervals for a week and for 25 users

The following Figure 2.2 which prepared in the MATHLAB 7.6 shows error rates according to the 120 combinations of 12 k numbers and 10 threshold numbers for a part of the data collection of the requested IPs for a week and for 25 users;

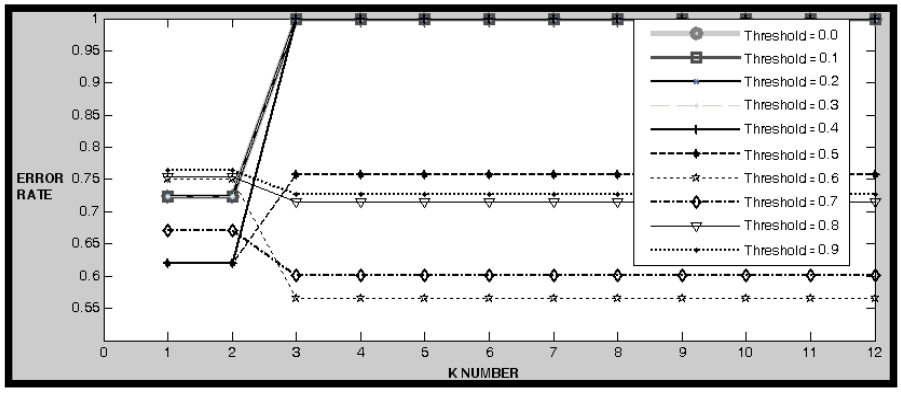

Figure 2.2 Error rates according to the 120 combinations of 12 k numbers and 10 threshold numbers for a part of the data collection of the requested IPs for a week and for 25 users

For threshold from 0.0 to 0.3, error rate parabola shows the same characteristic features and the smallest error rate is caught when k is 1 and 2 as about 0.73. For threshold values 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9 the smallest error rates are 0.62, 0.62, 0.55, 0.60, 0.71 and 0.72 when k numbers are 1, 1, 3, 3, 3, and 3 respectively. Thus, it is seen that to be realized the optimum k is 3 and the optimum threshold number is 0.6.

2.5 Outlier Detection with Double-sided Control Mechanism

When the system constitutes the cluster structures which are created with the optimum k and threshold numbers according to the two different data warehouses, the spinal of the double-sided control mechanism for outlier detection would be obtained. Because controlling operations of the data which will be tested for outlier does over the clusters which are created by the KNN. Firstly, while the wireless network is being listened, the data which will be tested is collected at regular intervals according to both time interval frequencies and requested IPs and then, the user data in the new data collection is tested by the following pseudo code. This function has a double-sided control mechanism. In other words, this function is used for both data collections which contain time interval frequencies or requested IPs.

The function takes two parameters as the array which contains the clusters and the data which will be tested and returns the usual score for outlier detection as percentage. There are two if blocks for two data test types. If the data is numeric, it contains the time interval frequencies and function uses the first condition part. Then, the client IP is controlled. If this IP does not appear in the system, the usual score returns as zero. Else, all users in the cluster where this client IP is in, are found and for each user, the similarities between the time interval frequencies in the training data and the test data are calculated one by one and these values are added together. Finally, the average similarity is calculated and this value is returned as the usual score.

If the data is not numeric, it means the data contains the required IPs and function uses the second condition part. Then, the client IP is controlled. If this IP does not appear in the system, the usual score returns as zero. Else, all users in the cluster where this client IP is in are found and the vector space model, of the users in this same cluster and the test data for each users, is created together. Then, the similarities, between the test data and training data according to the matrix, which is formed by the vector space model, by the fraction of Euclidean distance formula,

are calculated one by one for each user, and these values are added together. Finally, the average similarity is calculated and this value is returned as the usual score.

```
Function Outlier_Detection(parameters: the array which contains the
clusters, the data which will be tested)
     If the data is numeric Then
           If The client IP in the data which will be tested does
           not appear in the training data warehouse Then Return 0
           Else
                 Find the cluster where the client IP is in
                 For each user in this cluster:
                       Calculate similarities between the test data
                       and training data according to the matrix
                       which contains time interval frequencies by
                       Euclidean distance formula
                       Calculate the sum of these similarity values
                 Calculate average of the similarities between the
                 test data and the training data in this same
                 cluster
                 Return this average number which shows the usual
                 score as percentage
           End If
     Else
           If The client IP in the data which will be tested does
           not appear in the training data warehouse Then Return 0
           Else
                 Find the cluster where the client IP is in
                 Create the vector space model of the users in this
                 same cluster and the test data
                 For each user in this cluster:
                       Calculate weight similarities between the
                       test data and training data according to the
                       matrix, which is formed by the vector space
                       model, by the fraction of Euclidean distance
                       formula
                       Calculate the sum of these similarity values
                 Calculate weight average of the weight
                 similarities between the test data and the
                 training data in this same cluster
                 Return this average number which shows the usual
                 score as percentage
           End If
     End If
```

2.6 Outlier Detection with Different Priority Weight Values and Tests

The same users are located in different clusters according to the requested IPs and time intervals; therefore, testing the requested IPs and time interval data warehouses is done one by one. The most important subject of testing operation is

to decide threshold number. In other words, deciding if a request is outlier or normal for a data must been done according to an optimum threshold number.

At a study of recent date about k-Nearest Neighbor classifier for intrusion detection (Liao & Vemuri, 2002), there are two classes as outlier and normal. While testing, the k-nearest neighbors are found for new data and the average simulations of these neighbors are calculated. According to a certain threshold value and this average number, outlier detection is done. However, in this method, while calculating the average similarities, the weight values for all similarity values of k nearest neighbors are taken is 1; in other words, they have the same priorities. To prevent this situation, weight multipliers must be used while calculating the average according to the similarity values. These weight multipliers are the similarity values in the main matrix between the user who has the new data and the users who have the data in the same cluster where the user who has requested this data is. Thus, the priorities of the users would not be same. In this study, the following formula is used for this aim (Hardy, Littlewood, & Pólya, 1988);

(Sum of the products of Euclidean similarity and weight values) / (Sum of these weight values).

In the study of Liao and Vemuri, there is a certain threshold value (Liao & Vemuri, 2002). In the tests of data warehouses with time intervals, the results with same priorities weight multipliers and a certain threshold value are as very successful as the results with different priorities weight multipliers and different threshold values. However, in the tests of data warehouses with requested IPs, the results with same priorities weight multipliers and a certain threshold value are not as successful as the results with different priorities weight multipliers and different threshold values. Different threshold values are calculated one by one for each test data and according to this value, outlier detection is done. The threshold value in the project is taken as the geometric mean of the similarity values in the new matrix after the vector space model of the users in the same cluster.

In clustering, the similarity matrix in Appendix – C which is for all 86 users in the system is used and optimum threshold and k numbers are found, thus five clusters are created. Then test operations are done over these clusters. In tests, totally 258 known requests (inside domain) and 172 novel requests (out of domain) are used. These requests are tested according to both different threshold values with different priority weight values and average similarities with different priority weight values, and certain 0.5 threshold value (it gives the highest rate for the data warehouse) and average similarities with same priority weight values. The known requests are tested as three parts and the novel requests are tested as two parts.

Table 2.13 Detection Rates of Known Requests as Normal and Anomaly According to Different Threshold Values with Different Priority Weight Values and Average Similarities with Different Priority Weight Values. Also, False Negative and False Positive Rate According to this method.

Tests	**Tot. Kno. Req.**	**Tot. Kno. Ano. Req.**	**Tot. Det. Ano. Req.**	**Ano. Det. Rate (%)**	**Fal. Neg. Rate (%)**	**Tot. Kno. Nor. Req.**	**Tot. Det. Nor. Req.**	**Nor. Det. Rate (%)**	**Fal. Pos. Rate (%)**	**Tot. Det. Rate (%)**
1.	86	75	75	100.00	0.00	11	10	90.91	9.09	98.94
2.	86	66	66	100.00	0.00	20	20	100.00	0.00	100.00
3.	86	83	83	100.00	0.00	03	03	100.00	0.00	100.00
Total	258	224	224	**100.00**	**0.00**	34	33	**97.06**	**2.94**	**99.61**

(Tot. : Total, Kno. : Known, Ano. : Anomaly, Req. : Request, Det. : Detection, Nor. : Normal, Fal. : False, Neg. : Negative, Pos. : Positive)

(Total Detection Rate gives weighted mean according to normal and anomaly detections)

Table 2.14 Detection Rates of Known Requests as Normal and Anomaly According to 0.5 Threshold Value and Average Similarities with Same Priority Weight Values. Also, False Negative and False Positive Rate According to this method.

Tests	**Tot. Kno. Req.**	**Tot. Kno. Ano. Req.**	**Tot. Det. Ano. Req.**	**Ano. Det. Rate (%)**	**Fal. Neg. Rate (%)**	**Tot. Kno. Nor. Req.**	**Tot. Det. Nor. Req.**	**Nor. Det. Rate (%)**	**Fal. Pos. Rate (%)**	**Tot. Det. Rate (%)**
1.	86	75	59	78.67	21.33	11	11	100.00	0.00	81.40
2.	86	66	66	100.00	0.00	20	16	80.00	20.00	95.35
3.	86	83	83	100.00	0.00	03	03	100.00	0.00	100.00
Total	258	224	208	**92.90**	**7.10**	34	30	**88.24**	**11.76**	**92.25**

(Tot. : Total, Kno. : Known, Ano. : Anomaly, Req. : Request, Det. : Detection, Nor. : Normal, Fal. : False, Neg. : Negative, Pos. : Positive)

(Total Detection Rate gives weighted mean according to normal and anomaly detections)

The Table 2.13 shows for detection rates of known requests as normal and anomaly according to the different threshold values with different priority weight values and average similarities with different priority weight values are obtained with total detection rate as 99.61%. Also, the Table 2.14 shows again that very successful results, for detection rates of known requests as normal and anomaly according to 0.5 threshold value and average similarities with same priority weight values, are obtained with total detection rate as 92.25%.

There is not a great difference between two results which come from these two methods at separation of known requests as normal and anomaly; however, there is a great difference between the results of tests of novel requests which come from the first method, which has different threshold values with different priority weight values and average similarities with different priority weight values, and the second method, which has 0.5 threshold value and average similarities with same priority weight values.

If the Table 2.15 for the first method and the Table 2.16 for the second method are compared, it is realized that at anomaly detection, these two methods are successful with results as 87.21% and 91.86%; however, at normal detection, the second method, with 0.5 threshold value and average similarities with same priority weight values, is has a success rate of 37.21%.

The first method is again stabilized at detection rate with 80.23% for normal detection. As a result, the method, which has different threshold values with different priority weight values and average similarities with different priority weight values, is stabilizer with high rate scores at all departments than the second method, which has 0.5 threshold value and average similarities with same priority weight values.

Table 2.15 Detection Rates of Novel Requests as Normal and Anomaly According to Different Threshold Values with Different Priority Weight Values and Average Similarities with Different Priority Weight Values. Also, False Negative and False Positive Rate According to this method.

Tests	Tot. Kno. Req.	Tot. Kno. Ano. Req.	Tot. Det. Ano. Req.	Ano. Det. Rate (%)	Fal. Neg. Rate (%)	Tot. Kno. Nor. Req.	Tot. Det. Nor. Req.	Nor. Det. Rate (%)	Fal. Pos. Rate (%)	Tot. Det. Rate (%)
1.	86	43	35	81.40	21.33	43	34	79.07	0.00	80.23
2.	86	43	40	93.02	0.00	43	35	81.40	20.00	87.21
Total	172	86	75	**87.21**	**7.10**	86	68	**80.23**	**11.76**	**83.72**

(Tot. : Total, Nov. : Novel, Ano. : Anomaly, Req. : Request, Det. : Detection, Nor. : Normal, Fal. : False, Neg. : Negative, Pos. : Positive)

(Total Detection Rate gives weighted mean according to normal and anomaly detections)

Table 2.16 Detection Rates of Novel Requests as Normal and Anomaly According to 0.5 Threshold Value and Average Similarities with Same Priority Weight Values. Also, False Negative and False Positive Rate According to this method.

Tests	Tot. Kno. Req.	Tot. Kno. Ano. Req.	Tot. Det. Ano. Req.	Ano. Det. Rate (%)	Fal. Neg. Rate (%)	Tot. Kno. Nor. Req.	Tot. Det. Nor. Req.	Nor. Det. Rate (%)	Fal. Pos. Rate (%)	Tot. Det. Rate (%)
1.	86	43	39	90.70	9.30	43	14	32.56	67.44	61.63
2.	86	43	40	93.02	6.98	43	18	41.86	58.14	67.44
Total	172	86	79	**91.86**	**8.14**	86	32	**37.21**	**62.79**	**64.53**

(Tot. : Total, Nov. : Novel, Ano. : Anomaly, Req. : Request, Det. : Detection, Nor. : Normal, Fal. : False, Neg. : Negative, Pos. : Positive)

(Total Detection Rate gives weighted mean according to normal and anomaly detections)

In Appendix – A and Appendix – B, the lists of results which give the Table 2.15 and the Table 2.16 is located, because the most blatant difference is shown there. In Appendix – A, bold results shows anomaly; the others are normal for novel requests according to different threshold values with different priority weight values and average similarities with different priority weight values. There are 86 anomaly requests and 86 normal requests; this successful method finds 75 anomaly requests and only 69 normal requests. In Appendix – B, bold results are normal; the others are anomaly for novel requests according to 0.5 threshold value and average similarities with same priority weight values. There are 86 anomaly requests and 86 normal requests; however, this method finds 79 anomaly requests and only 32 normal requests.

In the other statistical results; the total detection rate of known and novel request detection for the first method at anomaly detection is 96.45% (224 + 75 / 224 + 86) and false negative rate is 3.55%; however, the result for the second method at anomaly detection is 92.58% (208 + 79 / 224 + 86) and false negative rate is 7.42%.

The total detection rate of known and novel request detection for the first method at normal detection is 85.00% (33 + 69 / 34 + 86) and false positive rate is 15.00%; however, the result for the second method at normal detection is 51.67% (30 + 32 / 34 + 86) and false positive rate is 48.33%.

Finally, the total detection rate for the first method, at both anomaly and normal detection, is 93.26% (257 + 144 / 258 + 172); however, the result for the second method, at both anomaly and normal detection, is 81.16% (238 + 111 / 258 + 172). The results of false negative and false positive rates at the first method are also more successful than the results at the second method.

At outlier detection with time interval data warehouse, the false positive and false negative rates change between 0.0% and 1.0%. At a recent study “Intrusion detection using text processing techniques with a binary-weighted cosine metric” (Rawat, Gulati, Pujari & Vemuri, 2006), the false positive rates are dissected without false negative rates. These results are between 0.0% and 1.0%.

At outlier detection with requested IPs data warehouse, the false positive and false negative rates change between 0.0% and 15.0%. In “The k nearest neighbor algorithm predicted rehabilitation potential better than current Clinical Assessment Protocol” (Zhu, Chen, Hirdes & Stolee, 2007), the false positive rates are dissected without false negative rates. The false positive rate finds minimum 24% according to the anomaly results with KNN classification method. As a result, outlier detection with the KNN clustering is more successful than KNN classification.

CHAPTER THREE
THE FEATURES OF THE DETECTION SYSTEM WITH INTERFACES

3.1 Detection System Step by Step

The detection system is created at the Microsoft.NET platform and it is written with C# programming language because, the requirements of the detection system are a useful interface, dynamic file operation comments and a systematic code batch and these platform and programming language supply all requirements. When this system starts the training operation, it firstly needs a data collection in a text file and the type of the data collection is not necessary, because the system can separate the types, requested IPs as string or time intervals as numeric, automatically. At background, the all operations are done according to this type orderly. The following Figure 3.1 shows the start screen of this detection system and firstly, it needs a data collection to calculate the similarity matrix by the pushing the `CALCULATE SIMULATION` button.

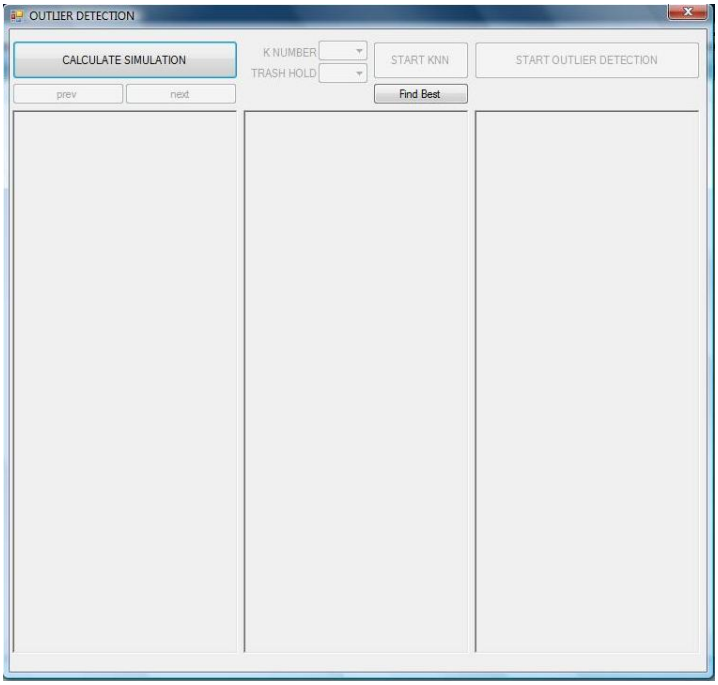

Figure 3.1 The screen shot picture of the start window of the detection system

3.1.1 Calculation of Similarity Matrix

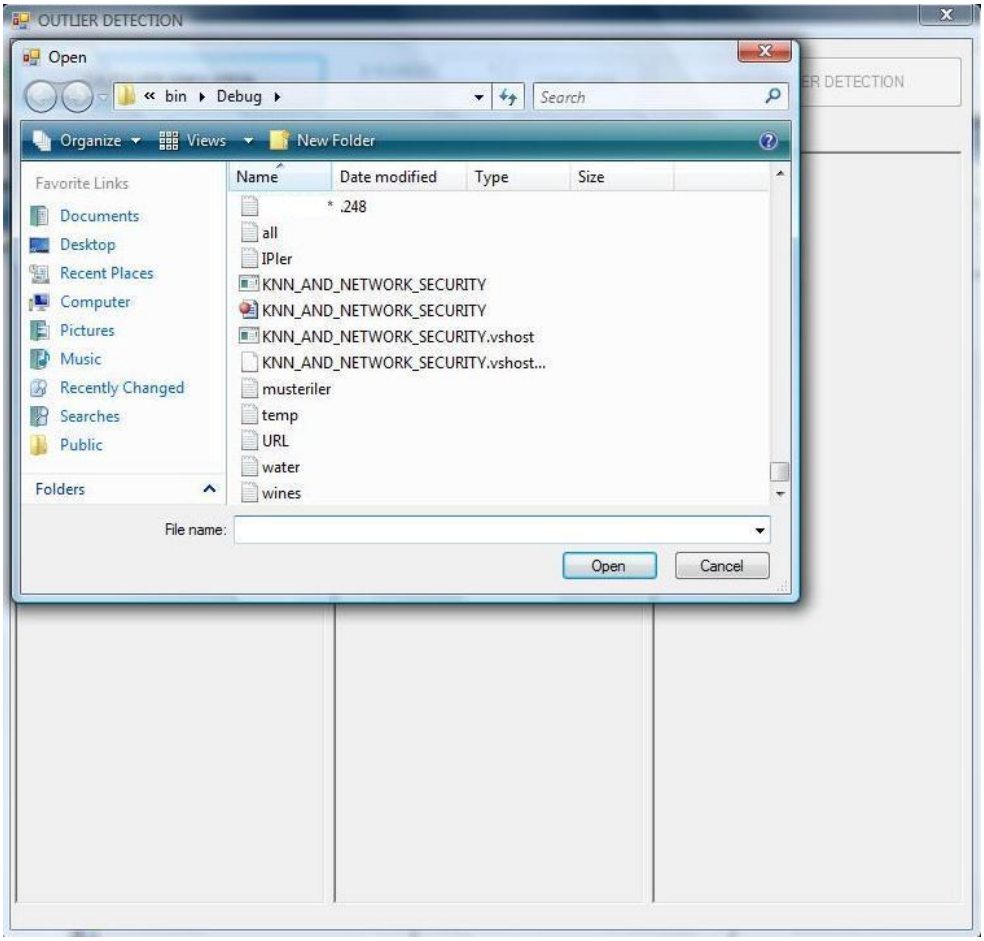

Figure 3.2 The screen shot picture of the file browsing for calculating the simulation matrix

The Figure 3.2 shows the file browsing screen for calculating the simulation matrix after pushing the `CALCULATE SIMULATION` button. After choosing a text file, according to the type of the data in the selected text file, Euclidean formula or only the fraction of the Euclidean formula ($\sum$ (Xi * Yi)) is used to create the simulation matrix which is the data warehouse of the clustering system. The following Figure 3.3 shows the requested IP address data collection in the text file. In this file, each line puts requested IPs for a user and these IP addresses are separated by a space character. The system reads all lines one by one and makes a word analysis. Finally, a dictionary, which uses to help for the simulation matrix, is created like the following Figure 3.4.

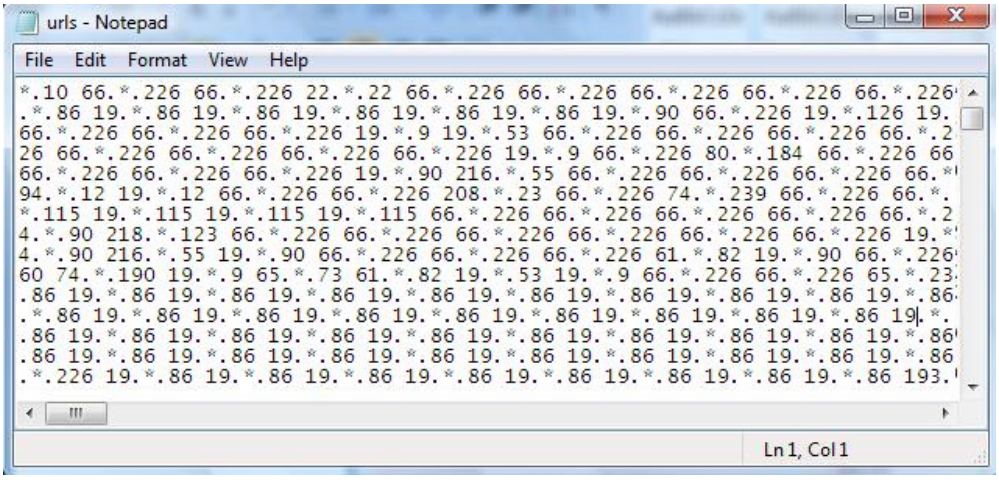

Figure 3.3 The screen shot picture of the data collection of the requested IPs

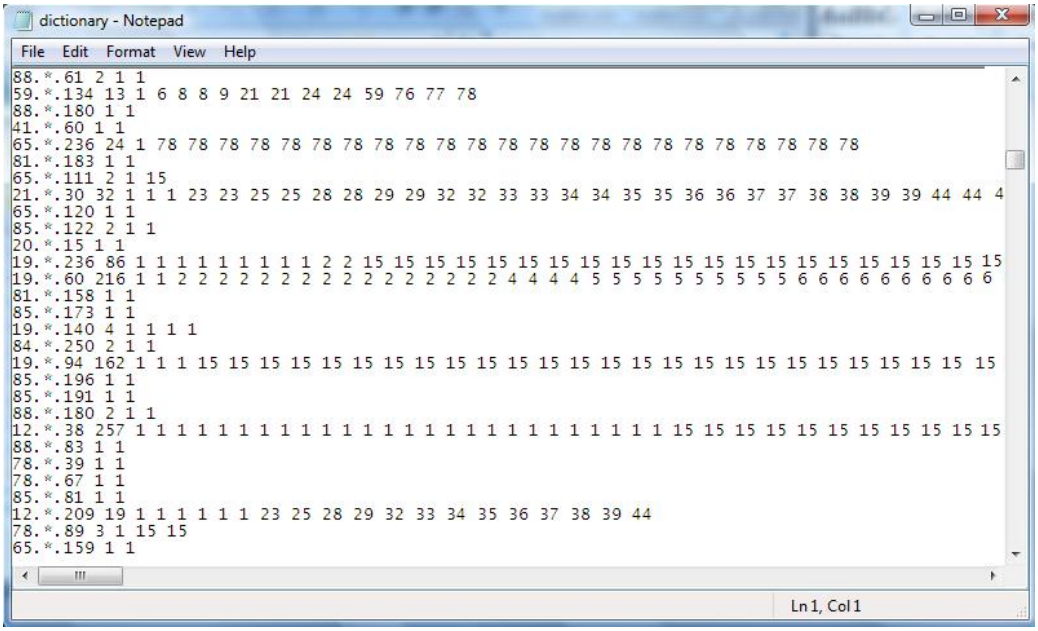

Figure 3.4 The screen shot picture of the created dictionary according to the data collection

This dictionary puts a requested IP on each line and then, the count of this IP address comes as separated by a space character. After this data, the ids of the users who request this IP address orderly, according to their frequencies. Firstly, the vector space model weight values are prepared according to the frequencies in this dictionary, and finally, similarity matrix is created by the fraction of the Euclidean formula ($\sum$ (Xi * Yi)).

The following Figure 3.5 shows a part of the output of the simulation matrix, and the helper `prev. ` and `next` buttons supply passes to similarity data of the other users. This Figure 3.5 also shows other necessary buttons for the clustering operation as k and threshold numbers dropdown lists, and k-nearest neighbor and the button which gives the best clustering result by a statistical analysis, error rate calculations without choosing any k and threshold numbers.

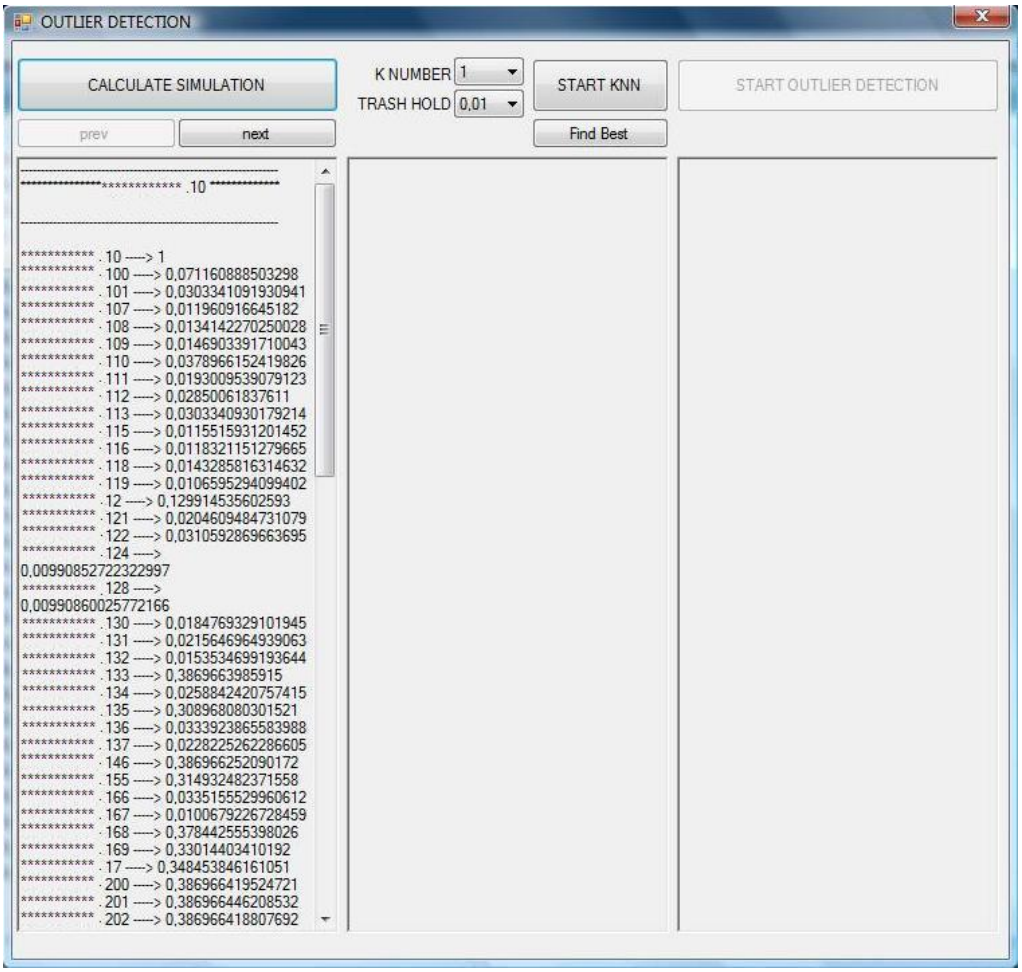

Figure 3.5 The screen shot picture of the detection system after similarity calculation has finished, and it shows a part of the similarity matrix

The following Figure 3.6 shows the situations by pushing the ‘next’ button three times and the three part of the simulation matrix which contains similarity data for three users.

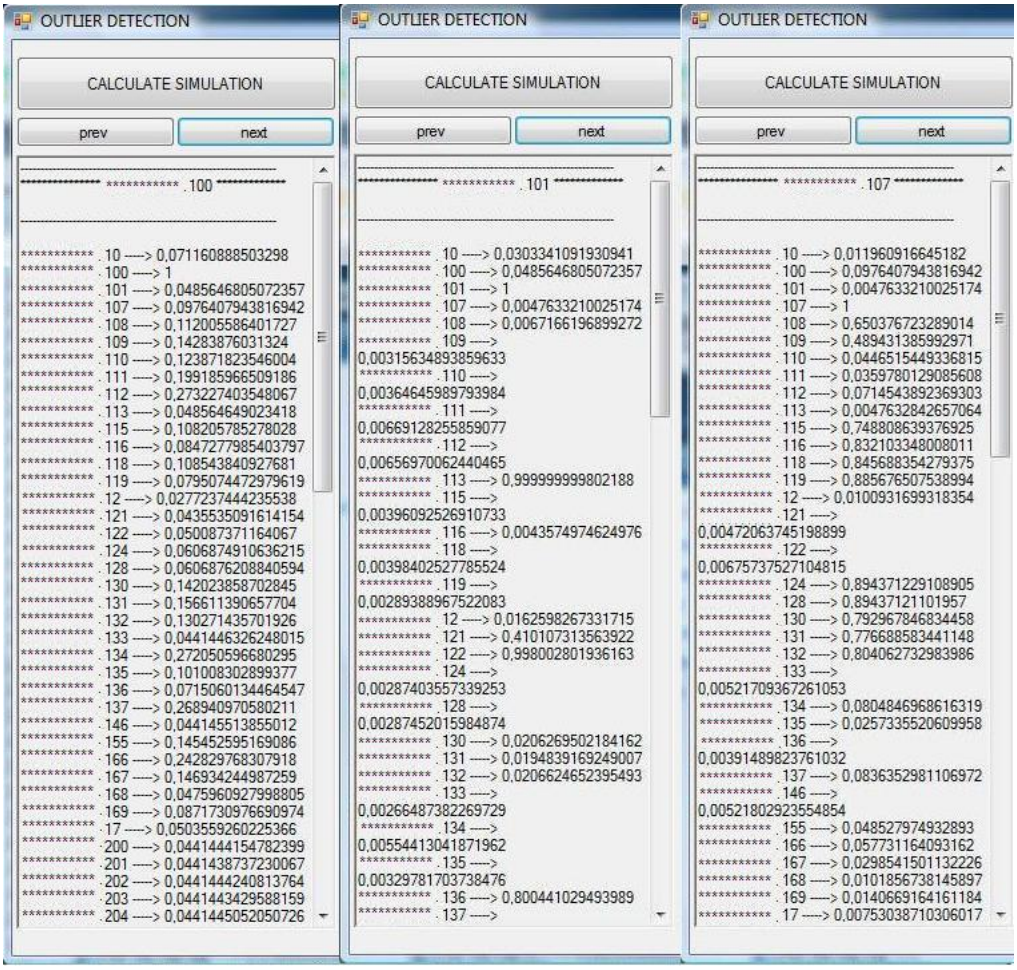

Figure 3.6 The situations by pushing the 'next' button three times

3.1.2 Clustering Operation and Finding Best

The following Figure 3.7 shows a clustering result where k number is 1 and threshold number is 0.01 according to using the simulation matrix which has passed the vector space model. The feature of this clustering result is the basic model of the k- nearest neighbor algorithm, because only one neighbor is controlled with minimum threshold number. According to this result, over ten clusters are created and users can not be located healthy, because with these basic k and threshold parameters, k- nearest neighbor algorithm cannot be caught all common features between the users and the similar characteristic users can be located in the different clusters. For this aim, k number must be increased, threshold number must be increased or both of them must be increased; however, which one and how much?

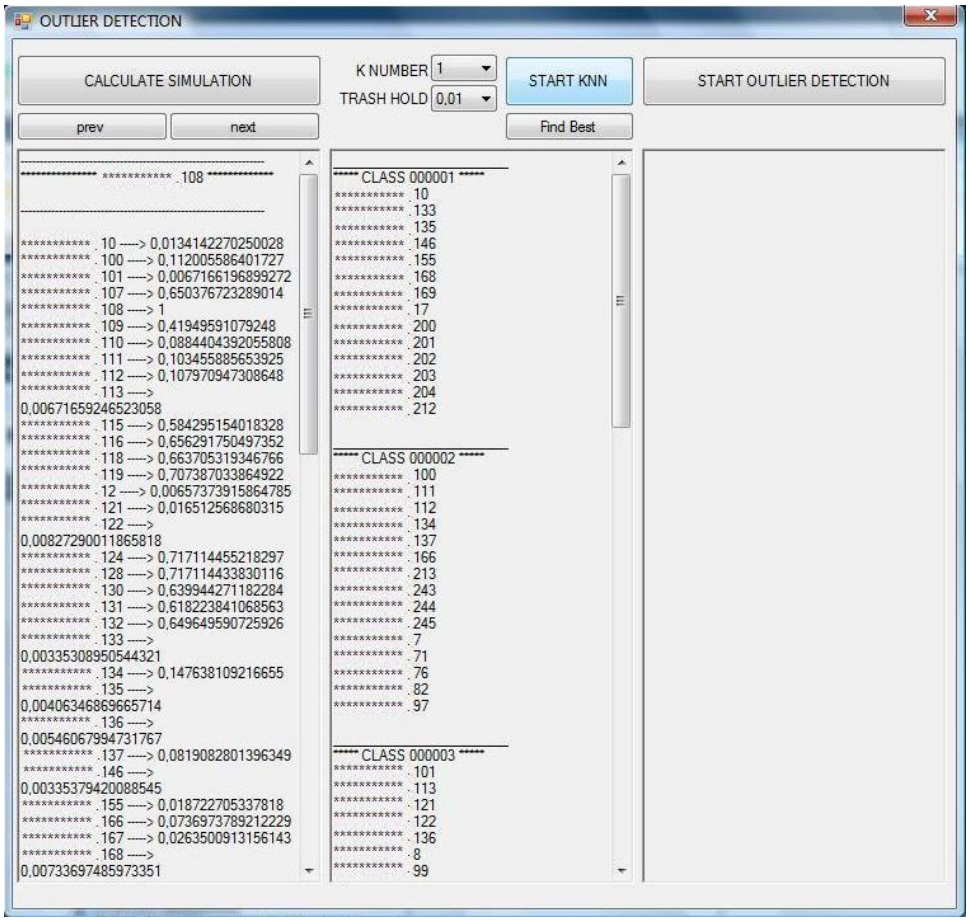

Figure 3.7 The clusters with k number as 1 and threshold number as 0.01

If only k number is increased for the most successful clustering result, it can be thought that a better clustering result would come. However, it is seen that there is again a bad cluster model with only two clusters, because it is not wanted that 86 users are in only two clusters for healthy outlier detection operations. The first cluster has 83 users and the second cluster has only 3 users. It is seen that again there is not a successful separation. Therefore, it can be tested with other ways that only threshold number is increased or both k number and threshold number are increased.

The following Figure 3.8 shows these clusters according to k number as 27 and threshold number as 0.01 again.

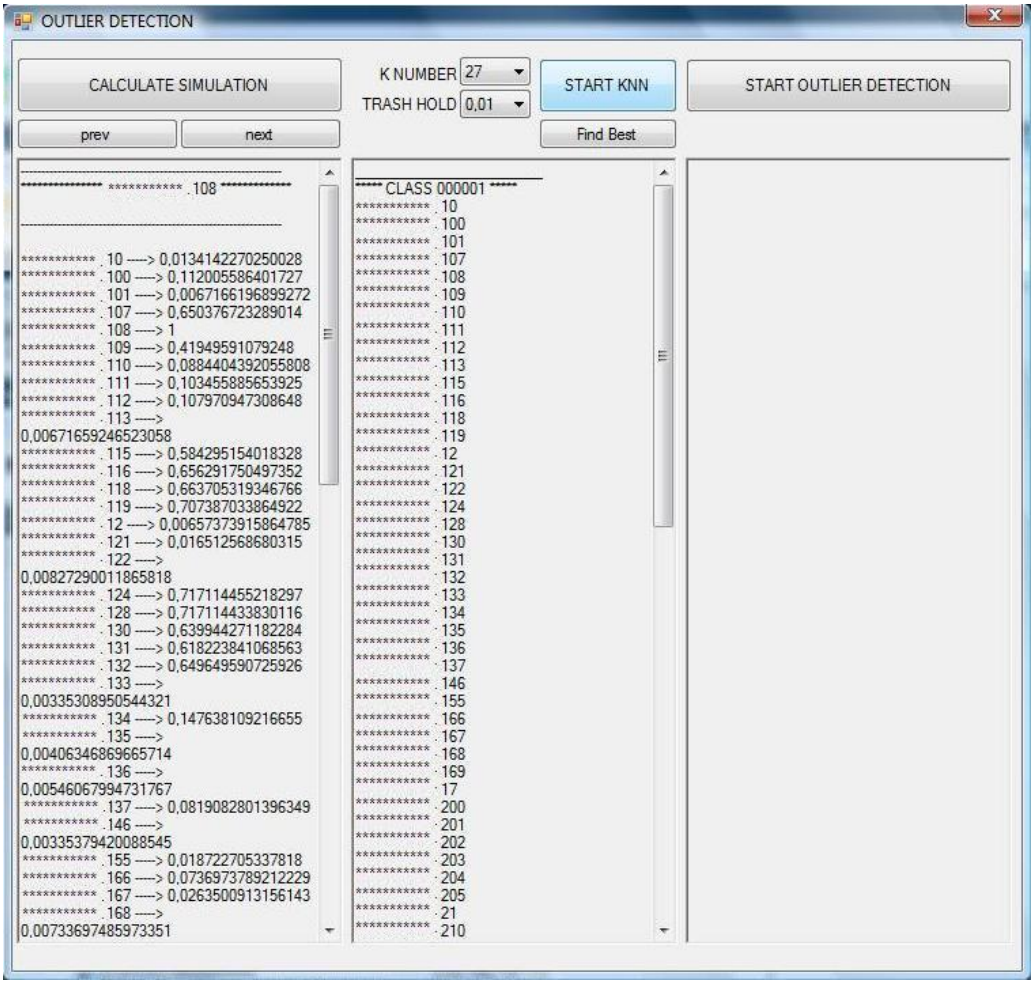

Figure 3.8 The clusters with k number as 27 and threshold number as 0.01

If only threshold number is increased for the most successful clustering result, it can be thought again that a better clustering result would come. However, it is seen that there is again a bad cluster model with over 30 clusters, because it is not wanted that 86 users are in 30 clusters for healthy outlier detection operations. Averagely, there are three users in each cluster. However, the most of these clusters has only one user and it is seen that again there is not a successful separation. Therefore, the last way can be tested as both k number and threshold number are increased.

The following Figure 3.9 shows these clusters according to k number as 1 and threshold number as 0.8 again.

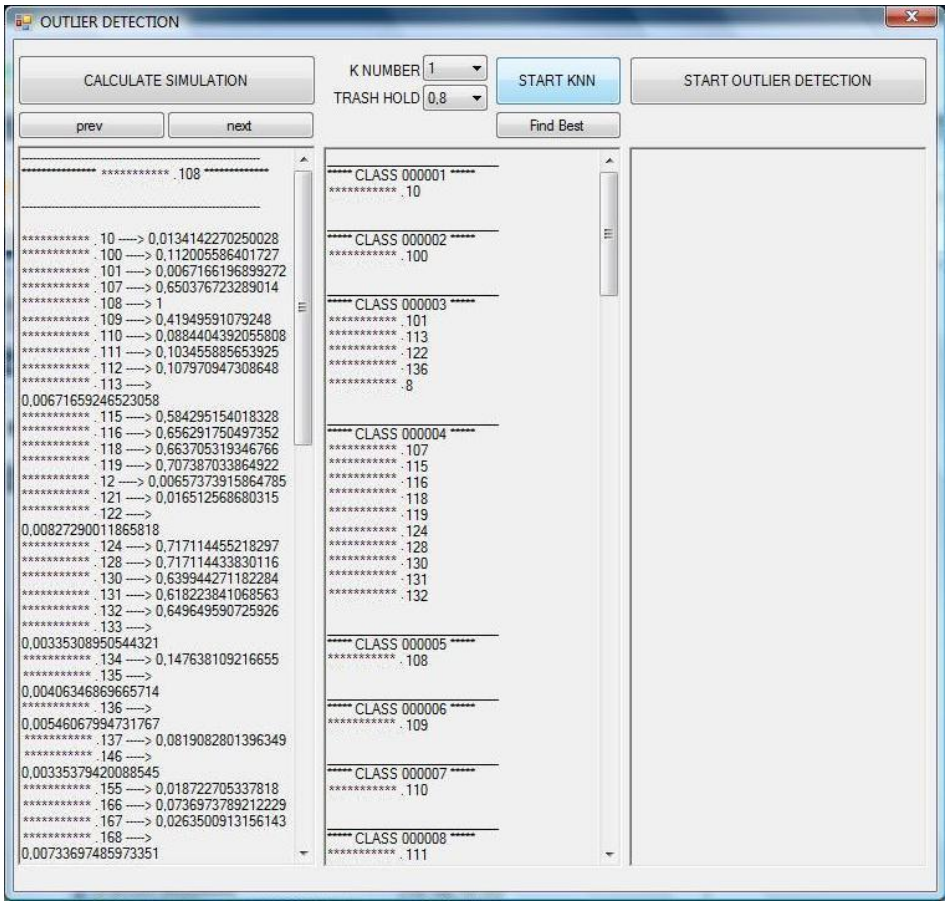

Figure 3.9 The clusters with k number as 1 and threshold number as 0.8

For the most successful clustering model, optimum k and threshold numbers must be caught, because flexible k-nearest neighbor algorithm with optimum k and threshold numbers find the all common features between users and according to them, the clustering model is implemented.

It is calculated that how many k number and threshold number must be increased by using methods in “2.4 Finding Optimum K and Threshold Numbers” in Chapter Two. According to these methods, optimum k number is found as 27 and optimum threshold number is found as 0.05. By these optimum parameters, k-nearest neighbor creates five clusters and the first cluster has 12 users, the second cluster has 42 users, the third cluster has 10 users, the fourth cluster has 19 users and the fifth cluster has 3 users.

The following Figure 3.10 shows these clusters according to optimum k number as 27 and optimum threshold number as 0.05.

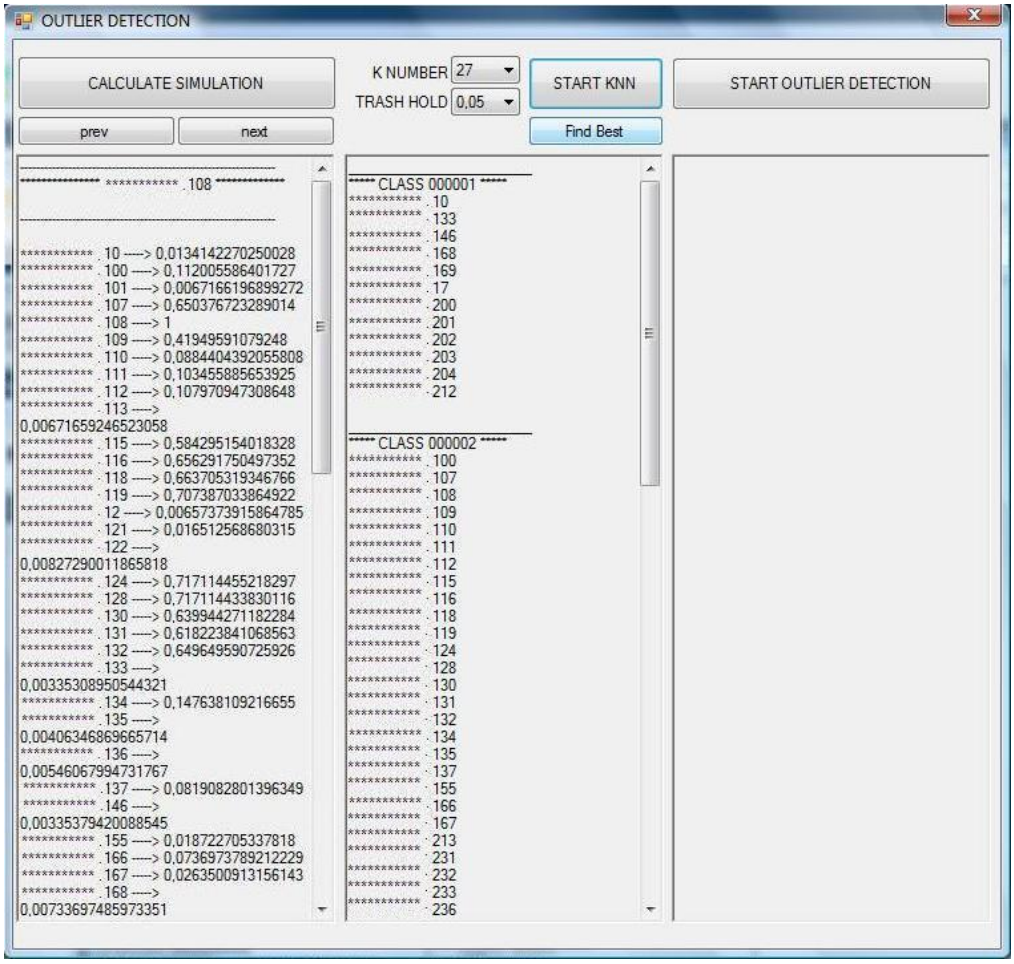

Figure 3.10 The clusters with optimum k number as 27 and optimum threshold number as 0.05

3.1.3 Outlier Detection

After finding the most successful clustering model, training operation is finished and the test operations can be started over this clustering model. For this aim, if 'OUTLIER DETECTION' button is pushed, a file browsing window comes and a text file which contains the test data must be selected. There is not a necessity for data type separation alternatively, because outlier detection part of the system has an automatic separation of the data collection of requested IPs from the data

collection of time interval. Therefore, only, the text file which contains the test requested IPs is selected from the file browsing like the following figure 3.11.

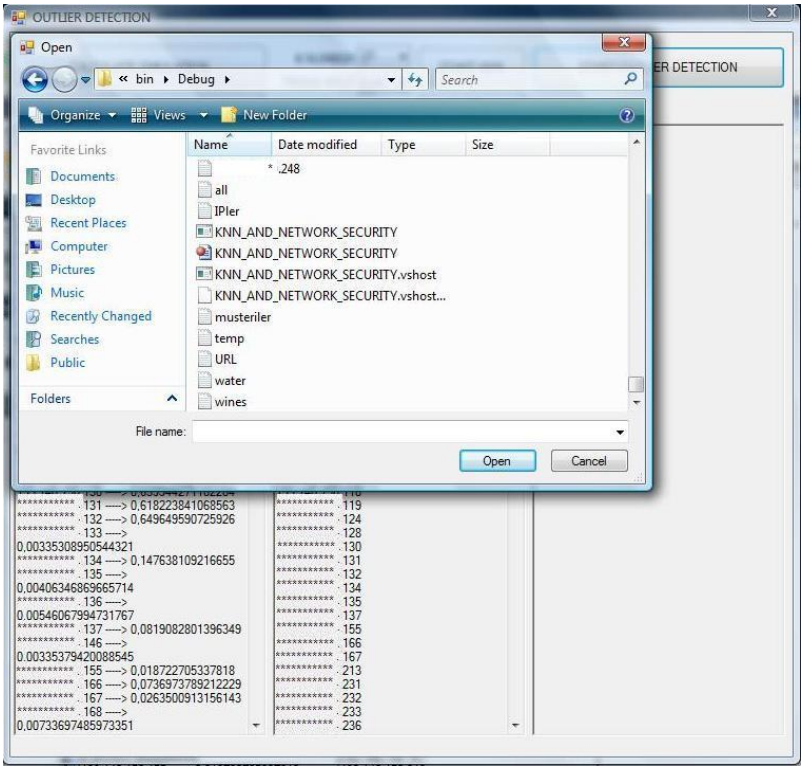

Figure 3.11 The screen shot picture of the file browsing for outlier detection operations

If a text file which contains two test data for a user as one of them is a normal behavior and other one is an anomaly behavior is selected, the following result outputs in the Figure 3.12 are found.

For outlier detection, the methods in "2.6 Outlier Detection with Different Priority Weight Values and Tests" in Chapter Two are used. According to these methods, different thresholds are found for these two test data as 0.0174 and 0.2518. The first test data needs a threshold value which is less than 0.3206 and it has a threshold value as 0.0174; thus, it is found as a normal behavior. The second test data needs a threshold value which is less than 0.1284 threshold value; however, it has a threshold value as 0.2518. Therefore, it is categorized as an anomaly behavior.

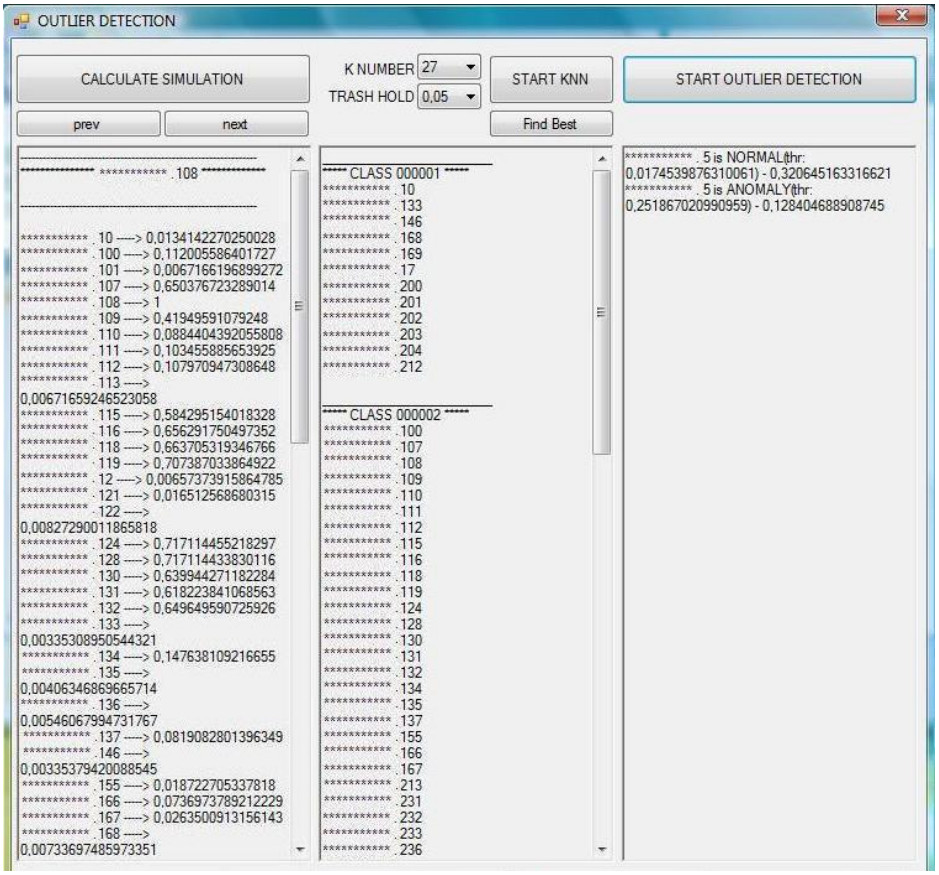

Figure 3.12 The outlier detection results for two test data of a user as one of them is a normal behavior and other one is an anomaly behavior.

3.2 Helper Documents of the Detection System

The detection system needs some helper and result documents to show the system administrator for success degree of the outlier detection operations and the other operations before it. Firstly, after the text file is selected by the system administrator for training operation, the detection system creates directory which has the same name with the training text file to create a batch for all documents.

The following Figure 3.13 shows the directory which has the same name with the training text file of which name is “urls.txt” as “urls”. It is created while simulation matrix is calculating at the beginning of the operations.

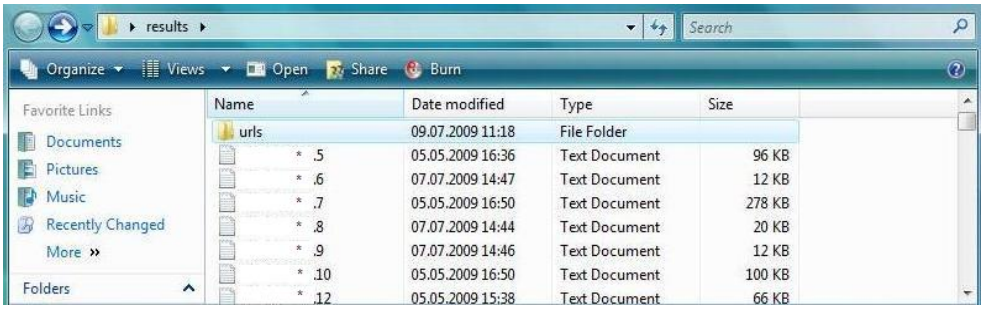

Figure 3.13 The batch directory, of helper and result, of which name is "urls"

The following Figure 3.14 shows the inside of the main batch directory. There are another three directories for clusters, matrices and some statistical results in this directory. Also, there are some files for help; for example, "dictionary.txt" is created and used by the vector space model before calculating the simulation matrix. Other files are used for finding optimum k number and threshold numbers and they put error rates. These files are created while finding the best clustering model operation.

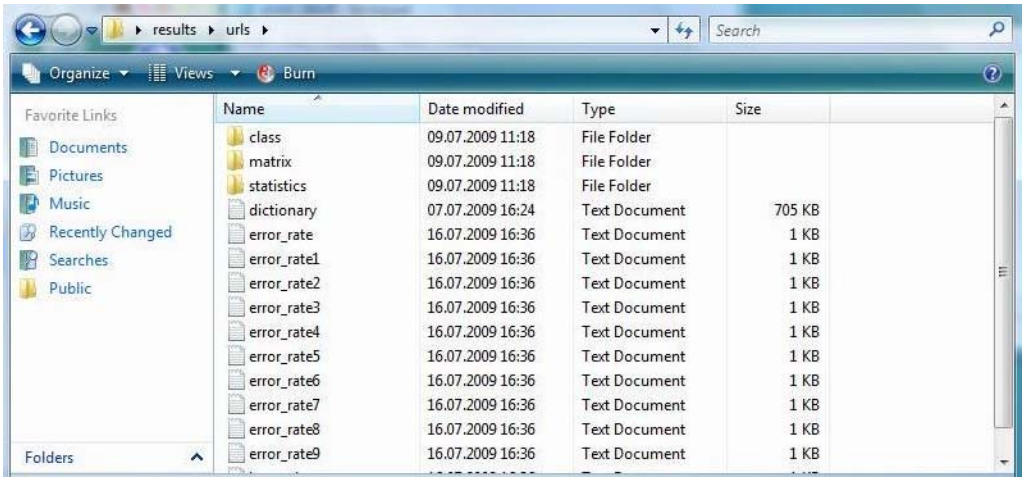

Figure 3.14 The inside of the batch directory

The following Figure 3.15 shows the minimum error rate which is given by optimum k number and optimum threshold number as 0.3517. The optimum threshold is 0.05, because "the error_rate4.txt" has this value. (error_rate.txt for 0.01, error_rate1.txt for 0.02, error_rate2.txt for 0.03, error_rate3.txt for 0.04, error_rate4.txt for 0.05, …). The optimum k number is 27, because the 27th line has

this 0.3517; thus, it can be said that the files are created according to threshold numbers and k number is based on the line number.

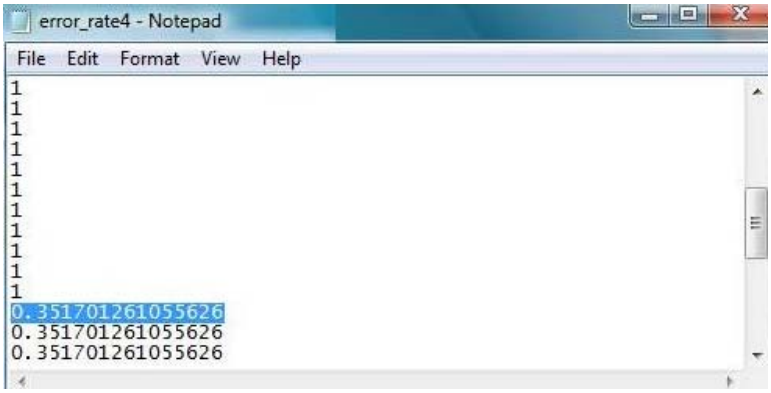

Figure 3.15 The minimum error rate, 0.3517 is on the 27th line in the "error_rate4.txt" file

The following Figure 3.16 shows the inside of the cluster directory in the main batch directory, "urls". This directory puts the all tested cluster combinations by the system administrator. Also, there is the file of which name is "27_0.05.txt", it puts the best cluster model with optimum k number and optimum threshold number.

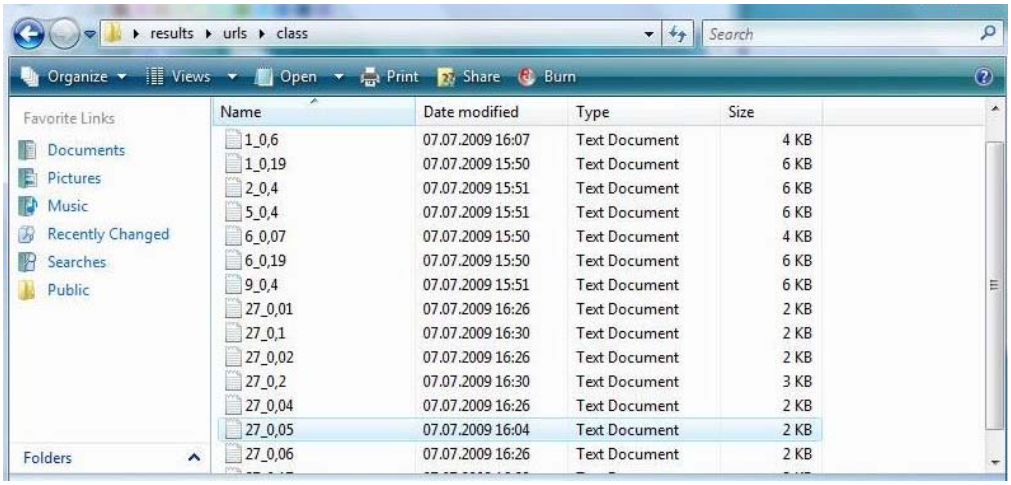

Figure 3.16 The inside of the cluster directory

The following Figure 3.17 shows the users in five clusters at the best cluster model according to optimum k number as 27 and threshold number as 0.05. By these result files, the system does not work for the same requests again and again.

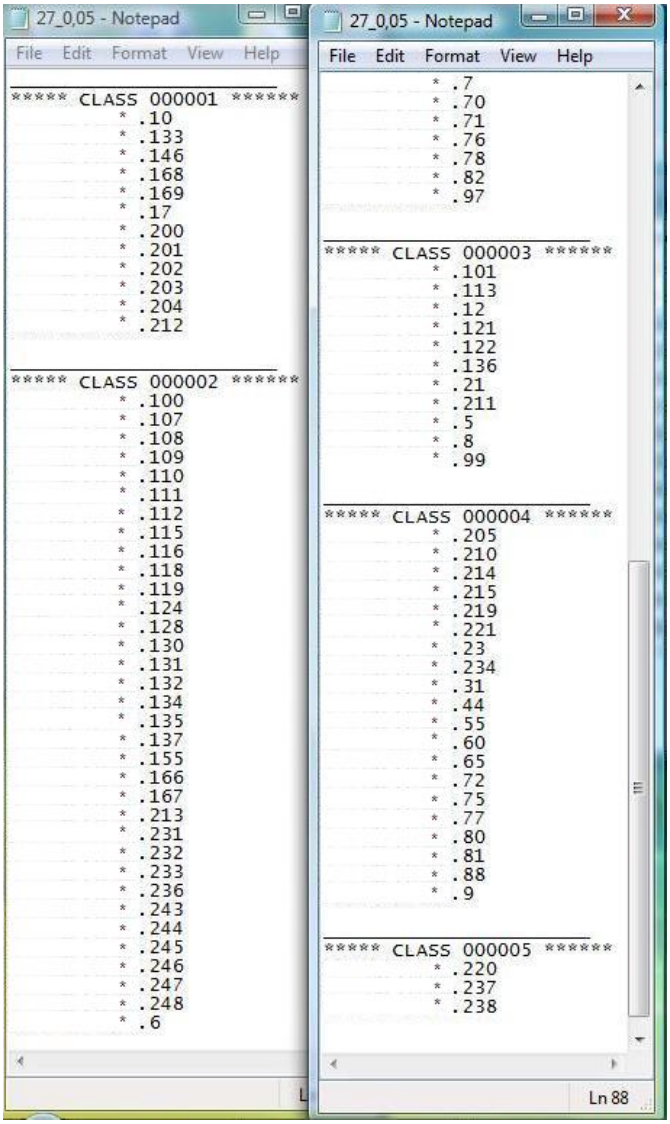

Figure 3.17 The best clustering model for the training data with optimum parameters

Another directory is for matrices in the main batch directory. In this matrices directory, there are text files which contain all similarity values between all users. They are separated by user names of themselves in text files of themselves. They are created one by one when the system administrator pushes the next button not to lose the speed of the detection system.

The following Figure 3.18 shows the list of text files which, all users have one by one, in the matrices directory.

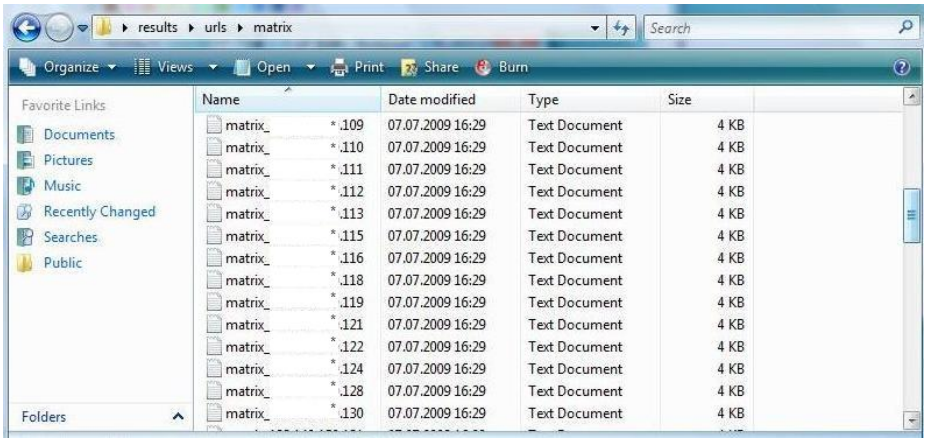

Figure 3.18 The matrices, which have the similarity values between the users, in text files.

The following Figure 3.19 similarity values in “matrix_*.5.txt” file.

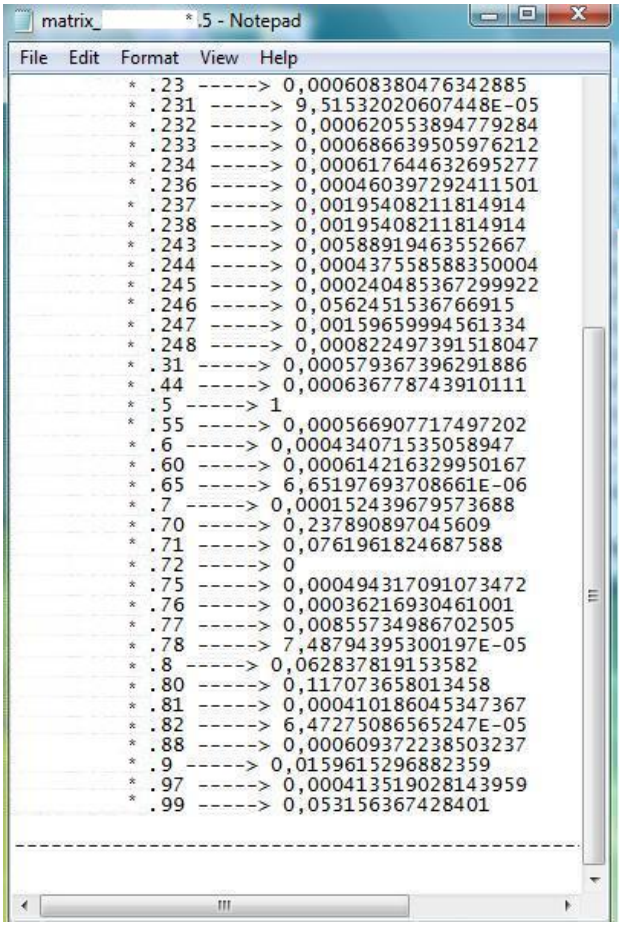

Figure 3.19 The similarity values between user *.5 and the other users

The following Figure 3.20 shows the text files in the statistics results directory. Each text file contains the data about only one clustering model with k number and threshold names.

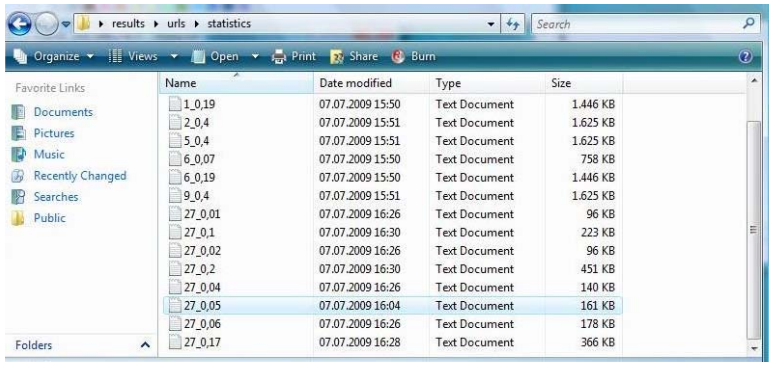

Figure 3.20 The inside of the statistics results directory.

The following text file shows the statistical results for the best clustering model. The first result is about count of users in the clusters, the second result is about minimum and maximum values of each attribute one by one and the third result is about average values of each attribute one by one in the clusters. These results can help the system administrator about the users in the clusters.

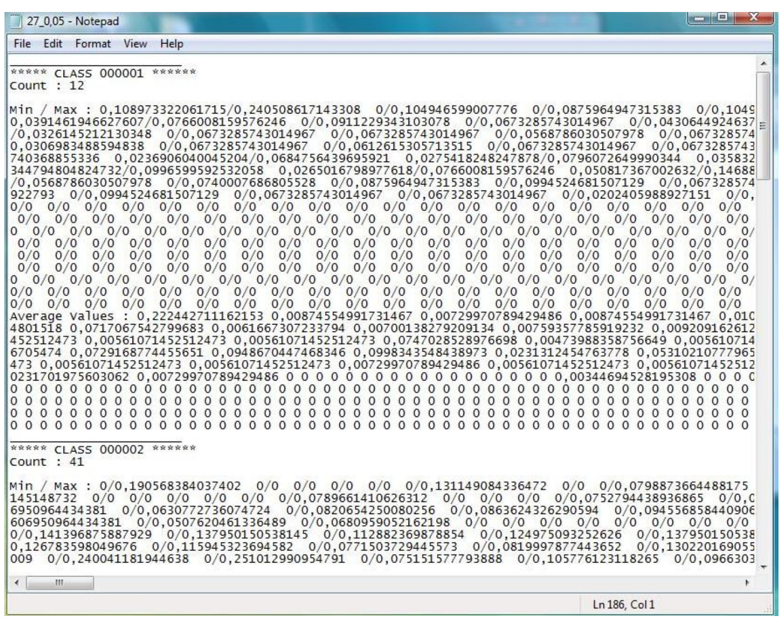

Figure 3.21 The statistical result text of the best clustering model which has k number as 27 and threshold number as 0.05.

CHAPTER FOUR
CONCLUSION

Wireless network structure is open to attacks and unwanted listens. To prevent this negative situation, security of wireless network wins alternative approaches through this study. By the double-sided control mechanism in the system, the security becomes stronger. Since this study has double-sided control mechanism, alternative approaches are able to make rich with addition of new control mechanism and qualities of the security is able to be increased.

By finding optimum parameters of the k-Nearest Neighbor, result of the clustering operation becomes more reliable. By the using vector space model of text mining technique over requested IP addresses, affinities between users are caught and using the matrix after text mining technique supplies more healthy results for clustering.

Finally, at detection cycle it is realized that usage of different threshold values with different priority weight values and average similarities with different priority weight values supplies more successful results for both anomaly and normal detections than usage of certain threshold value and average similarities with same priority weight values.

REFERENCES

Aas, K. & Eikvil, L. (1999). Text categorization: A Survey *Norwegian Computing Center. Report NR* 941.

Bloedorn, E., Christiansen, A.D., Hill, W., Skorupka, C., Talbot, L.M. & Tivel, J. (August 2001). Data mining for network intrusion detection: How to get started. *MITRE Techniqual Report* from http://www.mitre.org/work/tech_papers/tech_papers_01/bloedorn_datamining/bloedorn_datamining.pdf

Evans, E.D. (2008), *Lincoln Laboratory of Massachusetts Institute of Technology* from http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From data mining to knowledge discovery in databases. *An overview, Advances in knowledge discovery and data mining, American Association for Artificial Intelligence, Menlo Park, CA,* 37-54.

Frincke, D. (Ed.) (2002). *Intrusion Detection* (153-154). IOS Press.

Hardy, G. H., Littlewood, J. E., & Pólya, G. (Eds.) (1988). *Inequalities* (2nd ed.), Cambridge University Press

Lazarevic, A., Dokas, P., Ertoz, L., Srivastava, J., Kumar, V. & Tan, P.N. (2002). Data mining for network intrusion detection. *NSF Workshop on Next Generation Data Mining, Baltimore, MD,* 21-30.

Liao, Y. & Vemuri, R. (2002). Use of k-nearest neighbor classifier for intrusion detection. *Computers & Security, vol. 21* (5), 439-448.

Manning, C.D., Raghavan, P. & Schütze, H. (Eds.) (2008). *An introduction to information retrieval* (109-134). Cambridge: Cambridge University Press.

Münz, G., Li S. & Carle G. (2007). Traffic anomaly detection using k-means clustering, *GI/ITG Workshop MMBnet*

Nobuhiro, Y. & Okamoto, S. (1997). An average-case analysis of the k-nearest neighbor classifier for noisy domains. *Proc. 15th Internat. Joint Conf. on Artificial Intelligence, Morgan Kaufmann, San Francisco, CA,* 238-243.

Porras, P. (1993), STAT - A state transition analysis tool for intrusion detection, *University of California at Santa Barbara, Santa Barbara, CA*

Rawat S., Gulati V.P., Pujari A.K. & Vemuri V.R. (2006). Intrusion detection using text processing techniques with a binary-weighted cosine metric. *Journal of Information Assurance and Security*

Stallings, W. (2006). C*ryptography and network security - Fourth Edition* (570 - 574). Prentice Hall

Wei S., Mirkovic J. & Kissel E. (2006). Profiling and clustering internet hosts. *Proceedings of the 2006 International Conference*

Wen-chao, H. & Huan, Z. (2004). A study of clustering-based intrusion detection algorithm. *Postgraduate Journal, vol. 21(3),* 101-105

Zhu M., Chen W., Hirede J.P. & Stolee P. (2007). The k nearest neighbor algorithm predicted rehabilitation potential better than current Clinical Assessment Protocol. *J Clin Epidemiol 2007; 60: 1015-1021*

APPENDIX – A – SUCCESSFUL DETECTION RESULTS

thr. : Threshold

*.5 is NORMAL(thr: 0.0174539876310061) - 0.320645163316621
***.5 is ANOMALY(thr: 0.251867020990959) - 0.128404688908745**
*.6 is NORMAL(thr: 0.178087306634871) - 0.313216712316617
***.6 is ANOMALY(thr: 0.154167328328825) - 0.0934315659242721**
*.7 is NORMAL(thr: 0.0951219693095421) - 0.2708668018256
***.7 is NORMAL(thr: 0.0685634539218164) - 0.129409382997259**
*.8 is NORMAL(thr: 0.301801409497693) - 0.66129941437706
***.8 is ANOMALY(thr: 0.354862557919018) - 0.246677147263121**
*.9 is NORMAL(thr: 0.315005309929128) - 0.681293339430643
***.9 is ANOMALY(thr: 0.0440402169816955) - 0.0197033665301373**
*.10 is NORMAL(thr: 0.613925469118261) - 0.921424366565332
***.10 is ANOMALY(thr: 0.366472398365181) - 0.336366292495017**
*.12 is NORMAL(thr: 0.0814901900579828) - 0.208648304946105
***.12 is NORMAL(thr: 0.0927479406296448) - 0.115874317769778**
*.17 is NORMAL(thr: 0.907742325894268) - 0.947044816147944
***.17 is ANOMALY(thr: 0.702185166057215) - 0.486744477654964**
*.21 is NORMAL(thr: 0.246794252986409) - 0.432947743983527
***.21 is NORMAL(thr: 0.294035312567355) - 0.394258625859193**
*.23 is ANOMALY(thr: 0.608637814115545) - 0.506461225749875
***.23 is ANOMALY(thr: 0.0811959375285695) - 0.0081280608293042**
*.31 is ANOMALY(thr: 0.518474440079229) - 0.367337671648685
***.31 is ANOMALY(thr: 0.0673515927632015) - 0.0200339814155767**
*.44 is NORMAL(thr: 0.681465218559858) - 0.684864405615794
***.44 is ANOMALY(thr: 0.120556044613524) - 0.0212571402359513**
*.55 is ANOMALY(thr: 0.634214339161199) - 0.513688388724436
***.55 is ANOMALY(thr: 0.101554055507189) - 0.0188298831016237**
*.60 is NORMAL(thr: 0.589419262349225) - 0.758547891220146
***.60 is ANOMALY(thr: 0.0825847640913439) - 0.0178574307176566**
*.65 is NORMAL(thr: 0.633177624670738) - 0.751353226159711
***.65 is ANOMALY(thr: 0.0232991690779179) - 0.00190230635122121**
*.70 is NORMAL(thr: 0.214702328268745) - 0.296053187863501
***.70 is ANOMALY(thr: 0.188317670611577) - 0.169943489205068**
*.71 is NORMAL(thr: 0.172409188596559) - 0.293314183382469
***.71 is NORMAL(thr: 0.127777548235569) - 0.152604204906604**

*.72 is ANOMALY(thr: 0.629961508351832) - 0.355070518945137

***.72 is ANOMALY(thr: 0.624915775225511) - 0.119756893129721**

*.75 is NORMAL(thr: 0.384030561233623) - 0.669467959885096

***.75 is ANOMALY(thr: 0.0691826036042367) - 0.0568367130219497**

*.76 is NORMAL(thr: 0.254709296189228) - 0.270171099028209

***.76 is ANOMALY(thr: 0.220168146778566) - 0.118001589579742**

*.77 is NORMAL(thr: 0.160189215392312) - 0.532436689203186

***.77 is ANOMALY(thr: 0.163450363023811) - 0.139016504261692**

*.78 is NORMAL(thr: 0.196270743681552) - 0.324634964359118

***.78 is ANOMALY(thr: 0.166163270035487) - 0.144747856015689**

*.80 is NORMAL(thr: 0.49986127893108) - 0.735644812107502

***.80 is NORMAL(thr: 0.0164490977429136) - 0.036164412240085**

*.81 is NORMAL(thr: 0.38110987489299) - 0.754120819270207

***.81 is ANOMALY(thr: 0.0431630847373719) - 0.0299909729454826**

*.82 is NORMAL(thr: 0.236505048645253) - 0.375161257406807

***.82 is ANOMALY(thr: 0.198652010273107) - 0.143311798264038**

*.88 is NORMAL(thr: 0.650620710546959) - 0.735644386688435

***.88 is ANOMALY(thr: 0.0992360712795061) - 0.0170786700328457**

*.97 is ANOMALY(thr: 0.214903872704277) - 0.140851541223845

***.97 is ANOMALY(thr: 0.188360359444595) - 0.114321754531346**

*.100 is NORMAL(thr: 0.232687376000712) - 0.346563629922971

***.100 is ANOMALY(thr: 0.210400844584999) - 0.122291780112049**

*.101 is NORMAL(thr: 0.424500418096354) - 0.768694650250286

***.101 is ANOMALY(thr: 0.429106709982328) - 0.311238832071129**

*.107 is NORMAL(thr: 0.264854693332223) - 0.284343523718588

***.107 is ANOMALY(thr: 0.240643666936126) - 0.0987677750820094**

*.108 is NORMAL(thr: 0.222971860786154) - 0.508581150852146

***.108 is ANOMALY(thr: 0.201447486747356) - 0.0958911054034624**

*.109 is NORMAL(thr: 0.180649890488879) - 0.291600036421428

***.109 is ANOMALY(thr: 0.164418327938933) - 0.0988741144855837**

*.110 is NORMAL(thr: 0.142756494815922) - 0.329925218904045

***.110 is ANOMALY(thr: 0.14344156063551) - 0.1176936886036**

*.111 is NORMAL(thr: 0.147667662392176) - 0.312951397030809

***.111 is ANOMALY(thr: 0.0531711908369291) - 0.0158336534601314**

*.112 is NORMAL(thr: 0.254784258493604) - 0.38044446038306

***.112 is ANOMALY(thr: 0.188945099461533) - 0.0639967255696205**

*.113 is ANOMALY(thr: 0.431989056899738) - 0.298021727092472

***.113 is ANOMALY(thr: 0.474065438414675) - 0.340017964403189**

*.115 is NORMAL(thr: 0.248010202971948) - 0.51577083371107

***.115 is ANOMALY(thr: 0.22027392206506) - 0.153548308828189**

*.116 is NORMAL(thr: 0.273474418015094) - 0.284583606178349

***.116 is ANOMALY(thr: 0.246973108279018) - 0.157437095069628**

*.118 is NORMAL(thr: 0.295098358317359) - 0.508168421933403

***.118 is ANOMALY(thr: 0.265339508584952) - 0.101373282561061**

*.119 is ANOMALY(thr: 0.288641666759101) - 0.127679655554813

***.119 is ANOMALY(thr: 0.260014044708767) - 0.100328614620453**

*.121 is NORMAL(thr: 0.152135330733784) - 0.497030540222516

***.121 is NORMAL(thr: 0.205744933036635) - 0.306593514962913**

*.122 is NORMAL(thr: 0.229012899029502) - 0.291005628812522

***.122 is ANOMALY(thr: 0.470311405637917) - 0.304724946165324**

*.124 is NORMAL(thr: 0.281960567695279) - 0.496125506416575

***.124 is ANOMALY(thr: 0.255017277976781) - 0.101142108309284**

*.128 is NORMAL(thr: 0.284645434986167) - 0.494694949799823

***.128 is ANOMALY(thr: 0.257984859676594) - 0.103207002638787**

*.130 is NORMAL(thr: 0.280355850487769) - 0.29156390397268

***.130 is ANOMALY(thr: 0.254369136239089) - 0.100341778277399**

*.131 is ANOMALY(thr: 0.281923976743472) - 0.201114053199308

***.131 is ANOMALY(thr: 0.256339574024806) - 0.100452354247166**

*.132 is ANOMALY(thr: 0.260501199627124) - 0.0983317542116922

***.132 is ANOMALY(thr: 0.260427859287834) - 0.100860042750309**

*.133 is NORMAL(thr: 0.936143087773431) - 0.953602283515172

***.133 is ANOMALY(thr: 0.838140803522518) - 0.475933552358016**

*.134 is NORMAL(thr: 0.221070201520423) - 0.398452527558239

***.134 is ANOMALY(thr: 0.19829648657914) - 0.117293329553321**

*.135 is NORMAL(thr: 0.130179727365923) - 0.319435977655002

***.135 is NORMAL(thr: 0.120677176697595) - 0.311504268390445**

*.136 is ANOMALY(thr: 0.0537234019331723) - 0.0247953843258046

***.136 is ANOMALY(thr: 0.351289225902095) - 0.145611967182981**

*.137 is NORMAL(thr: 0.261508935824303) - 0.289565768179563

***.137 is ANOMALY(thr: 0.223259802271019) - 0.129147802218908**

*.146 is NORMAL(thr: 0.942450461670521) - 0.959973540286527

***.146 is ANOMALY(thr: 0.757255659466323) - 0.737163118349255**

*.155 is NORMAL(thr: 0.123915782196917) - 0.279586155552114

***.155 is NORMAL(thr: 0.114334386842768) - 0.33341089218547**

*.166 is NORMAL(thr: 0.269428652283968) - 0.321768752775081

***.166 is NORMAL(thr: 0.191902581829393) - 0.358422067280698**

*.167 is NORMAL(thr: 0.251332329232396) - 0.288707589285203

***.167 is ANOMALY(thr: 0.22793614797542) - 0.125724873587602**

*.168 is NORMAL(thr: 0.937673586398542) - 0.95612400305492

***.168 is ANOMALY(thr: 0.690456603675436) - 0.333629294094135**

*.169 is NORMAL(thr: 0.860598261191014) - 0.956549784982519

***.169 is ANOMALY(thr: 0.614806801036321) - 0.471120573646276**

*.200 is NORMAL(thr: 0.94328537910987) - 0.960056718189338

***.200 is ANOMALY(thr: 0.838335948587577) - 0.475846597171483**

*.201 is NORMAL(thr: 0.942142379521384) - 0.958755225438353

***.201 is ANOMALY(thr: 0.831855973907671) - 0.460759347342136**

*.202 is NORMAL(thr: 0.938917059746662) - 0.956589924003249

***.202 is NORMAL(thr: 0.7185530279618) - 0.88423332972837**

*.203 is NORMAL(thr: 0.938902549323824) - 0.956470683071924

***.203 is ANOMALY(thr: 0.819911585927899) - 0.580573573532509**

*.204 is NORMAL(thr: 0.923604427988136) - 0.937695261070639

***.204 is ANOMALY(thr: 0.769436067970379) - 0.305893040706952**

*.205 is NORMAL(thr: 0.633857396187521) - 0.753891989304637

***.205 is ANOMALY(thr: 0.0234765796589782) - 0.0153019349786722**

*.210 is NORMAL(thr: 0.632614745562523) - 0.785338742062525

***.210 is ANOMALY(thr: 0.0300421651906819) - 0.0263304161471247**

*.211 is NORMAL(thr: 0.068798108358131) - 0.282726683339918

***.211 is ANOMALY(thr: 0.222954120309439) - 0.202682481160767**

*.212 is NORMAL(thr: 0.864690731184075) - 0.94715521820714

***.212 is ANOMALY(thr: 0.68978080347525) - 0.534047942747964**

*.213 is NORMAL(thr: 0.228737863454696) - 0.351952051995535

***.213 is ANOMALY(thr: 0.187961041192371) - 0.125840138507866**

*.214 is NORMAL(thr: 0.676148461626309) - 0.815363474176345

***.214 is ANOMALY(thr: 0.115062520882393) - 0.0139455170845341**

*.215 is NORMAL(thr: 0.42615095965873) - 0.753944362300561

***.215 is ANOMALY(thr: 0.108123484208427) - 0.0595037869808585**

*.219 is NORMAL(thr: 0.675978549489025) - 0.813047249332665

***.219 is ANOMALY(thr: 0.114315355394946) - 0.0257230955347923**

*.220 is ANOMALY(thr: 1) - 0.976712724602781

***.220 is ANOMALY(thr: 1) - 0.530874539033495**

*.221 is NORMAL(thr: 0.675464249522484) - 0.803949136215031

***.221 is ANOMALY(thr: 0.114414964473389) - 0.0267950411808146**

*.231 is ANOMALY(thr: 0.2360907460935) - 0.198781701932882

***.231 is ANOMALY(thr: 0.234048035713376) - 0.118810988253882**

*.232 is ANOMALY(thr: 0.28922258210402 9) - 0.264945030184761

***.232 is ANOMALY(thr: 0.243892445508138) - 0.119221492080451**

*.233 is NORMAL(thr: 0.245219423739764) - 0.392234831051345

***.233 is ANOMALY(thr: 0.100395837661135) - 0.0497636027440996**

*.234 is NORMAL(thr: 0.675490122312838) - 0.803953887662313

***.234 is ANOMALY(thr: 0.111797494150226) - 0.0299916187028773**

*.236 is NORMAL(thr: 0.195696489811131) - 0.331560804230888

***.236 is ANOMALY(thr: 0.0339848291251399) - 0.0104924513195296**

*.237 is ANOMALY(thr: 1) - 0.68857872363584

***.237 is ANOMALY(thr: 1) - 0.537669556388652**

*.238 is ANOMALY(thr: 1) - 0.62114249671771

***.238 is ANOMALY(thr: 1) - 0.504692784227788**

*.243 is NORMAL(thr: 0.187032702204068) - 0.270578400338606

***.243 is ANOMALY(thr: 0.141746013900572) - 0.104815534852924**

*.244 is ANOMALY(thr: 0.220484120870532) - 0.195762010998394

***.244 is ANOMALY(thr: 0.181825843536497) - 0.0746914858381315**

*.245 is NORMAL(thr: 0.152771942367934) - 0.322997890774658

***.245 is NORMAL(thr: 0.116703311965483) - 0.142406153554754**

*.246 is NORMAL(thr: 0.269962780593554) - 0.383230277414616

***.246 is ANOMALY(thr: 0.180417579342957) - 0.137169133644374**

*.247 is NORMAL(thr: 0.365979044550393) - 0.394125342667213

***.247 is ANOMALY(thr: 0.317038417864988) - 0.119865957707938**

*.248 is ANOMALY(thr: 0.312201456488815) - 0.153477742121749

***.248 is ANOMALY(thr: 0.281258235495905) - 0.171399221143072**

APPENDIX – B – UNSUCCESSFUL DETECTION RESULTS

thr. : Threshold

***.5 is ANOMALY(thr: 0.5) - 0.171470647379749**
*.5 is ANOMALY(thr: 0.5) - 0.213101209801905
***.6 is ANOMALY(thr: 0.5) - 0.305062498605695**
*.6 is ANOMALY(thr: 0.5) - 0.0959895286504971
***.7 is ANOMALY(thr: 0.5) - 0.289926084904487**
*.7 is ANOMALY(thr: 0.5) - 0.136902724005225
***.8 is ANOMALY(thr: 0.5) - 0.46129941437706**
*.8 is ANOMALY(thr: 0.5) - 0.246677147263121
***.9 is NORMAL(thr: 0.5) - 0.683868608721973**
*.9 is ANOMALY(thr: 0.5) - 0.0123259763389757
***.10 is NORMAL(thr: 0.5) - 0.921424366565332**
*.10 is ANOMALY(thr: 0.5) - 0.336366292495017
***.12 is ANOMALY(thr: 0.5) - 0.384104092622285**
*.12 is ANOMALY(thr: 0.5) - 0.166261189816057
***.17 is NORMAL(thr: 0.5) - 0.933802475872431**
*.17 is ANOMALY(thr: 0.5) - 0.468019970945066
***.21 is ANOMALY(thr: 0.5) - 0.457067255669553**
*.21 is ANOMALY(thr: 0.5) - 0.387015889120814
***.23 is ANOMALY(thr: 0.5) - 0.492683914117836**
*.23 is ANOMALY(thr: 0.5) - 0.0173440234518752
***.31 is ANOMALY(thr: 0.5) - 0.35216418960316**
*.31 is ANOMALY(thr: 0.5) - 0.0187891239703106
***.44 is NORMAL(thr: 0.5) - 0.645098554669689**
*.44 is ANOMALY(thr: 0.5) - 0.0290199440238984
***.55 is ANOMALY(thr: 0.5) - 0.473888532381814**
*.55 is ANOMALY(thr: 0.5) - 0.0095136446564558
***.60 is NORMAL(thr: 0.5) - 0.683296788395238**
*.60 is ANOMALY(thr: 0.5) - 0.0195785953042634
***.65 is NORMAL(thr: 0.5) - 0.680785856582601**
*.65 is ANOMALY(thr: 0.5) - 0.00474646304265661
***.70 is ANOMALY(thr: 0.5) - 0.276071703398169**
*.70 is ANOMALY(thr: 0.5) - 0.157405750640582
***.71 is ANOMALY(thr: 0.5) - 0.276926334232575**
*.71 is ANOMALY(thr: 0.5) - 0.155183888165214

***.72 is ANOMALY(thr: 0.5) - 0.341160553063658**

*.72 is ANOMALY(thr: 0.5) - 0.113991383057866

***.75 is NORMAL(thr: 0.5) - 0.644394975611165**

*.75 is ANOMALY(thr: 0.5) - 0.0306637193496548

***.76 is ANOMALY(thr: 0.5) - 0.247056941896036**

*.76 is ANOMALY(thr: 0.5) - 0.110995825326706

***.77 is NORMAL(thr: 0.5) - 0.644471261811096**

*.77 is ANOMALY(thr: 0.5) - 0.104513974230519

***.78 is ANOMALY(thr: 0.5) - 0.335210427414541**

*.78 is ANOMALY(thr: 0.5) - 0.134931016342477

***.80 is NORMAL(thr: 0.5) - 0.693138575284357**

*.80 is ANOMALY(thr: 0.5) - 0.0195679517558772

***.81 is NORMAL(thr: 0.5) - 0.708064475895844**

*.81 is ANOMALY(thr: 0.5) - 0.047578424905918

***.82 is ANOMALY(thr: 0.5) - 0.253845482925084**

*.82 is ANOMALY(thr: 0.5) - 0.151315775836677

***.88 is NORMAL(thr: 0.5) - 0.663943384496593**

*.88 is ANOMALY(thr: 0.5) - 0.0203984144126759

***.97 is ANOMALY(thr: 0.5) - 0.149449213624786**

*.97 is ANOMALY(thr: 0.5) - 0.125511643527819

***.100 is ANOMALY(thr: 0.5) - 0.357743544564726**

*.100 is ANOMALY(thr: 0.5) - 0.130101131264834

***.101 is NORMAL(thr: 0.5) - 0.560018253234603**

*.101 is ANOMALY(thr: 0.5) - 0.228809549074692

***.107 is ANOMALY(thr: 0.5) - 0.258830460834516**

*.107 is ANOMALY(thr: 0.5) - 0.1281463662957

***.108 is ANOMALY(thr: 0.5) - 0.350900213667413**

*.108 is ANOMALY(thr: 0.5) - 0.127884063790322

***.109 is ANOMALY(thr: 0.5) - 0.30648118088087**

*.109 is ANOMALY(thr: 0.5) - 0.128788002079548

***.110 is ANOMALY(thr: 0.5) - 0.350387126312413**

*.110 is ANOMALY(thr: 0.5) - 0.138216550248681

***.111 is ANOMALY(thr: 0.5) - 0.303687641897378**

*.111 is ANOMALY(thr: 0.5) - 0.0134717254763835

***.112 is ANOMALY(thr: 0.5) - 0.342663006911585**

*.112 is ANOMALY(thr: 0.5) - 0.0596806531805028

***.113 is ANOMALY(thr: 0.5) - 0.259865148267317**

*.113 is ANOMALY(thr: 0.5) - 0.264122985963633

***.115 is ANOMALY(thr: 0.5) - 0.359341129924223**

*.115 is ANOMALY(thr: 0.5) - 0.152577566194027

***.116 is ANOMALY(thr: 0.5) - 0.285690233555775**

*.116 is ANOMALY(thr: 0.5) - 0.154186261702911

***.118 is ANOMALY(thr: 0.5) - 0.349520255342567**

*.118 is ANOMALY(thr: 0.5) - 0.126527763578438

***.119 is ANOMALY(thr: 0.5) - 0.152376142058488**

*.119 is ANOMALY(thr: 0.5) - 0.127608131969523

***.121 is ANOMALY(thr: 0.5) - 0.497770600927063**

*.121 is ANOMALY(thr: 0.5) - 0.287857385925655

***.122 is ANOMALY(thr: 0.5) - 0.254309681573091**

*.122 is ANOMALY(thr: 0.5) - 0.236716800664225

***.124 is ANOMALY(thr: 0.5) - 0.351716075029846**

*.124 is ANOMALY(thr: 0.5) - 0.128998809684479

***.128 is ANOMALY(thr: 0.5) - 0.352097176021284**

*.128 is ANOMALY(thr: 0.5) - 0.130769454590564

***.130 is ANOMALY(thr: 0.5) - 0.301436911124591**

*.130 is ANOMALY(thr: 0.5) - 0.130642938567548

***.131 is ANOMALY(thr: 0.5) - 0.214215867923894**

*.131 is ANOMALY(thr: 0.5) - 0.128574136014664

***.132 is ANOMALY(thr: 0.5) - 0.128668004495994**

*.132 is ANOMALY(thr: 0.5) - 0.129379419575141

***.133 is NORMAL(thr: 0.5) - 0.940471836649492**

*.133 is ANOMALY(thr: 0.5) - 0.458308849333122

***.134 is ANOMALY(thr: 0.5) - 0.371932032191294**

*.134 is ANOMALY(thr: 0.5) - 0.129407411391239

***.135 is ANOMALY(thr: 0.5) - 0.325836740054406**

*.135 is ANOMALY(thr: 0.5) - 0.127995891898357

***.136 is ANOMALY(thr: 0.5) - 0.0742356019074597**

*.136 is ANOMALY(thr: 0.5) - 0.119578757348271

***.137 is ANOMALY(thr: 0.5) - 0.265230535738096**

*.137 is ANOMALY(thr: 0.5) - 0.116565326010426

***.146 is NORMAL(thr: 0.5) - 0.948858443201118**

*.146 is NORMAL(thr: 0.5) - 0.679969100454352

***.155 is ANOMALY(thr: 0.5) - 0.286588573522702**

*.155 is ANOMALY(thr: 0.5) - 0.130645554686059

***.166 is ANOMALY(thr: 0.5) - 0.292427724754455**

*.166 is ANOMALY(thr: 0.5) - 0.311832795167593

***.167 is ANOMALY(thr: 0.5) - 0.265223086409795**

*.167 is ANOMALY(thr: 0.5) - 0.129703399091478

***.168 is NORMAL(thr: 0.5) - 0.946198487515768**

*.168 is ANOMALY(thr: 0.5) - 0.308851894243652

***.169 is NORMAL(thr: 0.5) - 0.947498425298009**

*.169 is ANOMALY(thr: 0.5) - 0.458248717051858

***.200 is NORMAL(thr: 0.5) - 0.94896004479978**

*.200 is ANOMALY(thr: 0.5) - 0.458248717051858

***.201 is NORMAL(thr: 0.5) - 0.947441946722131**

*.201 is ANOMALY(thr: 0.5) - 0.441743912562954

***.202 is NORMAL(thr: 0.5) - 0.944262254543276**

*.202 is NORMAL(thr: 0.5) - 0.787899558388909

***.203 is NORMAL(thr: 0.5) - 0.94415309134999**

*.203 is NORMAL(thr: 0.5) - 0.554259235832344

***.204 is NORMAL(thr: 0.5) - 0.920739363566113**

*.204 is ANOMALY(thr: 0.5) - 0.283359627813357

***.205 is NORMAL(thr: 0.5) - 0.682038673491139**

*.205 is ANOMALY(thr: 0.5) - 0.0336801604960053

***.210 is NORMAL(thr: 0.5) - 0.73563320377264**

*.210 is ANOMALY(thr: 0.5) - 0.039481729566744

***.211 is ANOMALY(thr: 0.5) - 0.0737685946198971**

*.211 is ANOMALY(thr: 0.5) - 0.218599435589756

***.212 is NORMAL(thr: 0.5) - 0.934579673206871**

*.212 is NORMAL(thr: 0.5) - 0.518180457512286

***.213 is ANOMALY(thr: 0.5) - 0.320454186964501**

*.213 is ANOMALY(thr: 0.5) - 0.113997356699827

***.214 is NORMAL(thr: 0.5) - 0.735665491512704**

*.214 is ANOMALY(thr: 0.5) - 0.0175957392342232

***.215 is NORMAL(thr: 0.5) - 0.709107897756298**

*.215 is ANOMALY(thr: 0.5) - 0.0439616401218865

***.219 is NORMAL(thr: 0.5) - 0.734040430031866**

*.219 is ANOMALY(thr: 0.5) - 0.0439036039939323

***.220 is NORMAL(thr: 0.5) - 0.976712724602781**

*.220 is NORMAL(thr: 0.5) - 0.530874539033495

***.221 is NORMAL(thr: 0.5) - 0.726986764078071**

*.221 is ANOMALY(thr: 0.5) - 0.043292719450052

***.231 is ANOMALY(thr: 0.5) - 0.177843459575828**

*.231 is ANOMALY(thr: 0.5) - 0.103721114117601

***.232 is ANOMALY(thr: 0.5) - 0.236853826699777**

*.232 is ANOMALY(thr: 0.5) - 0.10359657845303648

***.233 is ANOMALY(thr: 0.5) - 0.34497159465881**

*.233 is ANOMALY(thr: 0.5) - 0.0441510003932409

***.234 is NORMAL(thr: 0.5) - 0.727009398014058**

*.234 is ANOMALY(thr: 0.5) - 0.0397209517421714

***.236 is ANOMALY(thr: 0.5) - 0.301264225598026**

*.236 is ANOMALY(thr: 0.5) - 0.0133845910937906

***.237 is NORMAL(thr: 0.5) - 0.68857872363584**

*.237 is NORMAL(thr: 0.5) - 0.537669556388652

***.238 is NORMAL(thr: 0.5) - 0.62114249671771**

*.238 is NORMAL(thr: 0.5) - 0.504692784227788

***.243 is ANOMALY(thr: 0.5) - 0.260597765300175**

*.243 is ANOMALY(thr: 0.5) - 0.113108348600155

***.244 is ANOMALY(thr: 0.5) - 0.183164373499641**

*.244 is ANOMALY(thr: 0.5) - 0.0680242618043234

***.245 is ANOMALY(thr: 0.5) - 0.268059153296761**

*.245 is ANOMALY(thr: 0.5) - 0.108099360564713

***.246 is ANOMALY(thr: 0.5) - 0.316701437665165**

*.246 is ANOMALY(thr: 0.5) - 0.120537106167356

***.247 is ANOMALY(thr: 0.5) - 0.343413215203625**

*.247 is ANOMALY(thr: 0.5) - 0.103593996056726

***.248 is ANOMALY(thr: 0.5) - 0.152088791701687**

*.248 is ANOMALY(thr: 0.5) - 0.151022816452493

APPENDIX – C – THE SIMILARITY MATRIX OF USERS ACCORDING TO REQUESTED IPS

Table C.1: The similarity matrix of users according to requested ips for the part from *.10 user to *.118 user

	***.10**	***.100**	***.101**	***.107**	***.108**	***.109**	***.110**	***.111**	***.112**	***.113**	***.115**	***.116**	***.118**
***.10**	1.000	0.071	0.030	0.012	0.013	0.015	0.038	0.019	0.029	0.030	0.012	0.012	0.014
***.12**	0.130	0.028	0.016	0.010	0.007	0.009	0.020	0.012	0.012	0.016	0.007	0.008	0.008
***.17**	0.348	0.050	0.002	0.008	0.005	0.010	0.008	0.018	0.011	0.002	0.009	0.007	0.007
***.21**	0.054	0.187	0.481	0.014	0.047	0.020	0.013	0.028	0.049	0.481	0.020	0.016	0.030
***.23**	0.002	0.016	0.003	0.015	0.027	0.015	0.015	0.010	0.018	0.003	0.011	0.012	0.012
***.31**	0.005	0.012	0.004	0.013	0.024	0.013	0.028	0.015	0.013	0.004	0.009	0.010	0.010
***.44**	0.003	0.009	0.004	0.007	0.020	0.010	0.011	0.007	0.013	0.004	0.003	0.003	0.003
***.5**	0.187	0.098	0.071	0.000	0.000	0.005	0.054	0.000	0.000	0.071	0.000	0.000	0.000
***.55**	0.002	0.008	0.003	0.006	0.018	0.009	0.010	0.006	0.012	0.003	0.003	0.003	0.003
***.6**	0.020	0.169	0.028	0.048	0.061	0.084	0.076	0.150	0.154	0.028	0.051	0.051	0.062
***.60**	0.003	0.032	0.003	0.007	0.020	0.025	0.011	0.007	0.013	0.003	0.003	0.003	0.004
***.65**	0.003	0.021	0.014	0.001	0.031	0.001	0.015	0.031	0.017	0.014	0.001	0.001	0.002
***.7**	0.011	0.086	0.008	0.023	0.029	0.032	0.026	0.032	0.048	0.008	0.025	0.016	0.030
***.70**	0.126	0.224	0.005	0.070	0.073	0.084	0.132	0.105	0.160	0.005	0.088	0.057	0.074
***.71**	0.041	0.329	0.028	0.067	0.080	0.090	0.084	0.133	0.153	0.028	0.071	0.060	0.087
***.72**	0.003	0.022	0.014	0.002	0.031	0.001	0.016	0.031	0.017	0.014	0.002	0.002	0.002
***.75**	0.017	0.102	0.014	0.027	0.076	0.053	0.051	0.090	0.085	0.014	0.035	0.030	0.032
***.76**	0.027	0.361	0.006	0.100	0.115	0.153	0.111	0.208	0.304	0.006	0.095	0.071	0.100
***.77**	0.036	0.182	0.009	0.054	0.065	0.078	0.075	0.126	0.147	0.009	0.054	0.044	0.067
***.78**	0.011	0.095	0.009	0.035	0.041	0.028	0.068	0.053	0.097	0.009	0.024	0.050	0.104
***.8**	0.027	0.065	0.887	0.011	0.011	0.006	0.003	0.011	0.006	0.887	0.009	0.010	0.010
***.80**	0.002	0.017	0.011	0.001	0.025	0.001	0.012	0.052	0.013	0.011	0.015	0.001	0.001
***.81**	0.006	0.060	0.002	0.005	0.013	0.009	0.010	0.034	0.026	0.002	0.002	0.002	0.025
***.82**	0.019	0.206	0.004	0.073	0.054	0.067	0.122	0.099	0.179	0.004	0.069	0.062	0.067
***.88**	0.002	0.014	0.003	0.011	0.024	0.013	0.011	0.007	0.012	0.003	0.007	0.008	0.008
***.9**	0.004	0.023	0.009	0.002	0.057	0.017	0.024	0.042	0.049	0.009	0.002	0.002	0.002
***.97**	0.018	0.220	0.006	0.063	0.107	0.110	0.115	0.128	0.130	0.006	0.062	0.050	0.061
***.100**	0.071	1.000	0.049	0.098	0.112	0.143	0.124	0.199	0.273	0.049	0.108	0.085	0.109
***.101**	0.030	0.049	1.000	0.005	0.007	0.003	0.004	0.007	0.007	1.000	0.004	0.004	0.004
***.107**	0.012	0.098	0.005	1.000	0.650	0.489	0.045	0.036	0.071	0.005	0.749	0.832	0.846
***.108**	0.013	0.112	0.007	0.650	1.000	0.419	0.088	0.103	0.108	0.007	0.584	0.656	0.664
***.109**	0.015	0.143	0.003	0.489	0.419	1.000	0.066	0.073	0.112	0.003	0.444	0.496	0.505
***.110**	0.038	0.124	0.004	0.045	0.088	0.066	1.000	0.095	0.111	0.004	0.029	0.037	0.036
***.111**	0.019	0.199	0.007	0.036	0.103	0.073	0.095	1.000	0.191	0.007	0.052	0.034	0.049
***.112**	0.029	0.273	0.007	0.071	0.108	0.112	0.111	0.191	1.000	0.007	0.076	0.065	0.111
***.113**	0.030	0.049	1.000	0.005	0.007	0.003	0.004	0.007	0.007	1.000	0.004	0.004	0.004

***.115**	0.012	0.108	0.004	0.749	0.584	0.444	0.029	0.052	0.076	0.004	1.000	0.756	0.780
***.116**	0.012	0.085	0.004	0.832	0.656	0.496	0.037	0.034	0.065	0.004	0.756	1.000	0.854
***.118**	0.014	0.109	0.004	0.846	0.664	0.505	0.036	0.049	0.111	0.004	0.780	0.854	1.000
***.119**	0.011	0.080	0.003	0.886	0.707	0.535	0.029	0.032	0.053	0.003	0.803	0.902	0.913
***.121**	0.020	0.044	0.410	0.005	0.017	0.018	0.033	0.025	0.026	0.410	0.004	0.005	0.022
***.122**	0.031	0.050	0.998	0.007	0.008	0.004	0.005	0.008	0.009	0.998	0.006	0.006	0.006
***.124**	0.010	0.061	0.003	0.894	0.717	0.533	0.029	0.027	0.046	0.003	0.806	0.912	0.922
***.128**	0.010	0.061	0.003	0.894	0.717	0.533	0.029	0.027	0.046	0.003	0.806	0.912	0.922
***.130**	0.018	0.142	0.021	0.793	0.640	0.508	0.054	0.075	0.083	0.021	0.727	0.801	0.828
***.131**	0.022	0.157	0.019	0.777	0.618	0.483	0.069	0.073	0.128	0.019	0.695	0.768	0.787
***.132**	0.015	0.130	0.021	0.804	0.650	0.506	0.064	0.070	0.094	0.021	0.726	0.805	0.842
***.133**	0.387	0.044	0.003	0.005	0.003	0.009	0.002	0.016	0.012	0.003	0.007	0.005	0.005
***.134**	0.026	0.272	0.006	0.080	0.148	0.094	0.108	0.125	0.173	0.006	0.084	0.058	0.060
***.135**	0.309	0.101	0.003	0.026	0.004	0.026	0.012	0.046	0.087	0.003	0.042	0.033	0.030
***.136**	0.033	0.072	0.800	0.004	0.005	0.026	0.032	0.017	0.031	0.800	0.003	0.004	0.042
***.137**	0.023	0.269	0.007	0.084	0.082	0.096	0.091	0.147	0.192	0.007	0.092	0.060	0.090
***.146**	0.387	0.044	0.003	0.005	0.003	0.009	0.002	0.016	0.012	0.003	0.007	0.005	0.005
***.155**	0.315	0.145	0.006	0.049	0.019	0.060	0.015	0.060	0.118	0.006	0.070	0.046	0.062
***.166**	0.034	0.243	0.014	0.058	0.074	0.070	0.103	0.139	0.218	0.014	0.080	0.047	0.091
***.167**	0.010	0.147	0.006	0.030	0.026	0.060	0.031	0.061	0.057	0.006	0.036	0.050	0.030
***.168**	0.378	0.048	0.003	0.010	0.007	0.012	0.002	0.017	0.012	0.003	0.011	0.010	0.010
***.169**	0.330	0.087	0.005	0.014	0.027	0.036	0.030	0.055	0.069	0.005	0.018	0.011	0.024
***.200**	0.387	0.044	0.003	0.005	0.003	0.009	0.002	0.016	0.012	0.003	0.007	0.005	0.005
***.201**	0.387	0.044	0.003	0.005	0.003	0.009	0.002	0.016	0.012	0.003	0.007	0.005	0.005
***.202**	0.387	0.044	0.003	0.005	0.003	0.009	0.002	0.016	0.012	0.003	0.007	0.005	0.005
***.203**	0.387	0.044	0.003	0.005	0.003	0.009	0.002	0.016	0.012	0.003	0.007	0.005	0.005
***.204**	0.387	0.044	0.003	0.005	0.003	0.009	0.002	0.016	0.012	0.003	0.007	0.005	0.005
***.205**	0.003	0.021	0.014	0.001	0.031	0.001	0.015	0.031	0.017	0.014	0.001	0.001	0.002
***.210**	0.003	0.021	0.014	0.001	0.031	0.001	0.015	0.031	0.017	0.014	0.001	0.001	0.002
***.211**	0.053	0.153	0.306	0.038	0.027	0.036	0.018	0.033	0.040	0.306	0.034	0.035	0.035
***.212**	0.361	0.076	0.004	0.031	0.034	0.028	0.013	0.029	0.035	0.004	0.032	0.031	0.031
***.213**	0.018	0.153	0.007	0.063	0.072	0.049	0.089	0.108	0.146	0.007	0.078	0.068	0.062
***.214**	0.003	0.008	0.003	0.007	0.020	0.010	0.011	0.007	0.013	0.003	0.003	0.003	0.003
***.215**	0.004	0.097	0.004	0.047	0.026	0.041	0.025	0.049	0.066	0.004	0.024	0.021	0.022
***.219**	0.003	0.008	0.003	0.007	0.020	0.010	0.011	0.007	0.013	0.003	0.003	0.003	0.003
***.220**	0.001	0.002	0.007	0.002	0.001	0.001	0.002	0.001	0.003	0.007	0.002	0.002	0.002
***.221**	0.003	0.008	0.003	0.007	0.020	0.010	0.011	0.007	0.013	0.003	0.003	0.003	0.003
***.231**	0.017	0.174	0.008	0.057	0.016	0.079	0.024	0.101	0.146	0.008	0.064	0.030	0.070
***.232**	0.016	0.120	0.008	0.119	0.038	0.069	0.050	0.075	0.137	0.008	0.058	0.051	0.050
***.233**	0.013	0.102	0.008	0.048	0.023	0.033	0.011	0.094	0.088	0.008	0.129	0.026	0.053
***.234**	0.003	0.008	0.003	0.007	0.020	0.010	0.011	0.007	0.013	0.003	0.003	0.003	0.003
***.236**	0.009	0.107	0.005	0.051	0.018	0.045	0.073	0.073	0.125	0.005	0.079	0.050	0.075

***.237**	0.001	0.002	0.007	0.002	0.001	0.001	0.002	0.001	0.003	0.007	0.002	0.002	0.002
***.238**	0.001	0.002	0.007	0.002	0.001	0.001	0.002	0.001	0.003	0.007	0.002	0.002	0.002
***.243**	0.041	0.274	0.016	0.078	0.109	0.114	0.118	0.167	0.223	0.016	0.080	0.059	0.085
***.244**	0.016	0.113	0.040	0.044	0.061	0.032	0.046	0.079	0.120	0.040	0.030	0.024	0.034
***.245**	0.012	0.055	0.024	0.021	0.022	0.025	0.021	0.023	0.057	0.024	0.035	0.023	0.024
***.246**	0.007	0.074	0.009	0.035	0.065	0.047	0.058	0.065	0.058	0.009	0.017	0.012	0.033
***.247**	0.005	0.012	0.018	0.010	0.072	0.043	0.008	0.007	0.015	0.018	0.009	0.010	0.011
***.248**	0.013	0.149	0.010	0.074	0.024	0.077	0.037	0.070	0.216	0.010	0.134	0.081	0.121

Table C.2: The similarity matrix of users according to requested ips for the part from *.119 user to *.136 user

	***.119**	***.12**	***.121**	***.122**	***.124**	***.128**	***.130**	***.131**	***.132**	***.133**	***.134**	***.135**	***.136**
***.10**	0.011	0.130	0.020	0.031	0.010	0.010	0.018	0.022	0.015	0.387	0.026	0.309	0.033
***.12**	0.008	1.000	0.008	0.016	0.006	0.006	0.010	0.011	0.009	0.038	0.015	0.034	0.031
***.17**	0.007	0.037	0.001	0.005	0.007	0.007	0.011	0.015	0.013	0.900	0.013	0.711	0.002
***.21**	0.011	0.055	0.257	0.482	0.011	0.011	0.024	0.027	0.023	0.029	0.017	0.034	0.394
***.23**	0.011	0.003	0.003	0.003	0.011	0.011	0.012	0.011	0.013	0.000	0.029	0.003	0.002
***.31**	0.009	0.003	0.003	0.006	0.008	0.008	0.015	0.009	0.009	0.008	0.026	0.009	0.003
***.44**	0.002	0.003	0.004	0.004	0.001	0.001	0.003	0.003	0.003	0.001	0.016	0.003	0.003
***.5**	0.000	0.149	0.029	0.071	0.000	0.000	0.000	0.000	0.000	0.017	0.000	0.014	0.057
***.55**	0.001	0.003	0.003	0.003	0.001	0.001	0.003	0.003	0.003	0.000	0.015	0.002	0.002
***.6**	0.037	0.008	0.055	0.030	0.033	0.033	0.130	0.113	0.123	0.012	0.107	0.049	0.061
***.60**	0.002	0.003	0.003	0.003	0.001	0.001	0.004	0.003	0.004	0.001	0.030	0.004	0.002
***.65**	0.002	0.003	0.006	0.014	0.001	0.001	0.001	0.001	0.001	0.000	0.031	0.000	0.011
***.7**	0.015	0.017	0.021	0.009	0.013	0.013	0.045	0.043	0.047	0.016	0.042	0.042	0.020
***.70**	0.052	0.052	0.006	0.008	0.047	0.047	0.079	0.079	0.090	0.012	0.132	0.049	0.005
***.71**	0.055	0.027	0.038	0.029	0.049	0.049	0.122	0.111	0.097	0.030	0.140	0.066	0.049
***.72**	0.002	0.003	0.006	0.014	0.002	0.002	0.002	0.002	0.002	0.000	0.032	0.001	0.011
***.75**	0.024	0.006	0.009	0.014	0.020	0.020	0.053	0.044	0.031	0.002	0.091	0.030	0.011
***.76**	0.055	0.012	0.033	0.008	0.044	0.044	0.118	0.144	0.121	0.010	0.277	0.056	0.048
***.77**	0.038	0.030	0.024	0.010	0.030	0.030	0.069	0.078	0.070	0.013	0.120	0.039	0.032
***.78**	0.018	0.006	0.006	0.009	0.011	0.011	0.039	0.033	0.079	0.010	0.060	0.094	0.008
***.8**	0.009	0.014	0.364	0.885	0.010	0.010	0.024	0.023	0.026	0.002	0.015	0.003	0.710
***.80**	0.001	0.020	0.005	0.011	0.001	0.001	0.016	0.001	0.001	0.000	0.025	0.021	0.009
***.81**	0.001	0.005	0.008	0.002	0.001	0.001	0.024	0.010	0.010	0.000	0.011	0.045	0.011
***.82**	0.051	0.018	0.041	0.007	0.051	0.051	0.109	0.157	0.135	0.012	0.134	0.035	0.059
***.88**	0.007	0.004	0.003	0.003	0.006	0.006	0.008	0.007	0.009	0.000	0.016	0.003	0.002
***.9**	0.001	0.009	0.004	0.009	0.001	0.001	0.002	0.013	0.002	0.001	0.075	0.001	0.007
***.97**	0.040	0.011	0.039	0.008	0.040	0.040	0.076	0.101	0.096	0.011	0.231	0.024	0.039
***.100**	0.080	0.028	0.044	0.050	0.061	0.061	0.142	0.157	0.130	0.044	0.272	0.101	0.072
***.101**	0.003	0.016	0.410	0.998	0.003	0.003	0.021	0.019	0.021	0.003	0.006	0.003	0.800
***.107**	0.886	0.010	0.005	0.007	0.894	0.894	0.793	0.777	0.804	0.005	0.080	0.026	0.004
***.108**	0.707	0.007	0.017	0.008	0.717	0.717	0.640	0.618	0.650	0.003	0.148	0.004	0.005

***.109**	0.535	0.009	0.018	0.004	0.533	0.533	0.508	0.483	0.506	0.009	0.094	0.026	0.026
***.110**	0.029	0.020	0.033	0.005	0.029	0.029	0.054	0.069	0.064	0.002	0.108	0.012	0.032
***.111**	0.032	0.012	0.025	0.008	0.027	0.027	0.075	0.073	0.070	0.016	0.125	0.046	0.017
***.112**	0.053	0.012	0.026	0.009	0.046	0.046	0.083	0.128	0.094	0.012	0.173	0.087	0.031
***.113**	0.003	0.016	0.410	0.998	0.003	0.003	0.021	0.019	0.021	0.003	0.006	0.003	0.800
***.115**	0.803	0.007	0.004	0.006	0.806	0.806	0.727	0.695	0.726	0.007	0.084	0.042	0.003
***.116**	0.902	0.008	0.005	0.006	0.912	0.912	0.801	0.768	0.805	0.005	0.058	0.033	0.004
***.118**	0.913	0.008	0.022	0.006	0.922	0.922	0.828	0.787	0.842	0.005	0.060	0.030	0.042
***.119**	1.000	0.008	0.004	0.005	0.986	0.986	0.865	0.841	0.873	0.004	0.057	0.010	0.002
***.121**	0.004	0.008	1.000	0.409	0.003	0.003	0.033	0.033	0.035	0.001	0.003	0.003	0.402
***.122**	0.005	0.016	0.409	1.000	0.005	0.005	0.023	0.021	0.023	0.005	0.008	0.006	0.799
***.124**	0.986	0.006	0.003	0.005	1.000	1.000	0.874	0.838	0.879	0.004	0.049	0.005	0.002
***.128**	0.986	0.006	0.003	0.005	1.000	1.000	0.874	0.838	0.879	0.004	0.049	0.005	0.002
***.130**	0.865	0.010	0.033	0.023	0.874	0.874	1.000	0.811	0.824	0.010	0.073	0.040	0.045
***.131**	0.841	0.011	0.033	0.021	0.838	0.838	0.811	1.000	0.797	0.006	0.116	0.041	0.040
***.132**	0.873	0.009	0.035	0.023	0.879	0.879	0.824	0.797	1.000	0.005	0.094	0.029	0.042
***.133**	0.004	0.038	0.001	0.005	0.004	0.004	0.010	0.006	0.005	1.000	0.015	0.790	0.002
***.134**	0.057	0.015	0.003	0.008	0.049	0.049	0.073	0.116	0.094	0.015	1.000	0.050	0.005
***.135**	0.010	0.034	0.003	0.006	0.005	0.005	0.040	0.041	0.029	0.790	0.050	1.000	0.003
***.136**	0.002	0.031	0.402	0.799	0.002	0.002	0.045	0.040	0.042	0.002	0.005	0.003	1.000
***.137**	0.054	0.015	0.007	0.010	0.048	0.048	0.081	0.105	0.090	0.023	0.146	0.065	0.006
***.146**	0.004	0.038	0.001	0.005	0.004	0.004	0.010	0.006	0.005	1.000	0.015	0.790	0.002
***.155**	0.031	0.035	0.006	0.008	0.014	0.014	0.042	0.060	0.060	0.797	0.052	0.687	0.005
***.166**	0.040	0.013	0.045	0.017	0.040	0.040	0.084	0.069	0.088	0.011	0.156	0.039	0.047
***.167**	0.029	0.004	0.004	0.009	0.029	0.029	0.080	0.061	0.057	0.018	0.094	0.031	0.005
***.168**	0.010	0.037	0.001	0.005	0.010	0.010	0.014	0.011	0.010	0.978	0.018	0.772	0.002
***.169**	0.010	0.033	0.006	0.007	0.009	0.009	0.029	0.025	0.017	0.848	0.052	0.671	0.004
***.200**	0.004	0.038	0.001	0.005	0.004	0.004	0.010	0.006	0.005	1.000	0.015	0.790	0.002
***.201**	0.004	0.038	0.001	0.005	0.004	0.004	0.010	0.006	0.005	1.000	0.015	0.790	0.002
***.202**	0.004	0.038	0.001	0.005	0.004	0.004	0.010	0.006	0.005	1.000	0.015	0.790	0.002
***.203**	0.004	0.038	0.001	0.005	0.004	0.004	0.010	0.006	0.005	1.000	0.015	0.790	0.002
***.204**	0.004	0.038	0.001	0.005	0.004	0.004	0.010	0.006	0.005	1.000	0.015	0.790	0.002
***.205**	0.002	0.003	0.006	0.014	0.001	0.001	0.001	0.001	0.001	0.000	0.031	0.000	0.011
***.210**	0.002	0.003	0.006	0.014	0.001	0.001	0.001	0.001	0.001	0.000	0.031	0.000	0.011
***.211**	0.028	0.081	0.134	0.307	0.026	0.026	0.051	0.041	0.041	0.017	0.031	0.031	0.251
***.212**	0.032	0.037	0.001	0.006	0.029	0.029	0.035	0.034	0.029	0.920	0.039	0.735	0.003
***.213**	0.051	0.008	0.030	0.010	0.051	0.051	0.080	0.077	0.080	0.018	0.166	0.060	0.039
***.214**	0.001	0.003	0.003	0.003	0.001	0.001	0.003	0.003	0.003	0.000	0.016	0.003	0.002
***.215**	0.047	0.003	0.004	0.004	0.015	0.015	0.021	0.029	0.021	0.001	0.020	0.049	0.004
***.219**	0.001	0.003	0.003	0.003	0.001	0.001	0.003	0.003	0.003	0.000	0.016	0.003	0.002
***.220**	0.002	0.000	0.001	0.007	0.002	0.002	0.002	0.002	0.002	0.003	0.002	0.003	0.004
***.221**	0.001	0.003	0.003	0.003	0.001	0.001	0.003	0.003	0.003	0.000	0.016	0.003	0.002

***.231**	0.023	0.003	0.005	0.008	0.015	0.015	0.042	0.040	0.046	0.004	0.019	0.076	0.006
***.232**	0.036	0.007	0.009	0.010	0.021	0.021	0.076	0.100	0.078	0.007	0.059	0.099	0.007
***.233**	0.017	0.004	0.005	0.008	0.009	0.009	0.069	0.055	0.067	0.012	0.068	0.097	0.007
***.234**	0.001	0.003	0.003	0.003	0.001	0.001	0.003	0.003	0.003	0.000	0.016	0.003	0.002
***.236**	0.030	0.006	0.004	0.005	0.010	0.010	0.032	0.042	0.036	0.011	0.062	0.068	0.004
***.237**	0.002	0.000	0.001	0.007	0.002	0.002	0.002	0.002	0.002	0.003	0.002	0.003	0.004
***.238**	0.002	0.000	0.001	0.007	0.002	0.002	0.002	0.002	0.002	0.003	0.002	0.003	0.004
***.243**	0.057	0.018	0.026	0.017	0.045	0.045	0.100	0.127	0.113	0.013	0.206	0.072	0.039
***.244**	0.021	0.006	0.019	0.042	0.018	0.018	0.044	0.061	0.058	0.020	0.105	0.040	0.032
***.245**	0.020	0.007	0.014	0.025	0.018	0.018	0.036	0.031	0.025	0.008	0.033	0.018	0.019
***.246**	0.011	0.023	0.006	0.009	0.011	0.011	0.066	0.026	0.055	0.003	0.049	0.027	0.007
***.247**	0.008	0.002	0.010	0.018	0.007	0.007	0.010	0.010	0.011	0.002	0.013	0.005	0.015
***.248**	0.032	0.008	0.006	0.015	0.013	0.013	0.044	0.058	0.067	0.008	0.091	0.096	0.008

Table C.3: The similarity matrix of users according to requested ips for the part from *.137 user to *.204 user

	***.137**	***.146**	***.155**	***.166**	***.167**	***.168**	***.169**	***.17**	***.200**	***.201**	***.202**	***.203**	***.204**
***.10**	0.023	0.387	0.315	0.034	0.010	0.378	0.330	0.348	0.387	0.387	0.387	0.387	0.387
***.12**	0.015	0.038	0.035	0.013	0.004	0.037	0.033	0.037	0.038	0.038	0.038	0.038	0.038
***.17**	0.043	0.900	0.717	0.010	0.016	0.881	0.764	1.000	0.900	0.900	0.900	0.900	0.900
***.21**	0.047	0.029	0.058	0.052	0.025	0.029	0.042	0.026	0.029	0.029	0.029	0.029	0.029
***.23**	0.031	0.000	0.004	0.005	0.003	0.009	0.003	0.000	0.000	0.000	0.000	0.000	0.000
***.31**	0.062	0.008	0.010	0.007	0.005	0.008	0.017	0.027	0.008	0.008	0.008	0.008	0.008
***.44**	0.032	0.001	0.004	0.006	0.003	0.001	0.004	0.001	0.001	0.001	0.001	0.001	0.001
***.5**	0.000	0.017	0.014	0.001	0.000	0.017	0.015	0.016	0.017	0.017	0.017	0.017	0.017
***.55**	0.029	0.000	0.004	0.005	0.003	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000
***.6**	0.108	0.012	0.070	0.117	0.059	0.012	0.061	0.011	0.012	0.012	0.012	0.012	0.012
***.60**	0.031	0.001	0.005	0.007	0.004	0.001	0.004	0.001	0.001	0.001	0.001	0.001	0.001
***.65**	0.001	0.000	0.002	0.050	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000
***.7**	0.045	0.016	0.035	0.059	0.027	0.015	0.022	0.020	0.016	0.016	0.016	0.016	0.016
***.70**	0.190	0.012	0.099	0.094	0.098	0.018	0.028	0.019	0.012	0.012	0.012	0.012	0.012
***.71**	0.161	0.030	0.080	0.142	0.072	0.032	0.054	0.027	0.030	0.030	0.030	0.030	0.030
***.72**	0.002	0.000	0.003	0.051	0.001	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000
***.75**	0.046	0.003	0.034	0.093	0.021	0.002	0.017	0.002	0.002	0.002	0.002	0.002	0.002
***.76**	0.239	0.010	0.136	0.226	0.127	0.028	0.050	0.009	0.010	0.010	0.010	0.010	0.010
***.77**	0.150	0.013	0.060	0.119	0.069	0.014	0.036	0.015	0.013	0.013	0.013	0.013	0.013
***.78**	0.043	0.010	0.087	0.099	0.071	0.010	0.011	0.009	0.010	0.010	0.010	0.010	0.010
***.8**	0.006	0.002	0.005	0.012	0.066	0.013	0.005	0.002	0.002	0.002	0.002	0.002	0.002
***.80**	0.001	0.000	0.002	0.040	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000
***.81**	0.035	0.000	0.003	0.048	0.002	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000
***.82**	0.235	0.012	0.045	0.138	0.123	0.042	0.011	0.038	0.012	0.012	0.012	0.012	0.012
***.88**	0.031	0.000	0.004	0.005	0.003	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000
***.9**	0.012	0.001	0.002	0.023	0.001	0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.001

***.97**	0.222	0.011	0.058	0.154	0.063	0.014	0.039	0.010	0.011	0.011	0.011	0.011	0.011
***.100**	0.269	0.044	0.145	0.243	0.147	0.048	0.087	0.050	0.044	0.044	0.044	0.044	0.044
***.101**	0.007	0.003	0.006	0.014	0.006	0.003	0.005	0.002	0.003	0.003	0.003	0.003	0.003
***.107**	0.084	0.005	0.049	0.058	0.030	0.010	0.014	0.008	0.005	0.005	0.005	0.005	0.005
***.108**	0.082	0.003	0.019	0.074	0.026	0.007	0.027	0.005	0.003	0.003	0.003	0.003	0.003
***.109**	0.096	0.009	0.060	0.070	0.060	0.012	0.036	0.010	0.009	0.009	0.009	0.009	0.009
***.110**	0.091	0.002	0.015	0.103	0.031	0.002	0.030	0.008	0.002	0.002	0.002	0.002	0.002
***.111**	0.147	0.016	0.060	0.139	0.061	0.017	0.055	0.018	0.016	0.016	0.016	0.016	0.016
***.112**	0.192	0.012	0.118	0.218	0.057	0.012	0.069	0.011	0.012	0.012	0.012	0.012	0.012
***.113**	0.007	0.003	0.006	0.014	0.006	0.003	0.005	0.002	0.003	0.003	0.003	0.003	0.003
***.115**	0.092	0.007	0.070	0.080	0.036	0.011	0.018	0.009	0.007	0.007	0.007	0.007	0.007
***.116**	0.060	0.005	0.046	0.047	0.050	0.010	0.011	0.007	0.005	0.005	0.005	0.005	0.005
***.118**	0.090	0.005	0.062	0.091	0.030	0.010	0.024	0.007	0.005	0.005	0.005	0.005	0.005
***.119**	0.054	0.004	0.031	0.040	0.029	0.010	0.010	0.007	0.004	0.004	0.004	0.004	0.004
***.121**	0.007	0.001	0.006	0.045	0.004	0.001	0.006	0.001	0.001	0.001	0.001	0.001	0.001
***.122**	0.010	0.005	0.008	0.017	0.009	0.005	0.007	0.005	0.005	0.005	0.005	0.005	0.005
***.124**	0.048	0.004	0.014	0.040	0.029	0.010	0.009	0.007	0.004	0.004	0.004	0.004	0.004
***.128**	0.048	0.004	0.014	0.040	0.029	0.010	0.009	0.007	0.004	0.004	0.004	0.004	0.004
***.130**	0.081	0.010	0.042	0.084	0.080	0.014	0.029	0.011	0.010	0.010	0.010	0.010	0.010
***.131**	0.105	0.006	0.060	0.069	0.061	0.011	0.025	0.015	0.006	0.006	0.006	0.006	0.006
***.132**	0.090	0.005	0.060	0.088	0.057	0.010	0.017	0.013	0.005	0.005	0.005	0.005	0.005
***.133**	0.023	1.000	0.797	0.011	0.018	0.978	0.848	0.900	1.000	1.000	1.000	1.000	1.000
***.134**	0.146	0.015	0.052	0.156	0.094	0.018	0.052	0.013	0.015	0.015	0.015	0.015	0.015
***.135**	0.065	0.790	0.687	0.039	0.031	0.772	0.671	0.711	0.790	0.790	0.790	0.790	0.790
***.136**	0.006	0.002	0.005	0.047	0.005	0.002	0.004	0.002	0.002	0.002	0.002	0.002	0.002
***.137**	1.000	0.023	0.085	0.144	0.089	0.023	0.034	0.043	0.023	0.023	0.023	0.023	0.023
***.146**	0.023	1.000	0.797	0.011	0.018	0.978	0.848	0.900	1.000	1.000	1.000	1.000	1.000
***.155**	0.085	0.797	1.000	0.058	0.061	0.779	0.728	0.717	0.797	0.797	0.797	0.797	0.797
***.166**	0.144	0.011	0.058	1.000	0.021	0.011	0.019	0.010	0.011	0.011	0.011	0.011	0.011
***.167**	0.089	0.018	0.061	0.021	1.000	0.018	0.016	0.016	0.018	0.018	0.018	0.018	0.018
***.168**	0.023	0.978	0.779	0.011	0.018	1.000	0.830	0.881	0.978	0.978	0.978	0.978	0.978
***.169**	0.034	0.848	0.728	0.019	0.016	0.830	1.000	0.764	0.848	0.848	0.848	0.848	0.848
***.200**	0.023	1.000	0.797	0.011	0.018	0.978	0.848	0.900	1.000	1.000	1.000	1.000	1.000
***.201**	0.023	1.000	0.797	0.011	0.018	0.978	0.848	0.900	1.000	1.000	1.000	1.000	1.000
***.202**	0.023	1.000	0.797	0.011	0.018	0.978	0.848	0.900	1.000	1.000	1.000	1.000	1.000
***.203**	0.023	1.000	0.797	0.011	0.018	0.978	0.848	0.900	1.000	1.000	1.000	1.000	1.000
***.204**	0.023	1.000	0.797	0.011	0.018	0.978	0.848	0.900	1.000	1.000	1.000	1.000	1.000
***.205**	0.001	0.000	0.002	0.050	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000
***.210**	0.001	0.000	0.002	0.050	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000
***.211**	0.056	0.017	0.038	0.045	0.035	0.016	0.025	0.015	0.017	0.017	0.017	0.017	0.017
***.212**	0.043	0.920	0.745	0.021	0.047	0.900	0.781	0.829	0.920	0.920	0.920	0.920	0.920
***.213**	0.097	0.018	0.090	0.149	0.031	0.018	0.037	0.022	0.018	0.018	0.018	0.018	0.018
***.214**	0.031	0.000	0.004	0.005	0.003	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000

***.215**	0.027	0.001	0.020	0.006	0.003	0.001	0.004	0.001	0.001	0.001	0.001	0.001	0.001
***.219**	0.031	0.000	0.004	0.005	0.003	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000
***.220**	0.003	0.004	0.002	0.004	0.004	0.003	0.003	0.002	0.003	0.003	0.003	0.003	0.003
***.221**	0.031	0.000	0.004	0.005	0.003	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000
***.231**	0.135	0.004	0.053	0.063	0.088	0.004	0.006	0.003	0.004	0.004	0.004	0.004	0.004
***.232**	0.130	0.007	0.103	0.107	0.067	0.007	0.032	0.007	0.007	0.007	0.007	0.007	0.007
***.233**	0.116	0.012	0.127	0.139	0.033	0.012	0.032	0.011	0.012	0.012	0.012	0.012	0.012
***.234**	0.031	0.000	0.004	0.005	0.003	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000
***.236**	0.062	0.011	0.102	0.091	0.022	0.011	0.056	0.010	0.011	0.011	0.011	0.011	0.011
***.237**	0.003	0.004	0.002	0.004	0.004	0.003	0.003	0.002	0.003	0.003	0.003	0.003	0.003
***.238**	0.003	0.004	0.002	0.004	0.004	0.003	0.003	0.002	0.003	0.003	0.003	0.003	0.003
***.243**	0.186	0.013	0.082	0.200	0.130	0.013	0.063	0.012	0.013	0.013	0.013	0.013	0.013
***.244**	0.064	0.020	0.095	0.135	0.016	0.020	0.021	0.032	0.020	0.020	0.020	0.020	0.020
***.245**	0.043	0.008	0.023	0.041	0.029	0.007	0.015	0.014	0.008	0.008	0.008	0.008	0.008
***.246**	0.030	0.003	0.019	0.156	0.007	0.003	0.007	0.003	0.003	0.003	0.003	0.003	0.003
***.247**	0.014	0.002	0.015	0.019	0.006	0.002	0.014	0.002	0.002	0.002	0.002	0.002	0.002
***.248**	0.151	0.008	0.183	0.088	0.167	0.008	0.065	0.007	0.008	0.008	0.008	0.008	0.008

Table C.4: The similarity matrix of users according to requested ips for the part from *.205 user to *.231 user

	***.205**	***.21**	***.210**	***.211**	***.212**	***.213**	***.214**	***.215**	***.219**	***.220**	***.221**	***.23**	***.231**
***.10**	0.003	0.054	0.003	0.053	0.361	0.018	0.003	0.004	0.003	0.001	0.003	0.002	0.017
***.12**	0.003	0.055	0.003	0.081	0.037	0.008	0.003	0.003	0.003	0.000	0.003	0.003	0.003
***.17**	0.000	0.026	0.000	0.015	0.829	0.022	0.000	0.001	0.000	0.002	0.000	0.000	0.003
***.21**	0.019	1.000	0.019	0.484	0.032	0.022	0.004	0.008	0.004	0.002	0.004	0.003	0.039
***.23**	0.114	0.003	0.114	0.002	0.000	0.025	0.986	0.796	0.986	0.004	0.986	1.000	0.004
***.31**	0.106	0.013	0.106	0.003	0.007	0.036	0.919	0.742	0.919	0.004	0.919	0.906	0.004
***.44**	0.115	0.004	0.115	0.003	0.001	0.026	1.000	0.808	1.000	0.003	1.000	0.986	0.004
***.5**	0.000	0.243	0.000	0.216	0.016	0.001	0.001	0.001	0.001	0.002	0.001	0.001	0.000
***.55**	0.106	0.003	0.106	0.002	0.000	0.023	0.917	0.740	0.917	0.005	0.917	0.904	0.003
***.6**	0.464	0.057	0.464	0.048	0.032	0.096	0.071	0.075	0.071	0.002	0.071	0.070	0.057
***.60**	0.110	0.005	0.110	0.003	0.001	0.024	0.955	0.771	0.955	0.003	0.955	0.942	0.004
***.65**	1.000	0.019	1.000	0.005	0.000	0.013	0.115	0.095	0.115	0.003	0.115	0.114	0.003
***.7**	0.008	0.012	0.008	0.030	0.024	0.035	0.004	0.010	0.004	0.001	0.004	0.004	0.030
***.70**	0.001	0.028	0.001	0.039	0.069	0.085	0.004	0.063	0.004	0.003	0.004	0.010	0.149
***.71**	0.018	0.164	0.018	0.146	0.063	0.135	0.012	0.041	0.012	0.001	0.012	0.012	0.063
***.72**	1.000	0.020	1.000	0.006	0.000	0.013	0.116	0.096	0.116	0.000	0.116	0.115	0.004
***.75**	0.644	0.042	0.644	0.037	0.028	0.044	0.086	0.124	0.086	0.003	0.086	0.085	0.037
***.76**	0.025	0.055	0.025	0.058	0.054	0.192	0.019	0.078	0.019	0.003	0.019	0.019	0.122
***.77**	0.158	0.026	0.158	0.055	0.027	0.101	0.197	0.195	0.197	0.001	0.197	0.196	0.061
***.78**	0.586	0.030	0.586	0.021	0.024	0.041	0.071	0.080	0.071	0.004	0.071	0.070	0.030
***.8**	0.013	0.427	0.013	0.271	0.003	0.006	0.003	0.004	0.003	0.006	0.003	0.024	0.007
***.80**	0.797	0.015	0.797	0.004	0.000	0.035	0.092	0.076	0.092	0.000	0.092	0.091	0.002

***.81**	0.076	0.006	0.076	0.009	0.000	0.033	0.663	0.536	0.663	0.003	0.663	0.654	0.035
***.82**	0.000	0.100	0.000	0.078	0.031	0.127	0.002	0.023	0.002	0.003	0.002	0.002	0.054
***.88**	0.114	0.004	0.114	0.002	0.000	0.025	0.992	0.801	0.992	0.003	0.992	0.978	0.004
***.9**	0.425	0.009	0.425	0.003	0.001	0.010	0.054	0.045	0.054	0.003	0.054	0.053	0.002
***.97**	0.019	0.035	0.019	0.040	0.030	0.120	0.011	0.011	0.011	0.002	0.011	0.011	0.059
***.100**	0.021	0.187	0.021	0.153	0.076	0.153	0.008	0.097	0.008	0.002	0.008	0.016	0.174
***.101**	0.014	0.481	0.014	0.306	0.004	0.007	0.003	0.004	0.003	0.007	0.003	0.003	0.008
***.107**	0.001	0.014	0.001	0.038	0.031	0.063	0.007	0.047	0.007	0.002	0.007	0.015	0.057
***.108**	0.031	0.047	0.031	0.027	0.034	0.072	0.020	0.026	0.020	0.001	0.020	0.027	0.016
***.109**	0.001	0.020	0.001	0.036	0.028	0.049	0.010	0.041	0.010	0.001	0.010	0.015	0.079
***.110**	0.015	0.013	0.015	0.018	0.013	0.089	0.011	0.025	0.011	0.002	0.011	0.015	0.024
***.111**	0.031	0.028	0.031	0.033	0.029	0.108	0.007	0.049	0.007	0.001	0.007	0.010	0.101
***.112**	0.017	0.049	0.017	0.040	0.035	0.146	0.013	0.066	0.013	0.003	0.013	0.018	0.146
***.113**	0.014	0.481	0.014	0.306	0.004	0.007	0.003	0.004	0.003	0.007	0.003	0.003	0.008
***.115**	0.001	0.020	0.001	0.034	0.032	0.078	0.003	0.024	0.003	0.002	0.003	0.011	0.064
***.116**	0.001	0.016	0.001	0.035	0.031	0.068	0.003	0.021	0.003	0.002	0.003	0.012	0.030
***.118**	0.002	0.030	0.002	0.035	0.031	0.062	0.003	0.022	0.003	0.002	0.003	0.012	0.070
***.119**	0.002	0.011	0.002	0.028	0.032	0.051	0.001	0.047	0.001	0.002	0.001	0.011	0.023
***.121**	0.006	0.257	0.006	0.134	0.001	0.030	0.003	0.004	0.003	0.001	0.003	0.003	0.005
***.122**	0.014	0.482	0.014	0.307	0.006	0.010	0.003	0.004	0.003	0.007	0.003	0.003	0.008
***.124**	0.001	0.011	0.001	0.026	0.029	0.051	0.001	0.015	0.001	0.002	0.001	0.011	0.015
***.128**	0.001	0.011	0.001	0.026	0.029	0.051	0.001	0.015	0.001	0.002	0.001	0.011	0.015
***.130**	0.001	0.024	0.001	0.051	0.035	0.080	0.003	0.021	0.003	0.002	0.003	0.012	0.042
***.131**	0.001	0.027	0.001	0.041	0.034	0.077	0.003	0.029	0.003	0.002	0.003	0.011	0.040
***.132**	0.001	0.023	0.001	0.041	0.029	0.080	0.003	0.021	0.003	0.002	0.003	0.013	0.046
***.133**	0.000	0.029	0.000	0.017	0.920	0.018	0.000	0.001	0.000	0.003	0.000	0.000	0.004
***.134**	0.031	0.017	0.031	0.031	0.039	0.166	0.016	0.020	0.016	0.002	0.016	0.029	0.019
***.135**	0.000	0.034	0.000	0.031	0.735	0.060	0.003	0.049	0.003	0.003	0.003	0.003	0.076
***.136**	0.011	0.394	0.011	0.251	0.003	0.039	0.002	0.004	0.002	0.004	0.002	0.002	0.006
***.137**	0.001	0.047	0.001	0.056	0.043	0.097	0.031	0.027	0.031	0.003	0.031	0.031	0.135
***.146**	0.000	0.029	0.000	0.017	0.920	0.018	0.000	0.001	0.000	0.004	0.000	0.000	0.004
***.155**	0.002	0.058	0.002	0.038	0.745	0.090	0.004	0.020	0.004	0.002	0.004	0.004	0.053
***.166**	0.050	0.052	0.050	0.045	0.021	0.149	0.005	0.006	0.005	0.004	0.005	0.005	0.063
***.167**	0.000	0.025	0.000	0.035	0.047	0.031	0.003	0.003	0.003	0.004	0.003	0.003	0.088
***.168**	0.000	0.029	0.000	0.016	0.900	0.018	0.000	0.001	0.000	0.003	0.000	0.009	0.004
***.169**	0.002	0.042	0.002	0.025	0.781	0.037	0.003	0.004	0.003	0.003	0.003	0.003	0.006
***.200**	0.000	0.029	0.000	0.017	0.920	0.018	0.000	0.001	0.000	0.003	0.000	0.000	0.004
***.201**	0.000	0.029	0.000	0.017	0.920	0.018	0.000	0.001	0.000	0.003	0.000	0.000	0.004
***.202**	0.000	0.029	0.000	0.017	0.920	0.018	0.000	0.001	0.000	0.003	0.000	0.000	0.004
***.203**	0.000	0.029	0.000	0.017	0.920	0.018	0.000	0.001	0.000	0.003	0.000	0.000	0.004
***.204**	0.000	0.029	0.000	0.017	0.920	0.018	0.000	0.001	0.000	0.003	0.000	0.000	0.004
***.205**	1.000	0.019	1.000	0.005	0.000	0.013	0.115	0.095	0.115	0.003	0.115	0.114	0.003

***.210**	1.000	0.019	1.000	0.005	0.000	0.013	0.115	0.095	0.115	0.007	0.115	0.114	0.003
***.211**	0.005	0.484	0.005	1.000	0.026	0.024	0.002	0.014	0.002	0.001	0.002	0.002	0.030
***.212**	0.000	0.032	0.000	0.026	1.000	0.035	0.000	0.038	0.000	0.003	0.000	0.000	0.043
***.213**	0.013	0.022	0.013	0.024	0.035	1.000	0.025	0.022	0.025	0.003	0.025	0.025	0.016
***.214**	0.115	0.004	0.115	0.002	0.000	0.025	1.000	0.807	1.000	0.005	1.000	0.986	0.004
***.215**	0.095	0.008	0.095	0.014	0.038	0.022	0.807	1.000	0.807	0.003	0.807	0.796	0.190
***.219**	0.115	0.004	0.115	0.002	0.000	0.025	1.000	0.807	1.000	0.005	1.000	0.986	0.004
***.220**	0.003	0.002	0.007	0.001	0.003	0.003	0.005	0.003	0.005	1.000	0.005	0.004	0.005
***.221**	0.115	0.004	0.115	0.002	0.000	0.025	1.000	0.807	1.000	0.005	1.000	0.986	0.004
***.231**	0.003	0.039	0.003	0.030	0.043	0.016	0.004	0.190	0.004	0.005	0.004	0.004	1.000
***.232**	0.003	0.028	0.003	0.048	0.087	0.057	0.019	0.078	0.019	0.005	0.019	0.019	0.077
***.233**	0.003	0.029	0.003	0.022	0.029	0.089	0.006	0.031	0.006	0.004	0.006	0.006	0.167
***.234**	0.115	0.004	0.115	0.002	0.000	0.025	1.000	0.807	1.000	0.005	1.000	0.986	0.004
***.236**	0.000	0.027	0.000	0.030	0.022	0.065	0.018	0.033	0.018	0.002	0.018	0.018	0.043
***.237**	0.003	0.002	0.007	0.001	0.003	0.003	0.005	0.003	0.005	1.000	0.005	0.004	0.005
***.238**	0.003	0.002	0.007	0.001	0.003	0.003	0.005	0.003	0.005	1.000	0.005	0.004	0.005
***.243**	0.024	0.043	0.024	0.046	0.044	0.119	0.018	0.075	0.018	0.002	0.018	0.017	0.120
***.244**	0.013	0.033	0.013	0.033	0.024	0.105	0.003	0.010	0.003	0.003	0.003	0.003	0.025
***.245**	0.001	0.014	0.001	0.022	0.016	0.027	0.002	0.005	0.002	0.002	0.002	0.002	0.024
***.246**	0.020	0.018	0.020	0.146	0.003	0.021	0.006	0.037	0.006	0.005	0.006	0.006	0.103
***.247**	0.010	0.013	0.010	0.009	0.002	0.017	0.012	0.016	0.012	0.017	0.012	0.012	0.015
***.248**	0.003	0.042	0.003	0.062	0.021	0.024	0.007	0.027	0.007	0.006	0.007	0.007	0.137

Table C.5: The similarity matrix of users according to requested ips for the part from *.232 user to *.31 user

	***.232**	***.233**	***.234**	***.236**	***.237**	***.238**	***.243**	***.244**	***.245**	***.246**	***.247**	***.248**	***.31**
***.10**	0.016	0.013	0.003	0.009	0.001	0.001	0.041	0.016	0.012	0.007	0.005	0.013	0.005
***.12**	0.007	0.004	0.003	0.006	0.000	0.000	0.018	0.006	0.007	0.023	0.002	0.008	0.003
***.17**	0.007	0.011	0.000	0.010	0.002	0.002	0.012	0.032	0.014	0.003	0.002	0.007	0.027
***.21**	0.028	0.029	0.004	0.027	0.002	0.002	0.043	0.033	0.014	0.018	0.013	0.042	0.013
***.23**	0.019	0.006	0.986	0.018	0.004	0.004	0.017	0.003	0.002	0.006	0.012	0.007	0.906
***.31**	0.020	0.006	0.919	0.017	0.004	0.004	0.028	0.004	0.017	0.006	0.013	0.010	1.000
***.44**	0.020	0.007	1.000	0.018	0.003	0.003	0.018	0.003	0.002	0.007	0.014	0.008	0.919
***.5**	0.001	0.001	0.001	0.000	0.002	0.002	0.006	0.000	0.000	0.056	0.002	0.001	0.001
***.55**	0.018	0.006	0.917	0.016	0.005	0.005	0.016	0.003	0.002	0.006	0.011	0.006	0.842
***.6**	0.076	0.043	0.071	0.071	0.002	0.002	0.121	0.070	0.032	0.078	0.015	0.113	0.080
***.60**	0.020	0.008	0.955	0.018	0.003	0.003	0.025	0.004	0.002	0.053	0.011	0.009	0.878
***.65**	0.003	0.003	0.115	0.000	0.003	0.003	0.024	0.013	0.001	0.020	0.010	0.003	0.106
***.7**	0.029	0.043	0.004	0.020	0.001	0.001	0.054	0.026	0.013	0.020	0.005	0.034	0.004
***.70**	0.115	0.057	0.004	0.035	0.003	0.003	0.167	0.080	0.042	0.063	0.012	0.109	0.029
***.71**	0.091	0.097	0.012	0.081	0.001	0.001	0.159	0.115	0.032	0.089	0.067	0.058	0.015
***.72**	0.004	0.004	0.116	0.001	0.000	0.000	0.024	0.013	0.001	0.021	0.013	0.004	0.107
***.75**	0.093	0.049	0.086	0.034	0.003	0.003	0.114	0.064	0.016	0.042	0.017	0.069	0.080

***.76**	0.124	0.121	0.019	0.122	0.003	0.003	0.284	0.121	0.037	0.140	0.083	0.172	0.019
***.77**	0.087	0.075	0.197	0.056	0.001	0.001	0.162	0.063	0.030	0.066	0.033	0.085	0.185
***.78**	0.075	0.034	0.071	0.156	0.004	0.004	0.077	0.085	0.013	0.125	0.012	0.108	0.065
***.8**	0.007	0.007	0.003	0.005	0.006	0.006	0.014	0.036	0.021	0.008	0.016	0.009	0.003
***.80**	0.002	0.035	0.092	0.000	0.000	0.000	0.019	0.010	0.001	0.016	0.008	0.003	0.084
***.81**	0.013	0.033	0.663	0.033	0.003	0.003	0.030	0.002	0.008	0.063	0.009	0.005	0.610
***.82**	0.058	0.046	0.002	0.028	0.003	0.003	0.147	0.087	0.044	0.010	0.005	0.099	0.040
***.88**	0.019	0.006	0.992	0.018	0.003	0.003	0.021	0.003	0.002	0.006	0.012	0.007	0.911
***.9**	0.003	0.003	0.054	0.002	0.003	0.003	0.034	0.018	0.001	0.011	0.010	0.004	0.050
***.97**	0.085	0.074	0.011	0.047	0.002	0.002	0.184	0.043	0.025	0.055	0.013	0.088	0.020
***.100**	0.120	0.102	0.008	0.107	0.002	0.002	0.274	0.113	0.055	0.074	0.012	0.149	0.012
***.101**	0.008	0.008	0.003	0.005	0.007	0.007	0.016	0.040	0.024	0.009	0.018	0.010	0.004
***.107**	0.119	0.048	0.007	0.051	0.002	0.002	0.078	0.044	0.021	0.035	0.010	0.074	0.013
***.108**	0.038	0.023	0.020	0.018	0.001	0.001	0.109	0.061	0.022	0.065	0.072	0.024	0.024
***.109**	0.069	0.033	0.010	0.045	0.001	0.001	0.114	0.032	0.025	0.047	0.043	0.077	0.013
***.110**	0.050	0.011	0.011	0.073	0.002	0.002	0.118	0.046	0.021	0.058	0.008	0.037	0.028
***.111**	0.075	0.094	0.007	0.073	0.001	0.001	0.167	0.079	0.023	0.065	0.007	0.070	0.015
***.112**	0.137	0.088	0.013	0.125	0.003	0.003	0.223	0.120	0.057	0.058	0.015	0.216	0.013
***.113**	0.008	0.008	0.003	0.005	0.007	0.007	0.016	0.040	0.024	0.009	0.018	0.010	0.004
***.115**	0.058	0.129	0.003	0.079	0.002	0.002	0.080	0.030	0.035	0.017	0.009	0.134	0.009
***.116**	0.051	0.026	0.003	0.050	0.002	0.002	0.059	0.024	0.023	0.012	0.010	0.081	0.010
***.118**	0.050	0.053	0.003	0.075	0.002	0.002	0.085	0.034	0.024	0.033	0.011	0.121	0.010
***.119**	0.036	0.017	0.001	0.030	0.002	0.002	0.057	0.021	0.020	0.011	0.008	0.032	0.009
***.121**	0.009	0.005	0.003	0.004	0.001	0.001	0.026	0.019	0.014	0.006	0.010	0.006	0.003
***.122**	0.010	0.008	0.003	0.005	0.007	0.007	0.017	0.042	0.025	0.009	0.018	0.015	0.006
***.124**	0.021	0.009	0.001	0.010	0.002	0.002	0.045	0.018	0.018	0.011	0.007	0.013	0.008
***.128**	0.021	0.009	0.001	0.010	0.002	0.002	0.045	0.018	0.018	0.011	0.007	0.013	0.008
***.130**	0.076	0.069	0.003	0.032	0.002	0.002	0.100	0.044	0.036	0.066	0.010	0.044	0.015
***.131**	0.100	0.055	0.003	0.042	0.002	0.002	0.127	0.061	0.031	0.026	0.010	0.058	0.009
***.132**	0.078	0.067	0.003	0.036	0.002	0.002	0.113	0.058	0.025	0.055	0.011	0.067	0.009
***.133**	0.007	0.012	0.000	0.011	0.003	0.003	0.013	0.020	0.008	0.003	0.002	0.008	0.008
***.134**	0.059	0.068	0.016	0.062	0.002	0.002	0.206	0.105	0.033	0.049	0.013	0.091	0.026
***.135**	0.099	0.097	0.003	0.068	0.003	0.003	0.072	0.040	0.018	0.027	0.005	0.096	0.009
***.136**	0.007	0.007	0.002	0.004	0.004	0.004	0.039	0.032	0.019	0.007	0.015	0.008	0.003
***.137**	0.130	0.116	0.031	0.062	0.003	0.003	0.186	0.064	0.043	0.030	0.014	0.151	0.062
***.146**	0.007	0.012	0.000	0.011	0.004	0.004	0.013	0.020	0.008	0.003	0.002	0.008	0.008
***.155**	0.103	0.127	0.004	0.102	0.002	0.002	0.082	0.095	0.023	0.019	0.015	0.183	0.010
***.166**	0.107	0.139	0.005	0.091	0.004	0.004	0.200	0.135	0.041	0.156	0.019	0.088	0.007
***.167**	0.067	0.033	0.003	0.022	0.004	0.004	0.130	0.016	0.029	0.007	0.006	0.167	0.005
***.168**	0.007	0.012	0.000	0.011	0.003	0.003	0.013	0.020	0.007	0.003	0.002	0.008	0.008
***.169**	0.032	0.032	0.003	0.056	0.003	0.003	0.063	0.021	0.015	0.007	0.014	0.065	0.017
***.200**	0.007	0.012	0.000	0.011	0.003	0.003	0.013	0.020	0.008	0.003	0.002	0.008	0.008

***.201**	0.007	0.012	0.000	0.011	0.003	0.003	0.013	0.020	0.008	0.003	0.002	0.008	0.008
***.202**	0.007	0.012	0.000	0.011	0.003	0.003	0.013	0.020	0.008	0.003	0.002	0.008	0.008
***.203**	0.007	0.012	0.000	0.011	0.003	0.003	0.013	0.020	0.008	0.003	0.002	0.008	0.008
***.204**	0.007	0.012	0.000	0.011	0.003	0.003	0.013	0.020	0.008	0.003	0.002	0.008	0.008
***.205**	0.003	0.003	0.115	0.000	0.003	0.003	0.024	0.013	0.001	0.020	0.010	0.003	0.106
***.210**	0.003	0.003	0.115	0.000	0.007	0.007	0.024	0.013	0.001	0.020	0.010	0.003	0.106
***.211**	0.048	0.022	0.002	0.030	0.001	0.001	0.046	0.033	0.022	0.146	0.009	0.062	0.003
***.212**	0.087	0.029	0.000	0.022	0.003	0.003	0.044	0.024	0.016	0.003	0.002	0.021	0.007
***.213**	0.057	0.089	0.025	0.065	0.003	0.003	0.119	0.105	0.027	0.021	0.017	0.024	0.036
***.214**	0.019	0.006	1.000	0.018	0.005	0.005	0.018	0.003	0.002	0.006	0.012	0.007	0.919
***.215**	0.078	0.031	0.807	0.033	0.003	0.003	0.075	0.010	0.005	0.037	0.016	0.027	0.742
***.219**	0.019	0.006	1.000	0.018	0.005	0.005	0.018	0.003	0.002	0.006	0.012	0.007	0.919
***.220**	0.005	0.004	0.005	0.002	1.000	1.000	0.002	0.003	0.002	0.005	0.017	0.006	0.004
***.221**	0.019	0.006	1.000	0.018	0.005	0.005	0.018	0.003	0.002	0.006	0.012	0.007	0.919
***.231**	0.077	0.167	0.004	0.043	0.005	0.005	0.120	0.025	0.024	0.103	0.015	0.137	0.004
***.232**	1.000	0.100	0.019	0.098	0.005	0.005	0.134	0.101	0.017	0.065	0.021	0.144	0.020
***.233**	0.100	1.000	0.006	0.128	0.004	0.004	0.105	0.055	0.026	0.023	0.021	0.144	0.006
***.234**	0.019	0.006	1.000	0.018	0.005	0.005	0.018	0.003	0.002	0.006	0.012	0.007	0.919
***.236**	0.098	0.128	0.018	1.000	0.002	0.002	0.110	0.048	0.037	0.037	0.010	0.272	0.017
***.237**	0.005	0.004	0.005	0.002	1.000	1.000	0.002	0.003	0.002	0.005	0.017	0.006	0.004
***.238**	0.005	0.004	0.005	0.002	1.000	1.000	0.002	0.003	0.002	0.005	0.017	0.006	0.004
***.243**	0.134	0.105	0.018	0.110	0.002	0.002	1.000	0.095	0.039	0.093	0.009	0.153	0.028
***.244**	0.101	0.055	0.003	0.048	0.003	0.003	0.095	1.000	0.026	0.063	0.009	0.055	0.004
***.245**	0.017	0.026	0.002	0.037	0.002	0.002	0.039	0.026	1.000	0.004	0.008	0.041	0.017
***.246**	0.065	0.023	0.006	0.037	0.005	0.005	0.093	0.063	0.004	1.000	0.185	0.053	0.006
***.247**	0.021	0.021	0.012	0.010	0.017	0.017	0.009	0.009	0.008	0.185	1.000	0.024	0.013
***.248**	0.144	0.144	0.007	0.272	0.006	0.006	0.153	0.055	0.041	0.053	0.024	1.000	0.010

Table C.6: The similarity matrix of users according to requested ips for the part from *.44 user to *.77 user

	***.44**	***.5**	***.55**	***.6**	***.60**	***.65**	***.7**	***.70**	***.71**	***.72**	***.75**	***.76**	***.77**
***.10**	0.003	0.187	0.002	0.020	0.003	0.003	0.011	0.126	0.041	0.003	0.017	0.027	0.036
***.12**	0.003	0.149	0.003	0.008	0.003	0.003	0.017	0.052	0.027	0.003	0.006	0.012	0.030
***.17**	0.001	0.016	0.000	0.011	0.001	0.000	0.020	0.019	0.027	0.000	0.002	0.009	0.015
***.21**	0.004	0.243	0.003	0.057	0.005	0.019	0.012	0.028	0.164	0.020	0.042	0.055	0.026
***.23**	0.986	0.001	0.904	0.070	0.942	0.114	0.004	0.010	0.012	0.115	0.085	0.019	0.196
***.31**	0.919	0.001	0.842	0.080	0.878	0.106	0.004	0.029	0.015	0.107	0.080	0.019	0.185
***.44**	1.000	0.001	0.917	0.072	0.955	0.115	0.004	0.004	0.012	0.116	0.087	0.020	0.197
***.5**	0.001	1.000	0.001	0.000	0.001	0.000	0.000	0.238	0.076	0.000	0.000	0.000	0.009
***.55**	0.917	0.001	1.000	0.065	0.876	0.106	0.004	0.003	0.011	0.107	0.079	0.018	0.181
***.6**	0.072	0.000	0.065	1.000	0.069	0.464	0.070	0.075	0.156	0.465	0.404	0.188	0.165
***.60**	0.955	0.001	0.876	0.069	1.000	0.110	0.004	0.017	0.033	0.111	0.084	0.030	0.193
***.65**	0.115	0.000	0.106	0.464	0.110	1.000	0.008	0.001	0.018	1.000	0.644	0.025	0.158

***.7**	0.004	0.000	0.004	0.070	0.004	0.008	1.000	0.028	0.085	0.008	0.031	0.068	0.038
***.70**	0.004	0.238	0.003	0.075	0.017	0.001	0.028	1.000	0.141	0.002	0.118	0.207	0.117
***.71**	0.012	0.076	0.011	0.156	0.033	0.018	0.085	0.141	1.000	0.018	0.060	0.239	0.125
***.72**	0.116	0.000	0.107	0.465	0.111	1.000	0.008	0.002	0.018	1.000	0.644	0.025	0.158
***.75**	0.087	0.000	0.079	0.404	0.084	0.644	0.031	0.118	0.060	0.644	1.000	0.134	0.153
***.76**	0.020	0.000	0.018	0.188	0.030	0.025	0.068	0.207	0.239	0.025	0.134	1.000	0.195
***.77**	0.197	0.009	0.181	0.165	0.193	0.158	0.038	0.117	0.125	0.158	0.153	0.195	1.000
***.78**	0.071	0.000	0.065	0.336	0.069	0.586	0.044	0.023	0.074	0.586	0.400	0.133	0.143
***.8**	0.003	0.063	0.002	0.025	0.002	0.013	0.007	0.039	0.043	0.013	0.013	0.005	0.017
***.80**	0.092	0.117	0.084	0.370	0.088	0.797	0.014	0.001	0.025	0.796	0.513	0.020	0.126
***.81**	0.663	0.000	0.608	0.071	0.634	0.076	0.015	0.003	0.027	0.077	0.058	0.032	0.156
***.82**	0.003	0.000	0.002	0.167	0.003	0.000	0.052	0.174	0.162	0.001	0.050	0.217	0.128
***.88**	0.991	0.001	0.909	0.071	0.947	0.114	0.004	0.004	0.012	0.115	0.086	0.025	0.198
***.9**	0.054	0.016	0.049	0.233	0.051	0.425	0.018	0.002	0.010	0.425	0.287	0.031	0.078
***.97**	0.012	0.000	0.010	0.118	0.024	0.019	0.055	0.138	0.159	0.020	0.115	0.219	0.121
***.100**	0.009	0.098	0.008	0.169	0.032	0.021	0.086	0.224	0.329	0.022	0.102	0.361	0.182
***.101**	0.004	0.071	0.003	0.028	0.003	0.014	0.008	0.005	0.028	0.014	0.014	0.006	0.009
***.107**	0.007	0.000	0.006	0.048	0.007	0.001	0.023	0.070	0.067	0.002	0.027	0.100	0.054
***.108**	0.020	0.000	0.018	0.061	0.020	0.031	0.029	0.073	0.080	0.031	0.076	0.115	0.065
***.109**	0.010	0.005	0.009	0.084	0.025	0.001	0.032	0.084	0.090	0.001	0.053	0.153	0.078
***.110**	0.011	0.054	0.010	0.076	0.011	0.015	0.026	0.132	0.084	0.016	0.051	0.111	0.075
***.111**	0.007	0.000	0.006	0.150	0.007	0.031	0.032	0.105	0.133	0.031	0.090	0.208	0.126
***.112**	0.013	0.000	0.012	0.154	0.013	0.017	0.048	0.160	0.153	0.017	0.085	0.304	0.147
***.113**	0.004	0.071	0.003	0.028	0.003	0.014	0.008	0.005	0.028	0.014	0.014	0.006	0.009
***.115**	0.003	0.000	0.003	0.051	0.003	0.001	0.025	0.088	0.071	0.002	0.035	0.095	0.054
***.116**	0.003	0.000	0.003	0.051	0.003	0.001	0.016	0.057	0.060	0.002	0.030	0.071	0.044
***.118**	0.003	0.000	0.003	0.062	0.004	0.002	0.030	0.074	0.087	0.002	0.032	0.100	0.067
***.119**	0.002	0.000	0.001	0.037	0.002	0.002	0.015	0.052	0.055	0.002	0.024	0.055	0.038
***.121**	0.004	0.029	0.003	0.055	0.003	0.006	0.021	0.006	0.038	0.006	0.009	0.033	0.024
***.122**	0.004	0.071	0.003	0.030	0.003	0.014	0.009	0.008	0.029	0.014	0.014	0.008	0.010
***.124**	0.001	0.000	0.001	0.033	0.001	0.001	0.013	0.047	0.049	0.002	0.020	0.044	0.030
***.128**	0.001	0.000	0.001	0.033	0.001	0.001	0.013	0.047	0.049	0.002	0.020	0.044	0.030
***.130**	0.003	0.000	0.003	0.130	0.004	0.001	0.045	0.079	0.122	0.002	0.053	0.118	0.069
***.131**	0.003	0.000	0.003	0.113	0.003	0.001	0.043	0.079	0.111	0.002	0.044	0.144	0.078
***.132**	0.003	0.000	0.003	0.123	0.004	0.001	0.047	0.090	0.097	0.002	0.031	0.121	0.070
***.133**	0.001	0.017	0.000	0.012	0.001	0.000	0.016	0.012	0.030	0.000	0.002	0.010	0.013
***.134**	0.016	0.000	0.015	0.107	0.030	0.031	0.042	0.132	0.140	0.032	0.091	0.277	0.120
***.135**	0.003	0.014	0.002	0.049	0.004	0.000	0.042	0.049	0.066	0.001	0.030	0.056	0.039
***.136**	0.003	0.057	0.002	0.061	0.002	0.011	0.020	0.005	0.049	0.011	0.011	0.048	0.032
***.137**	0.032	0.000	0.029	0.108	0.031	0.001	0.045	0.190	0.161	0.002	0.046	0.239	0.150
***.146**	0.001	0.017	0.000	0.012	0.001	0.000	0.016	0.012	0.030	0.000	0.003	0.010	0.013
***.155**	0.004	0.014	0.004	0.070	0.005	0.002	0.035	0.099	0.080	0.003	0.034	0.136	0.060

***.166**	0.006	0.001	0.005	0.117	0.007	0.050	0.059	0.094	0.142	0.051	0.093	0.226	0.119
***.167**	0.003	0.000	0.003	0.059	0.004	0.000	0.027	0.098	0.072	0.001	0.021	0.127	0.069
***.168**	0.001	0.017	0.000	0.012	0.001	0.000	0.015	0.018	0.032	0.000	0.002	0.028	0.014
***.169**	0.004	0.015	0.003	0.061	0.004	0.002	0.022	0.028	0.054	0.003	0.017	0.050	0.036
***.200**	0.001	0.017	0.000	0.012	0.001	0.000	0.016	0.012	0.030	0.000	0.002	0.010	0.013
***.201**	0.001	0.017	0.000	0.012	0.001	0.000	0.016	0.012	0.030	0.000	0.002	0.010	0.013
***.202**	0.001	0.017	0.000	0.012	0.001	0.000	0.016	0.012	0.030	0.000	0.002	0.010	0.013
***.203**	0.001	0.017	0.000	0.012	0.001	0.000	0.016	0.012	0.030	0.000	0.002	0.010	0.013
***.204**	0.001	0.017	0.000	0.012	0.001	0.000	0.016	0.012	0.030	0.000	0.002	0.010	0.013
***.205**	0.115	0.000	0.106	0.464	0.110	1.000	0.008	0.001	0.018	1.000	0.644	0.025	0.158
***.210**	0.115	0.000	0.106	0.464	0.110	1.000	0.008	0.001	0.018	1.000	0.644	0.025	0.158
***.211**	0.003	0.216	0.002	0.048	0.003	0.005	0.030	0.039	0.146	0.006	0.037	0.058	0.055
***.212**	0.001	0.016	0.000	0.032	0.001	0.000	0.024	0.069	0.063	0.000	0.028	0.054	0.027
***.213**	0.026	0.001	0.023	0.096	0.024	0.013	0.035	0.085	0.135	0.013	0.044	0.192	0.101
***.214**	1.000	0.001	0.917	0.071	0.955	0.115	0.004	0.004	0.012	0.116	0.086	0.019	0.197
***.215**	0.808	0.001	0.740	0.075	0.771	0.095	0.010	0.063	0.041	0.096	0.124	0.078	0.195
***.219**	1.000	0.001	0.917	0.071	0.955	0.115	0.004	0.004	0.012	0.116	0.086	0.019	0.197
***.220**	0.003	0.002	0.005	0.002	0.003	0.003	0.001	0.003	0.001	0.000	0.003	0.003	0.001
***.221**	1.000	0.001	0.917	0.071	0.955	0.115	0.004	0.004	0.012	0.116	0.086	0.019	0.197
***.231**	0.004	0.000	0.003	0.057	0.004	0.003	0.030	0.149	0.063	0.004	0.037	0.122	0.061
***.232**	0.020	0.001	0.018	0.076	0.020	0.003	0.029	0.115	0.091	0.004	0.093	0.124	0.087
***.233**	0.007	0.001	0.006	0.043	0.008	0.003	0.043	0.057	0.097	0.004	0.049	0.121	0.075
***.234**	1.000	0.001	0.917	0.071	0.955	0.115	0.004	0.004	0.012	0.116	0.086	0.019	0.197
***.236**	0.018	0.000	0.016	0.071	0.018	0.000	0.020	0.035	0.081	0.001	0.034	0.122	0.056
***.237**	0.003	0.002	0.005	0.002	0.003	0.003	0.001	0.003	0.001	0.000	0.003	0.003	0.001
***.238**	0.003	0.002	0.005	0.002	0.003	0.003	0.001	0.003	0.001	0.000	0.003	0.003	0.001
***.243**	0.018	0.006	0.016	0.121	0.025	0.024	0.054	0.167	0.159	0.024	0.114	0.284	0.162
***.244**	0.003	0.000	0.003	0.070	0.004	0.013	0.026	0.080	0.115	0.013	0.064	0.121	0.063
***.245**	0.002	0.000	0.002	0.032	0.002	0.001	0.013	0.042	0.032	0.001	0.016	0.037	0.030
***.246**	0.007	0.056	0.006	0.078	0.053	0.020	0.020	0.063	0.089	0.021	0.042	0.140	0.066
***.247**	0.014	0.002	0.011	0.015	0.011	0.010	0.005	0.012	0.067	0.013	0.017	0.083	0.033
***.248**	0.008	0.001	0.006	0.113	0.009	0.003	0.034	0.109	0.058	0.004	0.069	0.172	0.085

Table C.2: The similarity matrix of users according to requested ips for the part from *.78 user to *.97 user

	***.78**	***.8**	***.80**	***.81**	***.82**	***.88**	***.9**	***.97**
***.10**	0.011	0.027	0.002	0.006	0.019	0.002	0.004	0.018
***.12**	0.006	0.014	0.020	0.005	0.018	0.004	0.009	0.011
***.17**	0.009	0.002	0.000	0.000	0.038	0.000	0.001	0.010
***.21**	0.030	0.427	0.015	0.006	0.100	0.004	0.009	0.035
***.23**	0.070	0.024	0.091	0.654	0.002	0.978	0.053	0.011
***.31**	0.065	0.003	0.084	0.610	0.040	0.911	0.050	0.020
***.44**	0.071	0.003	0.092	0.663	0.003	0.991	0.054	0.012

***.5**	0.000	0.063	0.117	0.000	0.000	0.001	0.016	0.000
***.55**	0.065	0.002	0.084	0.608	0.002	0.909	0.049	0.010
***.6**	0.336	0.025	0.370	0.071	0.167	0.071	0.233	0.118
***.60**	0.069	0.002	0.088	0.634	0.003	0.947	0.051	0.024
***.65**	0.586	0.013	0.797	0.076	0.000	0.114	0.425	0.019
***.7**	0.044	0.007	0.014	0.015	0.052	0.004	0.018	0.055
***.70**	0.023	0.039	0.001	0.003	0.174	0.004	0.002	0.138
***.71**	0.074	0.043	0.025	0.027	0.162	0.012	0.010	0.159
***.72**	0.586	0.013	0.796	0.077	0.001	0.115	0.425	0.020
***.75**	0.400	0.013	0.513	0.058	0.050	0.086	0.287	0.115
***.76**	0.133	0.005	0.020	0.032	0.217	0.025	0.031	0.219
***.77**	0.143	0.017	0.126	0.156	0.128	0.198	0.078	0.121
***.78**	1.000	0.008	0.467	0.098	0.027	0.070	0.249	0.031
***.8**	0.008	1.000	0.010	0.002	0.041	0.003	0.008	0.005
***.80**	0.467	0.010	1.000	0.061	0.000	0.091	0.338	0.016
***.81**	0.098	0.002	0.061	1.000	0.006	0.658	0.036	0.012
***.82**	0.027	0.041	0.000	0.006	1.000	0.002	0.001	0.176
***.88**	0.070	0.003	0.091	0.658	0.002	1.000	0.053	0.017
***.9**	0.249	0.008	0.338	0.036	0.001	0.053	1.000	0.058
***.97**	0.031	0.005	0.016	0.012	0.176	0.017	0.058	1.000
***.100**	0.095	0.065	0.017	0.060	0.227	0.014	0.038	0.276
***.101**	0.009	0.887	0.011	0.002	0.005	0.003	0.014	0.008
***.107**	0.035	0.011	0.001	0.005	0.073	0.011	0.002	0.063
***.108**	0.041	0.011	0.025	0.013	0.054	0.024	0.057	0.107
***.109**	0.028	0.006	0.001	0.009	0.067	0.013	0.017	0.110
***.110**	0.068	0.003	0.012	0.010	0.122	0.011	0.024	0.115
***.111**	0.053	0.011	0.052	0.034	0.099	0.007	0.042	0.128
***.112**	0.097	0.006	0.013	0.026	0.179	0.012	0.049	0.130
***.113**	0.009	0.887	0.011	0.002	0.004	0.003	0.009	0.006
***.115**	0.024	0.009	0.015	0.002	0.069	0.007	0.002	0.062
***.116**	0.050	0.010	0.001	0.002	0.062	0.008	0.002	0.050
***.118**	0.104	0.010	0.001	0.025	0.067	0.008	0.002	0.061
***.119**	0.018	0.009	0.001	0.001	0.051	0.007	0.001	0.040
***.121**	0.006	0.364	0.005	0.008	0.041	0.003	0.004	0.039
***.122**	0.009	0.885	0.011	0.002	0.007	0.003	0.009	0.008
***.124**	0.011	0.010	0.001	0.001	0.051	0.006	0.001	0.040
***.128**	0.011	0.010	0.001	0.001	0.051	0.006	0.001	0.040
***.130**	0.039	0.024	0.016	0.024	0.109	0.008	0.002	0.076
***.131**	0.033	0.023	0.001	0.010	0.157	0.007	0.013	0.101
***.132**	0.079	0.026	0.001	0.010	0.135	0.009	0.002	0.096
***.133**	0.010	0.002	0.000	0.000	0.012	0.000	0.001	0.011
***.134**	0.060	0.015	0.025	0.011	0.134	0.016	0.075	0.231

***.135**	0.094	0.003	0.021	0.045	0.035	0.003	0.001	0.024
***.136**	0.008	0.710	0.009	0.011	0.059	0.002	0.007	0.039
***.137**	0.043	0.006	0.001	0.035	0.235	0.031	0.012	0.222
***.146**	0.010	0.002	0.000	0.000	0.012	0.000	0.001	0.011
***.155**	0.087	0.005	0.002	0.003	0.045	0.004	0.002	0.058
***.166**	0.099	0.012	0.040	0.048	0.138	0.005	0.023	0.154
***.167**	0.071	0.066	0.000	0.002	0.123	0.003	0.001	0.063
***.168**	0.010	0.013	0.000	0.000	0.042	0.000	0.001	0.014
***.169**	0.011	0.005	0.002	0.003	0.011	0.003	0.002	0.039
***.200**	0.010	0.002	0.000	0.000	0.012	0.000	0.001	0.011
***.201**	0.010	0.002	0.000	0.000	0.012	0.000	0.001	0.011
***.202**	0.010	0.002	0.000	0.000	0.012	0.000	0.001	0.011
***.203**	0.010	0.002	0.000	0.000	0.012	0.000	0.001	0.011
***.204**	0.010	0.002	0.000	0.000	0.012	0.000	0.001	0.011
***.205**	0.586	0.013	0.797	0.076	0.000	0.114	0.425	0.019
***.210**	0.586	0.013	0.797	0.076	0.000	0.114	0.425	0.019
***.211**	0.021	0.271	0.004	0.009	0.078	0.002	0.003	0.040
***.212**	0.024	0.003	0.000	0.000	0.031	0.000	0.001	0.030
***.213**	0.041	0.006	0.035	0.033	0.127	0.025	0.010	0.120
***.214**	0.071	0.003	0.092	0.663	0.002	0.992	0.054	0.011
***.215**	0.080	0.004	0.076	0.536	0.023	0.801	0.045	0.011
***.219**	0.071	0.003	0.092	0.663	0.002	0.992	0.054	0.011
***.220**	0.004	0.006	0.000	0.003	0.003	0.003	0.003	0.002
***.221**	0.071	0.003	0.092	0.663	0.002	0.992	0.054	0.011
***.231**	0.030	0.007	0.002	0.035	0.054	0.004	0.002	0.059
***.232**	0.075	0.007	0.002	0.013	0.058	0.019	0.003	0.085
***.233**	0.034	0.007	0.035	0.033	0.046	0.006	0.003	0.074
***.234**	0.071	0.003	0.092	0.663	0.002	0.992	0.054	0.011
***.236**	0.156	0.005	0.000	0.033	0.028	0.018	0.002	0.047
***.237**	0.004	0.006	0.000	0.003	0.003	0.003	0.003	0.002
***.238**	0.004	0.006	0.000	0.003	0.003	0.003	0.003	0.002
***.243**	0.077	0.014	0.019	0.030	0.147	0.021	0.034	0.184
***.244**	0.085	0.036	0.010	0.002	0.087	0.003	0.018	0.043
***.245**	0.013	0.021	0.001	0.008	0.044	0.002	0.001	0.025
***.246**	0.125	0.008	0.016	0.063	0.010	0.006	0.011	0.055
***.247**	0.012	0.016	0.008	0.009	0.005	0.012	0.010	0.013
***.248**	0.108	0.009	0.003	0.005	0.099	0.007	0.004	0.088