USER CLUSTERING AND TRAFFIC PREDICTION IN A TRUNKED RADIO SYSTEM

by

Hao Leo Chen
B.Eng., Zhejiang University, 1997

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Spring 2005

APPROVAL

Name:	Hao Leo Chen
Degree:	Master of Science
Title of thesis:	User Clustering and Traffic Prediction in a Trunked Radio System
Examining Committee:	Dr. Arthur Kirkpatrick Chair
	Dr. Ljiljana Trajković Senior Supervisor
	Dr. Martin Ester Supervisor
	Dr. Oliver Schulte Supervisor
	Dr. Qianping Gu Examiner
Date Approved:	February 14, 2005

SIMON FRASER UNIVERSITY

PARTIAL COPYRIGHT LICENCE

Abstract

Traditional statistical analysis of network data is often employed to determine traffic distribution, to summarize user's behavior patterns, or to predict future network traffic. Mining of network data may be used to discover hidden user groups, to detect payment fraud, or to identify network abnormalities. In our research we combine traditional traffic analysis with data mining technique. We analyze three months of continuous network log data from a deployed public safety trunked radio network. After data cleaning and traffic extraction, we identify clusters of talk groups by applying AutoClass tool and K-means algorithm on user's behavior patterns represented by the hourly number of calls. We propose a traffic prediction model by applying the classical SARIMA models on the clusters of users. The predicted network traffic agrees with the collected traffic data and the proposed cluster-based prediction approach performs well compared to the prediction based on the aggregate traffic.

To my parents and my wife!

"The **Tao** *is too great to be described by the name* **Tao**.
If it could be named so simply, it would not be the eternal **Tao**."

— *Tao Te Ching*, LAO TZU

Acknowledgments

I am deeply indebted to my senior supervisor, Dr. Ljiljana Trajković, for her patient support and guidance throughout this thesis. From her, I learned how to conduct research and how to write research papers. It was a great pleasure for me to conduct this thesis under her supervision.

I want to thank my supervisor Dr. Oliver Schulte for his valuable suggestions and inspiring discussions on Bayesian learning approaches. I want to thank my supervisor Dr. Martin Ester for his constructive comments about the clustering methods. I feel obliged to thank Dr. Qianping Gu who read my thesis carefully and gave me advice on thesis writing. The chair Dr. Arthur Kirkpatrick was of great help during my thesis defense. The defense would not have gone so smoothly without his coordination.

I also want to express my sincere appreciation to all the members in our CNL laboratory for their comments and suggestions on the thesis presentation, especially Kenny Shao and James Song for the valuable discussions on the topic of data analysis and prediction.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Analysis of traffic from operational wireless networks provides useful information about the network and users' behavior patterns. This information enables network operators to better understand the behavior of network users, to better use network resources, and, ultimately, to provide better quality of services.

Traffic prediction is important in assessing future network capacity requirements and in planning network development. Traditional prediction of network traffic usually considers aggregate traffic from individual network users. It also assumes a constant number of network users. This approach cannot easily adapt to a dynamic network environment where the number of users varies. An alternate approach that focuses on individual users is impractical in predicting the aggregate network traffic because of the high computational cost in cases where the network consists of thousands of users. Employing clustering technique for predicting aggregate network traffic bridges the apparent gap between these two approaches.

Data clustering may be used to identify and define customer groups in various business environments based on their purchasing patterns. In the telecommunication industry, clustering techniques may be used to identify traffic patterns, detect fraudulent activities, and discover users' mobility patterns. Network users are usually classified into user groups according to geographical location, organizational structure, payment plan, or behavior pattern. Patterns of users' behavior reflect the nature of

user activities and, as such, are inherently consistent and predictable. However, employing users' behavior patterns to classify user groups and to predict network traffic is non-trivial.

In this thesis, we analyze traffic data collected from a deployed network. We use hourly number of calls to represent individual user's calling behavior. We then predict network traffic based on the aggregate traffic and based on the identified clusters of users. Experimental results show that the cluster-based prediction produces results comparable to the traditional prediction of network traffic. The user cluster based traffic prediction approach may also address the computational cost and the dynamic number of users problems. An advantage of cluster-based prediction is that it may be used for predictions in networks with variable number of users. This approach provides a balance between a micro and a macro view of a network.

This thesis includes additional five chapters.

Chapter 2 begins with a brief introduction to the network and the traffic data that we analyzed. It is followed by the description of data preprocessing, data extraction and the results.

In Chapter 3, various statistical analysis routines have been applied to the traffic data on three levels: network, agency, and talk group levels. The analysis results include plots and basic statistical measures (maximum, minimum, mean value, and variance).

In Chapter 4, we discuss the general clustering techniques and principles. We apply the AutoClass clustering tool and K-means algorithm to classify talk groups into clusters based on their calling activities. We also compare the the clustering results of AutoClass and K-means.

In Chapter 5, we present the Seasonal Autoregressive Integrated Moving Average (SARIMA) time series prediction model. We discuss the model selection method and present the prediction results of the network traffic. We conclude with a comparison of the prediction results of cluster-based models and models based on aggregate traffic.

We conclude the thesis with Chapter 6. A short summary of experiences that we gained is given and the future work is addressed.

Appendics include additional database tables, SQL scripts, R scripts, and snippets

of AutoClass model and report files. Experimental results of conditional dependency analysis of the traffic data using Bayesian network are also presented.

Chapter 2

Data preparation

The traffic data analyzed in this thesis were obtained from E-Comm [1]. In this Chapter, we first introduce the architecture of the E-Comm network and the underlying technology. We also examine the database schema and describe the procedure for data cleaning and the traffic data extraction.

2.1 E-Comm network

2.1.1 E-Comm network structure overview

E-Comm is the regional emergency communications center for Southwest British Columbia, Canada. It provides emergency dispatch/communication services for a number of police and fire departments in the Greater Vancouver Regional District (GVRD), the Sunshine Coast Regional District, and the Whistler/Pemberton area. E-Comm serves sixteen agencies such as Royal Canadian Mounted Police (RCMP), fire and rescue, local police departments, ambulance, and industrial customers such as BC Translink [2]. Each agency has a number of affiliated talk groups and the entire network serves 617 talk groups. Figure 2.1 presents a rough geographical coverage of the E-Comm network.

Before the establishment of E-Comm, ambulance, fire, and police agencies could not communicate with each other effectively because they used separate radio systems.

The deployment of the E-Comm network in 1999 provided an integrated shared communications infrastructure to various emergence service agencies. It enables the cross communication between various agencies and municipalities.

The E-Comm network employs Enhanced Digital Access Communications System (EDACS), developed by M/A-COM [3] (formerly Comnet-Ericsson) in 1988. EDACS system is a group-oriented communication system that allows groups of users to communicate with each other regardless of their physical locations. The main advantages of this approach are improved coordination, efficient exchange of information, and efficient resource usage.

The E-Comm network consists of 11 cells. Each cell covers one or more municipalities, such as Vancouver, Richmond, and Burnaby. Identical radio frequencies are transmitted within one cell using multiple repeaters. This is known as simulcast. The basic talking unit in the trunked radio network is a talk group: a group of individual users working and communicating with each other to accomplish certain tasks. Although the E-Comm network is capable of both voice and data transmissions, we analyze only voice traffic because it accounts for more than 99% of the network traffic.

2.1.2 E-Comm network terminology

We explain briefly the following network terms:

System/Cell: A trunked radio network is divided into smaller areas in order to reuse the radio frequencies and to increase the network capacities. One system represents one service area and a cell is the synonymous of a system. One system could serve one or more municipalities, based on the frequencies availability and geographical connection. A unique system id is associated with each system. Within a system/cell, the radio signal is transmitted using the same range of frequencies.

Channel: A channel is a small range of radio frequencies or a time slot. Various numbers of channels are assigned in each system based on the traffic throughput

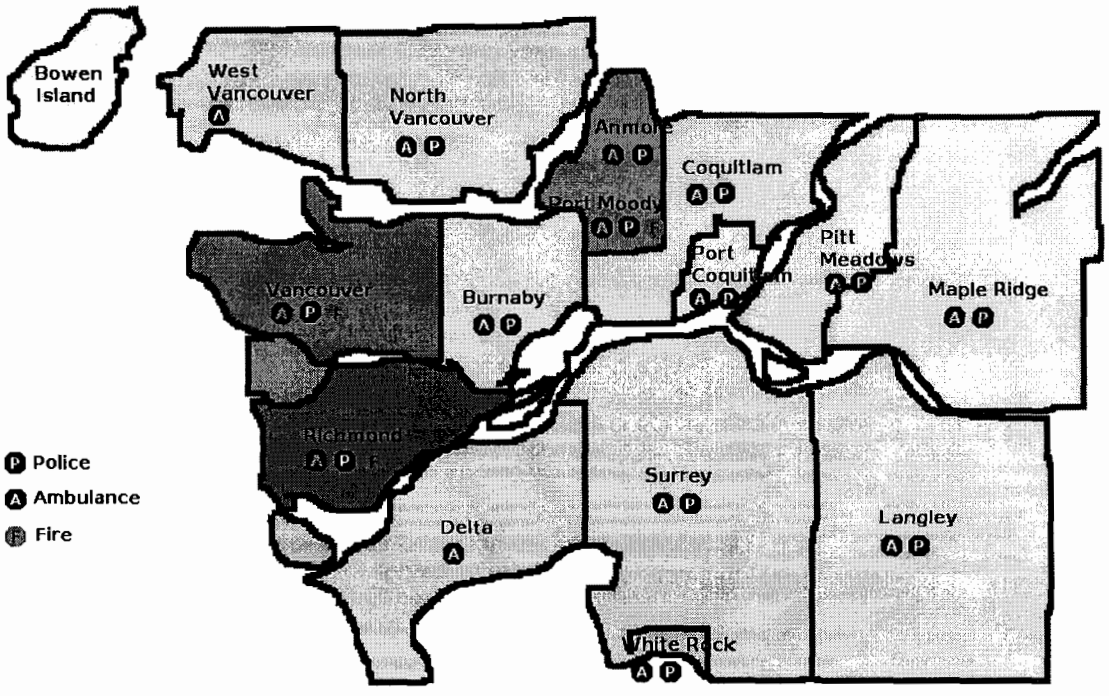

Figure 2.1: E-Comm network coverage in the Greater Vancouver Regional District.

and the system needs. Two types of radio channel are used in EDACS: control and traffic channels. There is one control channel in each cell, while the remaining channels are used as traffic channels. The control channel is used to send protocol messages between radios and the base station equipment for controlling the operation of the system. Traffic channels are used to transmit the voice or data messages between radios or between radios and the base stations.

Group Call: Group call is the typical call made in a trunked radio system. A group is a set of users who need to communicate regularly in order to accomplish certain tasks. For example, within a single city-wide system, the North and South fire services may each have one talk group, while the police may be subdivided into several talk groups. A user only needs to press the push-to-talk (PTT) button on the radio device to initiate a group call. All users belonging to

the same talk group will hear the communications in the group call irrespective of their physical locations. Most EDACS network operators have observed that more than 85% of calls are group calls [4].

Simulcast: In the E-Comm network, simulcast is used in a single cell. Within a cell, identical radio frequencies are transmitted simultaneously between two or more base station sites in order to improve signal strength and increase coverage.

Multi-System Call: It represents a single group call involving more than one system/cell. A user may initiate a group call without knowing the physical location of the group members. When all members of the talk group reside within one system, the group call is a single-system call occupying only one traffic channel in the system. However, when group members are distributed over multiple systems, the group call becomes a multi-system call that occupies one traffic channel in each system. Hence, the major difference between a multi-system call and a single-system call is that the first occupies additional channels and consumes more system resources. In the collected traffic data, more than 55 % of group calls are multi-system calls.

2.2 Network traffic Data

The traffic data received from E-Comm contains event log tables recording the activities occurred in the network. They are aggregated from the distributed database of the network management systems.

2.2.1 Database setup

Analyzed data records span from 2003-03-01 00:00:00 to 2003-05-31 23:59:59 continuously. The database size is $\sim$ 6G bytes, with 44,786,489 records for the 92 days of data. It consists of 92 event log tables, each containing one data's events generated in the network, such as the call establishment, call drop, and emergency call events. Its sheer volume was one of the main difficulties in our data analysis. For efficiency,

we converted the data from the MS Access format to plain text files and imported the records into the MySQL [5] database server under Linux platform.

2.2.2 Event log database schema

The complete twenty-six data fields in the event log table are:

1. Event_UTC_At: the timestamp of the event (granularity is 3 ms).
2. Duration$ms: the duration of the event in ms (granularity is 10 ms).
3. Network_Id: the identification of the network (constant in the database).
4. Node_Id: the identification of the network node (not populated).
5. System_Id: the identification of the system/cell involved in the event, ranging from 1 to 11.
6. Channel_Id: the identification of the channel involved in the event.
7. Slot_Id: this field is not populated in the database.
8. Caller: also known as LID (Logic ID). It is the caller's id, ranging from 1 to 16,000. The first 2,000 LIDs are assigned to either talk groups or individual users. The remaining LIDs are assigned to talk groups only.
9. Callee: the callee's id in the event, having the same value range as Caller.
10. Call_Type: the type of the call, such as group call, emergency call, and individual call.
11. Call_State: the state of the call event, such as assign channel, drop, and queue.
12. Call_Direction: the direction of the call (meaning unknown).
13. Voice_Call: a flag indicating a voice call.
14. Digital_Call: a flag indicating a digital call.

15. Interconnect_Call: a flag indicating a call interconnecting the EDACS and the Public Switched Telephone Network (PSTN).

16. Multi_System_Call: a flag indicating a multi-system call. It is only set in the event of call drop.

17. Confirmed_Call: a flag of the call. A call is a confirmed call when every member of a talk group has to confirm the call before the conversation begins.

18. Msg_Trunked_Call: a flag of the call (meaning unknown).

19. Preempt_Call: a flag of the call. A preempt_call has higher queue priority.

20. Primary_Call: a flag of the call (meaning unknown).

21. Queue_Depth: the depth of the current system queue at the event moment. It may be used to investigate the block rate of the system.

22. Queue_Pri: the priority number of the call in queue.

23. MCP (Multi-Channel-Partition): the partition number of channels (not populated).

24. Caller_Bill: set to 1 if the call is billable to the caller (not used in the current system).

25. Callee_Bill: set to 1 if the is billable to the callee (not used in the current system).

26. Reason_Code: the error reason code number, providing additional information if any error occurs during the call.

2.2.3 Topic of interest

Two open questions, emanating from the discussions with E-Comm staff and the analysis of database, are of particular interest to our analysis: the precise measurement of network usage per agency and the traffic forecast based on user's behavior patterns.

A. Precise measurement of network usage

The current billing policy for agencies in the E-Comm network is based on the geographic coverage and the approximate calling traffic volume of each agency. Traffic volume factors are further broken down into the number of radios, radio traffic, and user population. Shared radio infrastructure costs are allocated based on the coverage area, number of radios, radio traffic, and population.

Presently, there is no precise measuring method for the traffic generated by each individual user/talk group. The current database is an event log database, recording every activity that occurred in the network, such as the call establishment, call drop, and emergency calls. One group call may be recorded twice in the event log, as it generates both call assignment and call drop events. One single multi-system call involving several systems generates multiple entries in the database. Based solely on the raw traffic logs, the calculation of network resources used by one agency is inaccurate. Therefore, the traffic generated by agencies is not calculated directly from the event database. It is, instead, based on an assumed mean value of call duration, the coverage of cells by the agency, the number of radios possessed by the agency, and the number of records corresponding to the agency. It is unable to identify the number of calls made by each agency, the average/maximum number of systems involved in calls for each agency, and the network usage for each talk group. A sample of the data is shown in Section 2.4.

B. Traffic forecast based on user's behavior patterns

Users' behavior patterns in the trunked radio networks are different from the traditional telephone networks. Group calls involve more than two users, while traditional telephone calls connect only two persons. Furthermore, since the E-Comm network mainly serves emergency communications, the uncertainty of emergencies implies different behavior patterns of network users from users of ordinary telephone network. In addition, different agencies may have different behavior patterns. For example, the ambulance service may have different peak hours from the RCMP, while the fire department often dispatch group of firefighters to the accident sites together with the police groups.

Understanding user's behavior could help improve user satisfaction and be beneficial for the network optimization and the planning for network expansion. For example, if a police department plans to increase its manpower by increasing the patrol groups and using more radios. A reasonable assumption is that the new groups have similar behavior patterns to the existing users. New patrol groups may be classified into certain existing user groups based on their behavior patterns. Considering the number of new members in the user groups, we may forecast the network traffic based on the existing user's behavior patterns, thus to make better assessment on the network capacity.

2.3 Data preprocessing

The main difficulty in analyzing the network log data is the sheer volume of data. Data preprocessing is the fundamental and mandatory step for data analysis. It is used to clean the database and filter the outliers and redundant records. The current database includes: surplus data fields with useless entries, obscure data records, and inconsistent data fields. The goals of the data preprocessing step are to remove useless information and to remove the outliers. They are accomplished by acquiring the necessary domain knowledge from the system documentation and via interviews with the E-Comm staff. The preprocessing procedure is composed of database shrinking and cleaning.

2.3.1 Database shrinking

Not all data fields are useful to our analysis. Certain fields are not populated in the database (Node_Id and Slot_Id fields), while others have identical value or are unrelated to our research (Network_Id, Caller_Bill, and Callee_Bill). We are only interested in fields that could capture the user's behavior and network traffic. Thus, the step is to remove these unpopulated, identical, or unrelated fields from the database, such as the Digital_Call, Interconnect_Call, Confirmed_Call, Primary_Call, Caller_Bill, Callee_Bill, and Reason_Code fields.

From the twenty-six original fields in the database, nine fields are of particular interest to our analysis: *1*) Event_UTC_At, *2*) Duration$ms, *3*) System_Id, *4*) Channel_Id, *5*) Caller, *6*) Callee, *7*) Call_Type, *8*) Call_State, and *9*) Multi_System_Call.

2.3.2 Database cleaning

After reducing the database dimension to nine, we removed redundant records, such as records having call_type = 100 or records with duration = 0. Records with call_state = 1, which implies the *call drop* event, are redundant because each *call drop* event already has a corresponding *call assignment* event in the database. (Note that the reverse is not true.) Records with channel_id = 0 should also be removed as well because the channel id 0 represents the control channel whose traffic we have not considered. We keep the the records with call_type = 0, 1, 2, or 10, representing group call, individual call, emergency call, and start-emergency-call, respectively. The complete call_type table is given in Appendix A.

The result of data preprocessing step is a smaller and cleaner database. The number of records in each data table of original and cleaned databases are compared in Table 2.1. Approximately 55% records have been removed from the original database after preprocessing. Furthermore, due to the effect of the dimension reduction, the total size of the database has been reduced to only 19% of the original size.

2.4 Data extraction

The extraction of the network traffic may solve the first open question of imprecise traffic measurement, as described in Section 2.2.3. A sample of the cleaned database table is shown in Table 2.2. If a call is a multi-system call involving several systems, several records (one for each involved system) are created to represent this call in the original event log database. For example, based on the caller, callee, and duration information, records 1 and 6 represent one group call from caller 13905 to callee 401, involving systems 1 and 7 and lasting $\sim$ 1350 ms. Records 29, 31, 37, and 38 represent a group call from caller 13233 to callee 249, involving systems 2, 1, 7, and 6. Thus, the

Date	Original	Cleaned	Date	Original	Cleaned	Date	Original	Cleaned
2003/03/01	466,862	204,357	2003/04/01	578,834	260,752	2003/05/01	535,919	240,046
2003/03/02	415,715	184,973	2003/04/02	609,686	275,575	2003/05/02	536,092	240,585
2003/03/03	406,072	182,311	2003/04/03	503,666	225,041	2003/05/03	413,171	184,998
2003/03/04	464,534	207,016	2003/04/04	491,225	221,373	2003/05/04	393,421	176,878
2003/03/05	585,561	264,226	2003/04/05	479,043	215,979	2003/05/05	362,118	161,104
2003/03/06	605,987	271,514	2003/04/06	360,661	159,867	2003/05/06	463,040	202,153
2003/03/07	546,230	247,902	2003/04/07	423,915	189,111	2003/05/07	542,724	242,997
2003/03/08	513,459	233,982	2003/04/08	507,364	227,196	2003/05/08	559,787	248,127
2003/03/09	442,662	201,146	2003/04/09	563,334	252,753	2003/05/09	556,419	250,072
2003/03/10	419,570	186,201	2003/04/10	518,096	232,572	2003/05/10	471,745	213,051
2003/03/11	504,981	225,604	2003/04/11	501,114	224,941	2003/05/11	415,702	187,786
2003/03/12	516,306	233,140	2003/04/12	482,866	215,426	2003/05/12	381,057	170,031
2003/03/13	561,253	255,840	2003/04/13	406,548	180,903	2003/05/13	484,477	217,803
2003/03/14	550,732	248,828	2003/04/14	347,400	151,802	2003/05/14	530,492	236,520
2003/03/15	581,932	266,329	2003/04/15	429,918	190,384	2003/05/15	550,407	246,539
2003/03/16	519,893	237,804	2003/04/16	513,713	229,653	2003/05/16	514,825	231,259
2003/03/17	470,046	213,815	2003/04/17	515,302	231,966	2003/05/17	454,208	202,995
2003/03/18	583,717	267,938	2003/04/18	421,623	189,158	2003/05/18	448,726	202,213
2003/03/19	544,893	249,766	2003/04/19	414,045	183,778	2003/05/19	406,458	182,730
2003/03/20	575,978	262,049	2003/04/20	392,821	175,380	2003/05/20	421,129	187,064
2003/03/21	548,872	252,185	2003/04/21	325,268	143,316	2003/05/21	525,547	235,586
2003/03/22	525,830	240,821	2003/04/22	367,287	161,285	2003/05/22	574,971	258,432
2003/03/23	534,699	244,510	2003/04/23	428,419	187,621	2003/05/23	549,397	244,869
2003/03/24	475,808	215,582	2003/04/24	464,451	208,512	2003/05/24	502,278	225,573
2003/03/25	514,570	233,283	2003/04/25	471,794	211,731	2003/05/25	436,931	196,311
2003/03/26	589,203	267,982	2003/04/26	449,725	202,244	2003/05/26	394,320	176,583
2003/03/27	608,074	276,281	2003/04/27	369,049	165,248	2003/05/27	490,976	220,099
2003/03/28	503,455	227,615	2003/04/28	372,067	164,094	2003/05/28	517,567	232,240
2003/03/29	542,443	248,825	2003/04/29	464,529	206,596	2003/05/29	551,566	248,393
2003/03/30	446,921	203,254	2003/04/30	547,473	245,293	2003/05/30	556,295	250,757
2003/03/31	446,174	202,423				2003/05/31	511,056	229,872
Total:	16,012,432	7,257,502	Total:	13,721,236	6,129,550	Total:	15,052,821	6,743,666

Table 2.1: Number of records per day: original vs. cleaned database.

network operator cannot count the number of group calls made by a certain talk group or agency merely based on the original multiple entries. Furthermore, it is impossible to find the number of multi-system calls and the average number of systems in a multi-system call.

We explore the relationships of fields among similar records and find that, within a certain range, multiple records with identical caller id and callee id and similar call duration fields might represent one single group call in the database. Caused by the transmission latency and glitch in the distributed database system, the call duration is sometimes inconsistent. For example, records 1 (1340 ms) and 6 (1350 ms) in Table 2.2, have 10 ms difference in call duration field although they represent one single group call. Experimental results indicate that 50 ms difference in call duration is an acceptable choice when combining the multiple records (compared to 20 ms, 30 ms, or 100 ms).

The algorithm for extracting and combining the traffic data from the cleaned database is shown in Figure 2.2. It is implemented by Perl. A sample of the results of the traffic extraction from Table 2.2 is shown in Table 2.3. Record 1 in Table 2.3 is the combination of records 1 and 6 in Table 2.2, while record 7 corresponds to the combination of records 29, 31, 37, and 38 in Table 2.2.

2.5 Summary

In this Chapter, we provided a short presentation of the trunked radio systems and infrastructure of the E-Comm network. The importance of the data preprocessing have been illustrated using data shown in Table 2.1. We described the traffic data schema, data preprocessing, and traffic extraction. The data extraction process was used to extract traffic data by combining multiple entries of one group call into a single record. The result of data preprocessing, together with data extraction, is a clean and neat database with $\sim$ 81 % fewer records. A comparison of the number of records in original, cleaned, and extracted database is shown in Figure 2.3. The generated traffic data was used for further data analysis, clustering, and prediction.

No.	Date	Time (hh:mm:ss)	(ms)	Call duration	System id	Channel id	Caller	Callee	Call type	Call state	Multi-system call
1	2003-03-01	00:00:00	30	1340	1	12	13905	401	0	0	0
3	2003-03-01	00:00:00	179	3330	7	1	14663	249	0	0	0
4	2003-03-01	00:00:00	259	3330	6	3	14663	249	0	0	0
6	2003-03-01	00:00:00	489	1350	7	4	13905	401	0	0	0
7	2003-03-01	00:00:00	590	2990	6	4	4266	1443	0	0	0
10	2003-03-01	00:00:01	150	2840	1	2	6109	1817	0	0	0
22	2003-03-01	00:00:03	119	6110	9	6	15202	465	0	0	0
23	2003-03-01	00:00:03	119	6100	10	9	15202	465	0	0	0
24	2003-03-01	00:00:03	149	2980	2	6	16068	673	0	0	0
25	2003-03-01	00:00:03	370	6110	6	5	15202	465	0	0	0
29	2003-03-01	00:00:03	620	7550	2	7	13233	249	0	0	0
30	2003-03-01	00:00:03	700	2980	9	7	16068	673	0	0	0
31	2003-03-01	00:00:03	760	7560	1	3	13233	249	0	0	0
32	2003-03-01	00:00:03	830	1580	2	8	13333	245	0	0	0
33	2003-03-01	00:00:03	879	5790	7	5	12183	201	0	0	0
34	2003-03-01	00:00:03	970	1590	1	8	13333	245	0	0	0
36	2003-03-01	00:00:04	150	2970	1	9	6009	1817	0	0	0
37	2003-03-01	00:00:04	260	7560	7	6	13233	249	0	0	0
38	2003-03-01	00:00:04	340	7560	6	6	13233	249	0	0	0
41	2003-03-01	00:00:04	980	3810	1	12	13906	403	0	0	0
42	2003-03-01	00:00:05	169	2410	1	2	15906	401	0	0	0
46	2003-03-01	00:00:05	449	3800	7	7	13906	403	0	0	0
49	2003-03-01	00:00:05	679	2400	7	1	15906	401	0	0	0
50	2003-03-01	00:00:05	979	2160	6	7	4831	1443	0	0	0
53	2003-03-01	00:00:06	900	1300	2	9	9701	673	0	0	0
56	2003-03-01	00:00:07	409	1300	9	8	9701	673	0	0	0
60	2003-03-01	00:00:08	149	880	1	4	7003	786	0	0	0

Table 2.2: A sample of cleaned data.

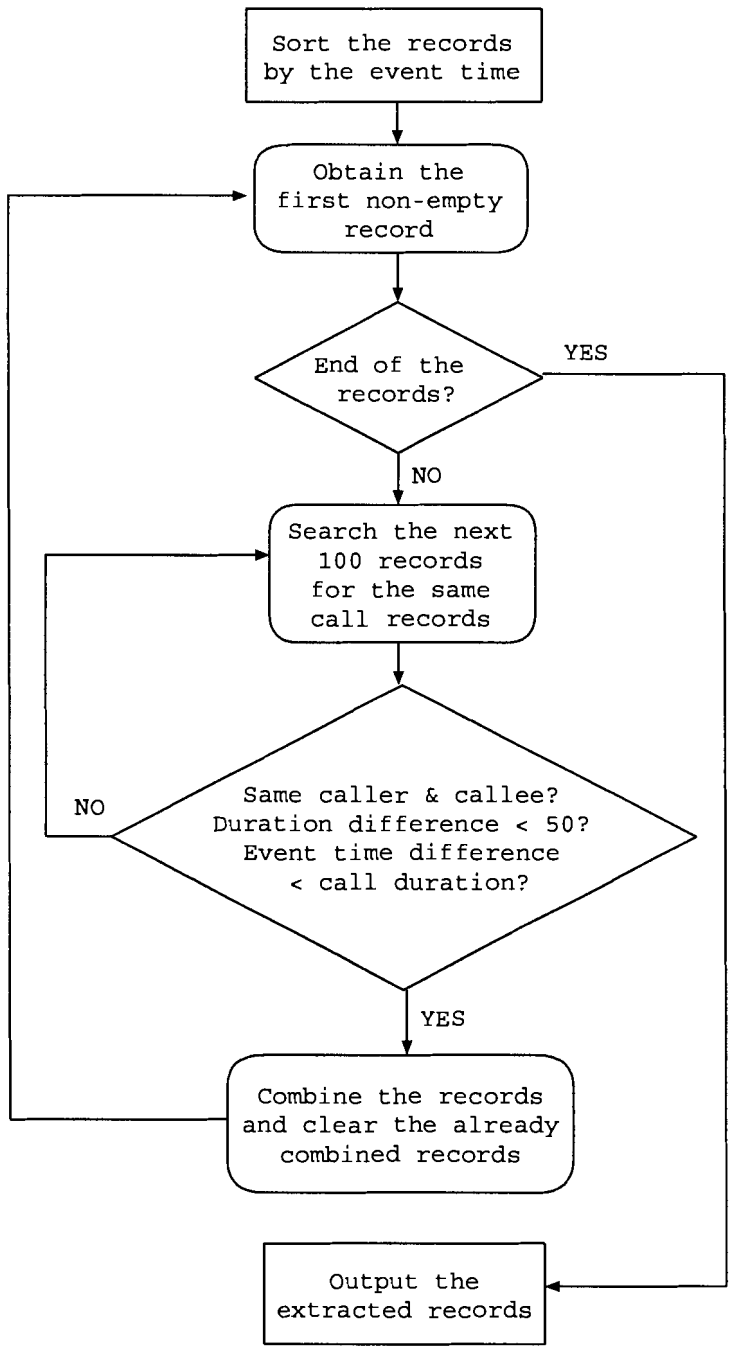

Figure 2.2: Algorithm for extracting traffic data.

No.	Date	Time (hh:mm:ss)	(ms)	Call duration	Caller id	Callee id	Call type	Call state	Multi system call	Number of systems	List of system(s)
1	2003-03-01	00:00:00	30	1340	13905	401	0	0	0	2	1, 7
2	2003-03-01	00:00:00	179	3330	14663	249	0	0	0	2	7, 6
3	2003-03-01	00:00:00	590	2990	4266	1443	0	0	0	1	6
4	2003-03-01	00:00:01	150	2840	6109	1817	0	0	0	1	1
5	2003-03-01	00:00:03	119	6110	15202	465	0	0	0	3	9, 10, 6
6	2003-03-01	00:00:03	149	2980	16068	673	0	0	0	2	2, 9
7	2003-03-01	00:00:03	620	7550	13233	249	0	0	0	4	2, 1, 7, 6
8	2003-03-01	00:00:03	830	1580	13333	245	0	0	0	2	2, 1
9	2003-03-01	00:00:03	879	5790	12183	201	0	0	0	1	7
10	2003-03-01	00:00:04	150	2970	6009	1817	0	0	0	1	1
11	2003-03-01	00:00:04	980	3810	13906	403	0	0	0	2	1, 7
12	2003-03-01	00:00:05	169	2410	15906	401	0	0	0	2	1, 7
13	2003-03-01	00:00:05	979	2160	4831	1443	0	0	0	1	6
14	2003-03-01	00:00:06	900	1300	9701	673	0	0	0	2	2, 9
15	2003-03-01	00:00:08	149	880	7003	786	0	0	0	1	1
16	2003-03-01	00:00:10	239	3420	4266	1443	0	0	0	1	6
17	2003-03-01	00:00:10	359	3930	15895	201	0	0	0	1	7
18	2003-03-01	00:00:12	450	1820	12277	417	0	0	0	3	2, 1, 5
19	2003-03-01	00:00:12	870	2810	13906	403	0	0	0	2	1, 7
20	2003-03-01	00:00:13	49	11720	14663	249	0	0	0	4	2, 1, 7, 6

Table 2.3: A sample of extracted traffic data.

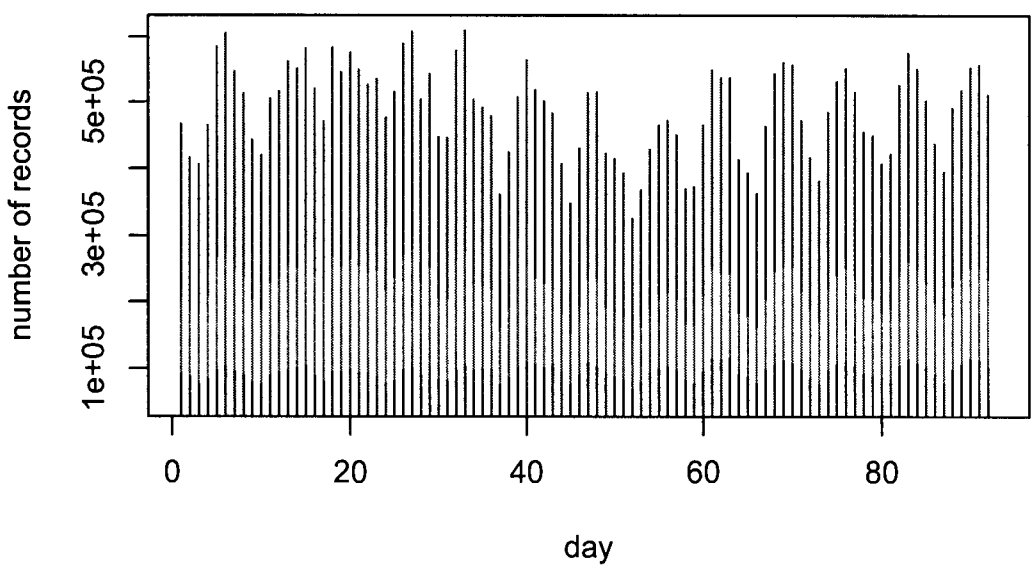

Figure 2.3: Comparison of number of records in original, cleaned, and extracted databases.

Chapter 3

Data analysis

Statistical analysis on the extracted traffic trace usually includes finding maximum, minimum, mean value, measure of variation, data plots, and histograms. Data network traffic may be measured in terms of the number of packets, number of connections, or number of bytes transmitted. Similarly, the traffic of voice networks may be measured by the number of calls and the call duration. We use the hourly number of calls to analyze the E-Comm network traffic on three levels: aggregate network, agency, and talk group level.

3.1 Analysis on Network level

On the network level, the traffic is the aggregation of all users' traffic. The analysis of network-level traffic provides overview of the network usage. The aggregate traffic of the entire network, in terms of hourly and daily number of calls, is shown in Figure 3.1. The upper and lower dotted lines indicate the maximum and the minimum number of calls, respectively. The middle dashed line is the mean value.

Figure 3.1 demonstrates the inherent cyclic patterns of the network traffic. We check the periodic patterns by applying the Fast Fourier Transform (FFT) on the network data to find the highest frequency in the hourly and the daily number of calls. The FFT reveals the high frequency components at 24 for the hourly number of calls and at 7 for the daily number of calls, as shown in Figure 3.2. We conclude that

the network traffic exhibits daily (24 hours) and weekly (168 hours) cycles in terms of number of calls. Similar daily and weekly cyclic traffic patterns of various networks have been observed in the literature [6], [7], [8].

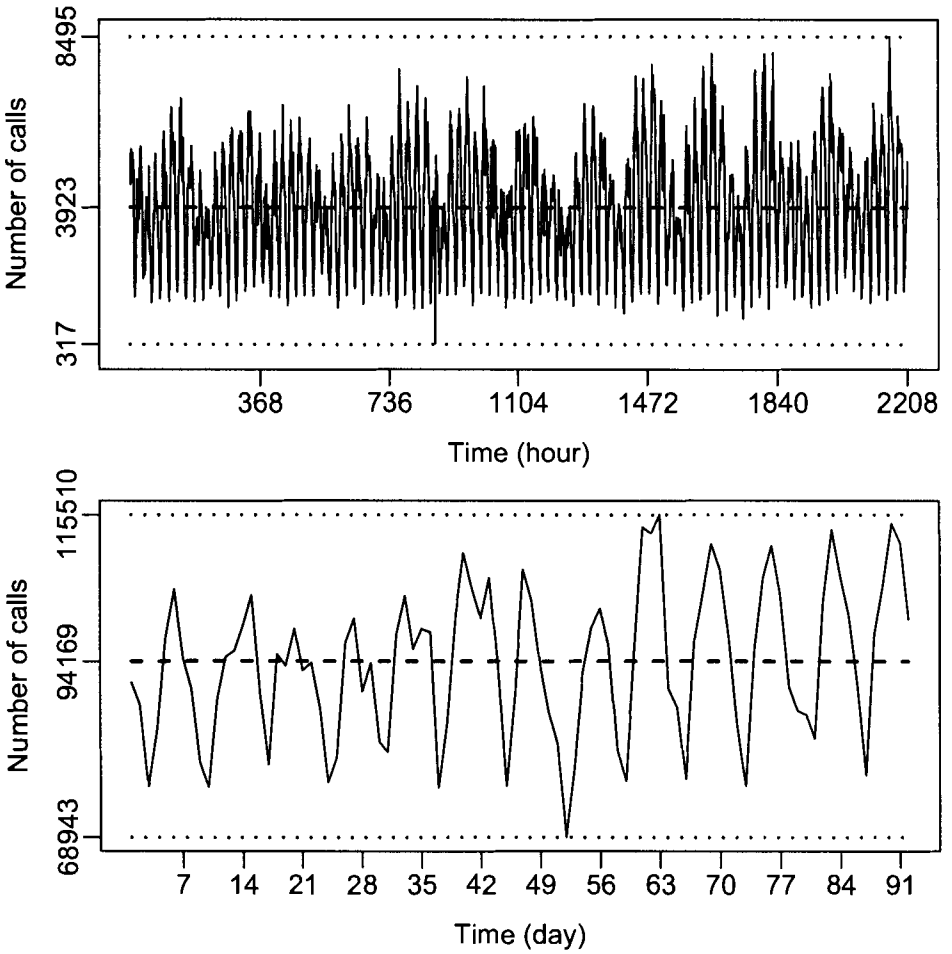

Figure 3.1: Statistical analysis of hourly (top) and daily (bottom) number of calls.

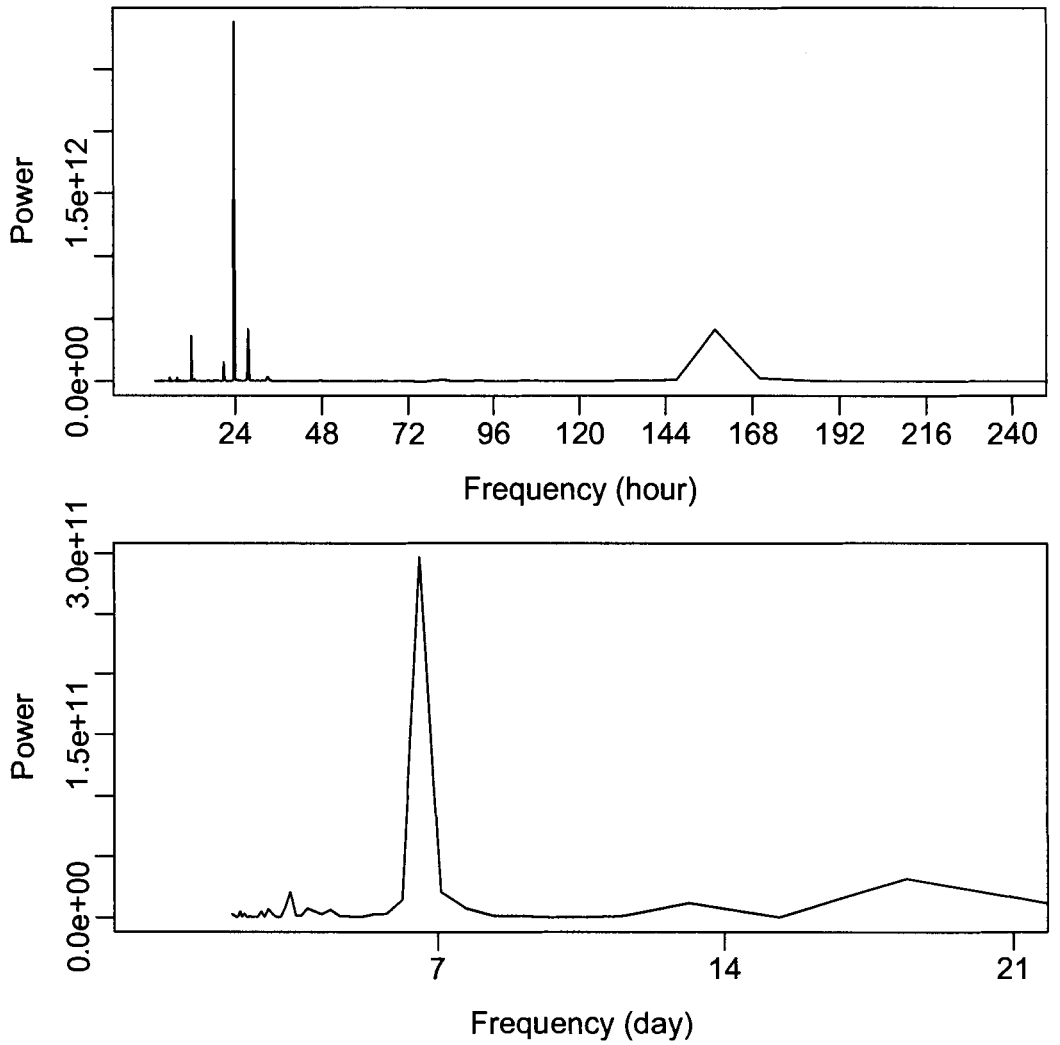

Figure 3.2: Fast Fourier Transform (FFT) analysis on hourly (top) and daily (bottom) number of calls. The high frequency components at 24 (top) and 7 (bottom) indicate that the network traffic exhibits daily (24 hours) and weekly (168 hours) cycles, respectively.

3.2 Analysis on agency level

Network users belong to various agencies such as RCMP, police, ambulance, and fire department. The study of agency behavior may help network operators identify the aggregate traffic patterns in the organizational usage of network resources. Agency names are eliminated to protect their privacy. Instead, we use agency id to identify the agency structure for talk groups.

The agency id in the E-Comm network ranges from 0 to 15. The agency id 0 represents unknown or corrupted agency group information of users. The network usage statistic data of each agency is summarized in Table 3.1. The rows are sorted in ascending order by the number of calls made by each agency. 92% of calls are made by three agencies with id 10, 2, 5, while the remaining 13 agencies account for only 8% of the calls. The average call duration ranges from 2.3 to 5.9 seconds. We also observed that more than 55% of calls in the network are multi-system calls. Beside the hourly number of calls, call duration is another major factor affecting the network resource usage. In order to measure how long and how many channels have been occupied by a call in the network, we define the network resource usage for a call as:

$$Network\ resource = Call\ duration * Number\ of\ systems.$$

Three different aspects of agency traffic are shown in Figure 3.3. We use different symbol in the figure to represent different agency. The top plot is the daily number of calls for each agency. The middle plot is the daily average call duration of each agency. The bottom plot represents the average number of systems involved in the calls of each agency. The daily average call duration is relatively constant for agencies, while the daily number of calls shows large variations among agencies.

3.3 Analysis on talk group level

The basic talking unit in the E-Comm network is a talk group. This is the finest unit for our analysis. Traffic analysis on the agency level is too coarse to capture the behavior of small talking units in the network. Even though each talk group belongs

Agency id	Number of calls	Average duration (ms)	Number of calls (%)	Number of multi-system calls	Number of multi-system calls (%)
20	22	2,329	0.00%	0	0.00%
15	37	2,239	0.00%	8	21.62%
8	129	4,230	0.00%	127	98.44%
7	2,963	4,080	0.03%	606	20.45%
14	5,523	3,279	0.06%	248	4.49%
0	10,037	3,278	0.11%	6,368	63.44%
13	13,590	5,986	0.15%	0	0.00%
6	39,363	3,871	0.45%	1,427	3.62%
11	58,622	3,861	0.67%	2,220	3.78%
4	82,482	3,175	0.95%	11,862	14.38%
1	91,417	3,857	1.05%	13,567	14.84%
3	117,289	4,024	1.35%	39,507	33.68%
21	282,907	3,480	3.26%	180,792	63.90%
10	950,725	3,438	10.97%	722,822	76.02%
2	2,527,096	3,853	29.16%	917,037	36.28%
5	4,481,384	3,838	51.72%	3,193,948	71.27%
Sum	8,663,586	3,772	100%	5,090,539	58.76%

Table 3.1: Agency network usage.

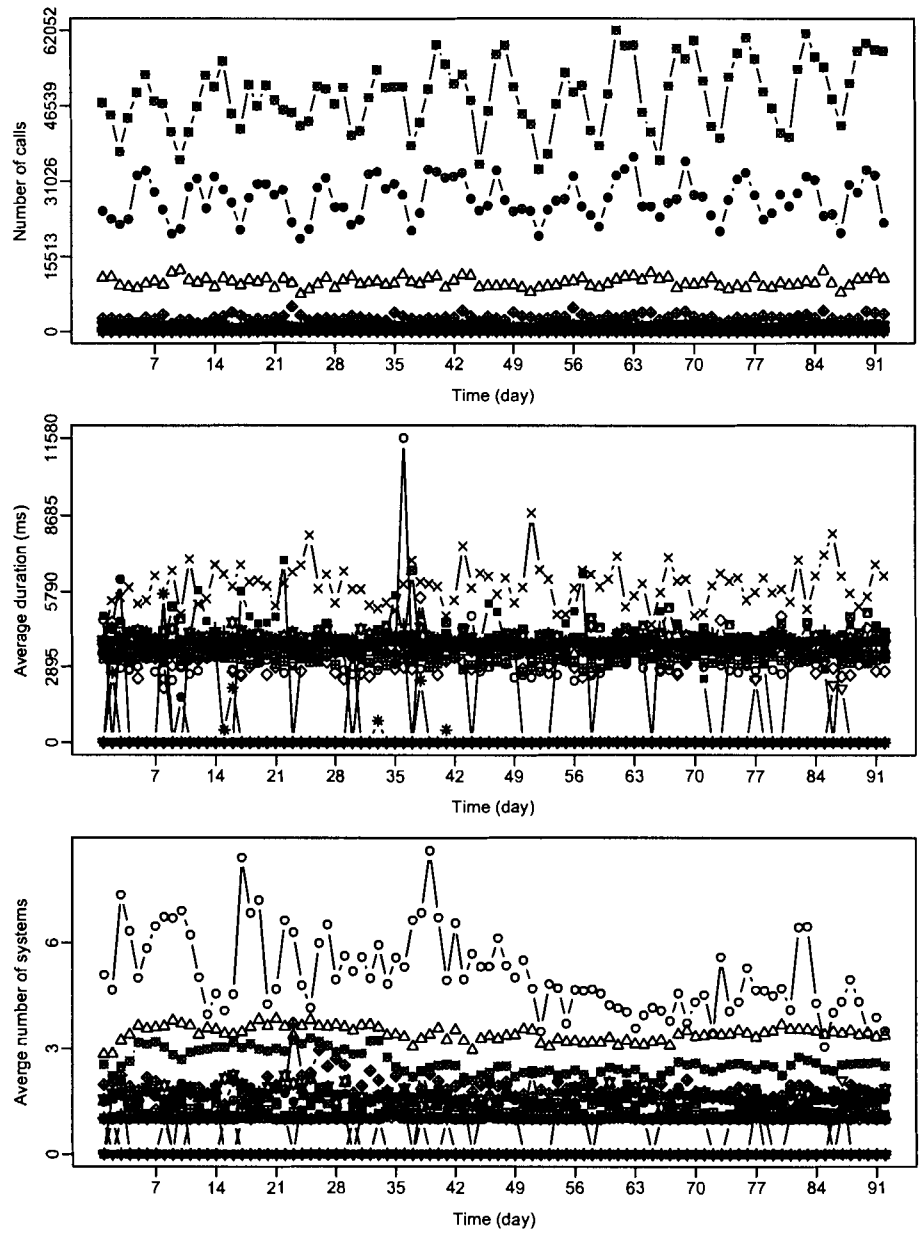

Figure 3.3: Traffic analysis by agencies (Top: the daily number of calls for each agency. Middle: the daily average call duration of each agency. Bottom: the average number of systems involved in the calls of each agency.

to a certain agency, the organizational structure does not necessarily imply similar usage patterns. Talk groups belonging to different agencies may have similar behavior, while talk groups within the same agency may have different behavior patterns.

A sample of talk groups' behavior patterns is shown in Table 3.2. The behavior patterns include average resources, average duration, and average number of systems involved in group calls. The talk groups are sorted in descending order of the total number of calls during the 92 days. The average call duration exhibits a relatively constant pattern with mean value of 3,621.50 ms and standard variance of 397 ms. To the contrary, the average number of systems involved in calls is quite different. For example, the members of talk group 1809 are usually distributed across more than 4 systems, while the members of talk group 785 often reside in one system when making calls. Accordingly, the number of systems engaged in a call greatly affects the network resource usage.

3.4 Summary

The preliminary statistical analysis of traffic data at different levels shows the diversity and complexity of network user's (talk group's) behavior. User's behavior exhibits patterns that may be used to categorize talk groups. We are particularly interested in building clusters of talk groups based on their behavior patterns. This topic is addressed in Chapter 4.

Talk group id	Agency id	Number of calls	Average resources used	Average duration (ms)	Average number of systems
801	2	461,128	4,756.23	3,489.76	1.35
817	2	382,065	6,953.52	3,484.94	1.96
465	5	363,138	11,421.93	3,640.22	3.11
785	2	354,324	4,390.52	3,532.91	1.23
1817	10	312,131	5,259.29	3,638.01	1.44
497	5	303,991	10,758.75	3,547.97	3.00
401	5	303,948	8,256.77	3,416.61	2.40
833	2	303,854	6,180.41	3,678.45	1.66
1801	10	294,687	15,968.30	3,836.57	4.14
1809	10	278,872	17954.45	3,844.45	4.62
481	5	278,634	11,805.02	3,240.11	3.61
471	5	276,404	10,548.83	3,543.19	2.95
673	1	260,813	6,392.49	3,427.50	1.85
449	5	258,019	9,159.69	3,711.86	2.43
433	5	226,492	8,558.56	3,695.66	2.30
786	2	225,612	4,939.96	4,653.82	1.06
418	5	207,583	6,868.27	3,259.42	2.08
289	5	159,649	16,216.39	3,473.22	4.61
249	5	145,875	23,454.87	4,939.05	4.73
...		...		...	

Table 3.2: Sample of the resource consumption for various talk groups.

Chapter 4

Data clustering

Data mining employs a variety of data analysis tools to discover hidden patterns and relationships in data sets. Clustering analysis, with its various objectives, groups or segments a collection of objects into subsets or clusters so that objects within one cluster are more "close" to each other than objects in distinct clusters. It attempts to find natural groups of components (or data) based on certain similarities. It is one of the powerful tools in data mining, with applications in a variety of fields including consumer data analysis, DNA classification, image processing, and vector quantization.

In this Chapter, we first describe the data used for the clustering analysis. We then introduce the AutoClass [9] tool and K-means [10] algorithm. The results of clustering and the comparison are also presented.

4.1 Data representing user's behavior

An object can be described by a set of measurements or by its relations to other objects. Customers' purchasing behavior may be characterized by shopping lists with the type and quantity of the commodities bought. Network users' behavior may be measured as the time of calls, the average length of the call, or the number of calls made in a certain period of time. Telecommunication companies often use call inter-arrival time and call holding time to calculate the blocking rate and to determine the

network usage. In the E-Comm network, the call inter-arrival time are exponentially distributed, while the call holding time fits a lognormal distribution [11].

The number of users' call is of particular interest to our analysis. A commonly used metric in the telecommunication industry is the hourly number of calls. It may be regarded as the footprint of a user's calling behavior. Units less than an hour (minute) is large enough to capture the calling activity since a call usually lasts 3$\sim$5 seconds in the E-Comm network. However, the one minute recording unit may impose large computational cost because of the huge number of data points ($92*24*60 = 132,480$). Units larger than an hour (day) are too coarse to capture user's behavior patterns and will reduce the number of data points to merely 92 in our analysis.

The talk group is the basic talking unit in the E-Comm network. Hence, we use a talk group's hourly number of calls to capture a user's behavior. The collected 92 days of traffic data (2,208 hours) imply that each talk group's calling behavior may be portrayed by the 2,208 ordered hourly numbers of calls. Samples of the hourly number of calls for talk groups 1 and 2 over 168-hour are shown in Figure 4.1, while talk group 20 and 263's calling behavior are shown in Figure 4.2. Table 4.1 shows a small sample of the user's calling behavior. The first column shows the talk group id. The remaining columns are the hourly number of calls starting from 2003-03-01 00:00:00 (hour 1) and ending at 2003-05-31 23:59:59 (hour 2208). One row corresponds to one talk group's calling behavior over the 2,208 hours. This will be used in our clustering analysis.

For simplicity and based on prior experience with clustering tools, we selected AutoClass [12] tool and K-means [10] algorithms to classify the calling patterns of talk groups.

4.2 AutoClass tool

A general approach to clustering is to view it as a density estimation problem. We assume that in addition to the observed variables for each data point, there is a hidden, unobserved variable indicating the "cluster membership" (cluster label). Hence, the data are assumed to be generated from a mixture model and that the labels (cluster

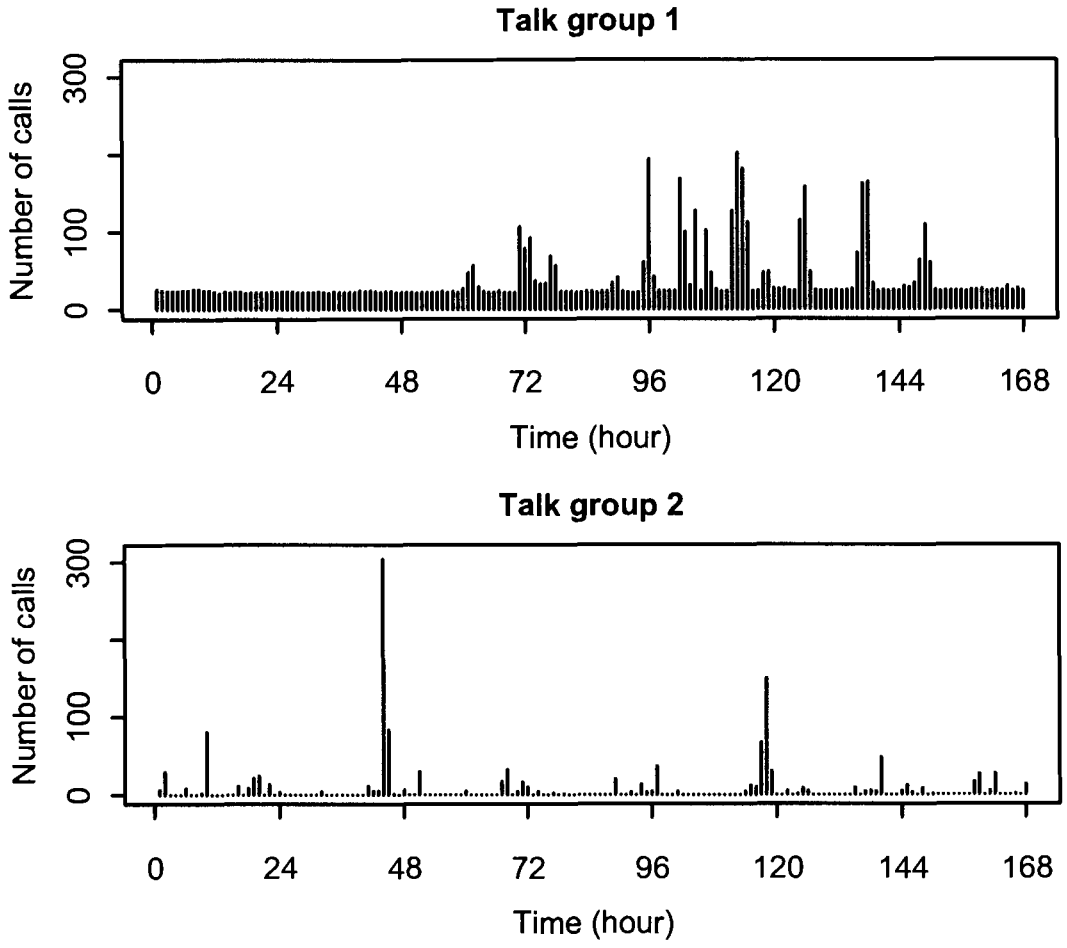

Figure 4.1: Calling patterns for talk groups 1 (top) and 2 (bottom) over the 168-hour period.

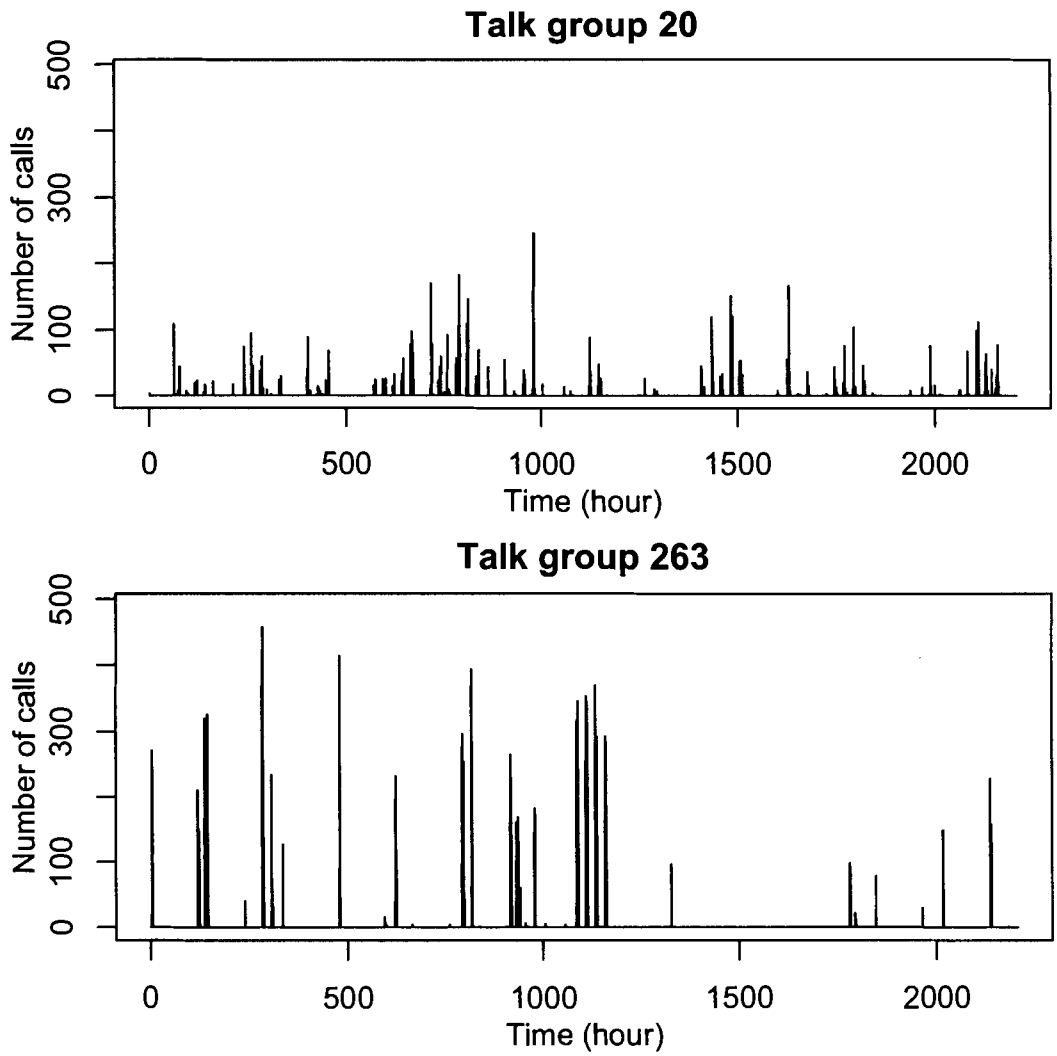

Figure 4.2: Calling patterns for talk groups 20 (top) and 263 (bottom) over the 168-hour period.

Talk group id	Hour 1	Hour 2	Hour 3	Hour 4	... ...	Hour 2206	Hour 2207	Hour 2208
0	26	25	24	20	...	30	24	26
1089	6	29	0	10	...	22	23	0
28	1	0	2	32	...	13	36	81
...	...	...	...	...	...	...	...	...
113	3	0	0	5	...	3	0	0
162	0	0	0	232	...	193	176	256
230	3	0	3	77	...	203	270	187
...	...	...	...	...	...	...	...	...

Table 4.1: Sample of hourly number of calls for various talk groups.

identification) are hidden. In general, a mixture model M has K clusters $C_i, i = 1, ..., K$, assigning a probability to a data point x as:

$$P(x|M) = \sum_{i=1}^{K} W_i * P(x|C_i, M),$$

where W_i is the mixture weight. Some clustering algorithms assume that the number of clusters K is known a priori.

AutoClass [12] is an unsupervised classification tool based on the classical finite mixture model [13]. According to Cheeseman, [9]

> "The goal of Bayesian unsupervised classification is to find the most probable set of class descriptions given the data and prior expectations."

In the past, AutoClass was applied to classify distinct user groups in Telus Mobility Cellular Digital Packet Data (CDPD) network [8].

AutoClass was developed by Bayesian Learning Group at NASA Ames Research Center [14]. We use AutoClass C version 3.3.4. The key features of AutoClass include:

- determining the optimal number of classes automatically
- handling both discrete and continuous values
- handling missing values

- "soft" probabilistic cluster membership instead of "hard" cluster membership.

AutoClass begins by creating a random classification and then manipulates it into a high probability classification through local changes. It repeats the process until it converges to a *local maximum.* It then starts over again and continues until a maximum number of specified tries. Each effort is called a *try.* The computed probability is intended to cover the entire parameter space around this maximum, rather than just the peak. Each new try begins with a certain number of clusters and may conclude with a smaller number of clusters. In general, AutoClass begins the process with a certain number of clusters that previous tries have indicated to be promising.

The input data for AutoClass are stored in two files: data file (.db2) and header file (.hd2). The data file are in vector format. The 2,208 number of calls for each talk group are extracted from database and stored in matrix structure. Each row stands for one talk group and each column is one of the 2,208 hourly number of calls, except that the first column is the identification number of a talk group. In the header file, we specify the data type, name, relative observation error for each column. Part of the header file is shown in Figure 4.2.

AutoClass uses a model file (.model) to describe the possible distribution model for each attribute of the data. Four types of models are currently supported in AutoClass:

- single_multinomial: models discrete attributes as multinomial distribution with missing values. It can handle symbolic or integer attributes that are conditionally independent of other attributes given the class label. Missing values will be represented by one of these existing values.

- single_normal_cn: models real valued attributes as normal distribution without missing values. The model parameters are mean and variance.

- single_normal_cm: models real valued attributes as normal distribution with missing values. The model can be applied to real scalar attributes using a log-transform of the attributes.

```
#Leo Chen, 2003-Oct-22
#the header file for E-Comm data user data clustering
num_db2_format_defs 2
#required
number_of_attributes 2209

# optional - default values are specified
# unknown_token '?'
separator_char ' '
# comment_char '#'

0 discrete nominal "talkgroup" range 1754
1 real scalar "NC[1]" zero_point 0.0 rel_error 0.001
2 real scalar "NC[2]" zero_point 0.0 rel_error 0.001
3 real scalar "NC[3]" zero_point 0.0 rel_error 0.001
... ...
... ...
2205 real scalar "NC[2205]" zero_point 0.0 rel_error 0.001
2206 real scalar "NC[2206]" zero_point 0.0 rel_error 0.001
2207 real scalar "NC[2207]" zero_point 0.0 rel_error 0.001
2208 real scalar "NC[2208]" zero_point 0.0 rel_error 0.001
```

Figure 4.3: Sample of the AutoClass header file.

- multi_normal_cn: a covariant normal model without missing values. This model applies to a set of real valued attributes, each with a constant measurement error and without missing values, which are conditionally independent of other attributes given the cluster label.

The model file used is given in Appendix B. A search parameters file (.s-param) is also used to adjust the search behavior of AutoClass. The most frequently used parameters are *start_j_list*, *fixed_j*, and *max_n_tries*.

- start_j_list: AutoClass will start the search with the certain number of clusters in the list.
- fixed_j: AutoClass will always search for the fixed_j number of clusters, if specified.
- max_n_tries: AutoClass stops search when it reaches the maximum number of the tries.

The detailed description of the remaining searching and reporting parameters may be found in the AutoClass manual [9], [15].

AutoClass used ~20 hours in searching for the best clustering of the 617 talk groups in the E-Comm data. The search results include three important values for the clustering:

- attribute influence values: presents the relative influence or significance of the attributes.
- cross-reference by case number: lists the primary class probability for each datum, ordered by the case number.
- cross-reference by class number: for each class, lists each datum in the class, ordered by case number.

The content of one clustering report is given in Appendix B. The ten best results of talk group clustering are summarized in Table 4.2. The number of talk groups in

Probability	Number of clusters	Number of tries
exp(-6548529.230)	24	653
exp(-6592578.320)	18	930
exp(-6619633.090)	21	940
exp(-6622783.940)	24	323
exp(-6626274.570)	17	542
exp(-6637269.320)	24	1084
exp(-6657627.910)	18	677
exp(-6658596.390)	19	918
exp(-6660040.920)	18	528
exp(-6671271.570)	12	385

Table 4.2: AutoClass results: 10 best clusters.

Cluster ID	Size	Cluster ID	Size	Cluster ID	Size
[1]	144	[2]	67	[3]	66
[4]	31	[5]	25	[6]	23
[7]	22	[8]	21	[9]	20
[10]	20	[11]	19	[12]	19
[13]	18	[14]	18	[15]	18
[16]	17	[17]	15	[18]	13
[19]	12	[20]	10	[21]	9
[22]	4	[23]	3	[24]	3

Table 4.3: AutoClass results: cluster sizes.

each cluster (cluster size) is also shown in the Table 4.3. Hourly number of calls for talk groups in clusters 5, 17, and 22 are shown in Figure 4.4. Talk groups in different clusters exhibit distinct calling behavior patterns.

4.3 K-means algorithm

K-means algorithm is one of the most commonly used data clustering algorithms. It partitions a set of objects into K clusters so that the resulting intra-cluster similarity is high while the inter-cluster similarity is low. The number of clusters K and the

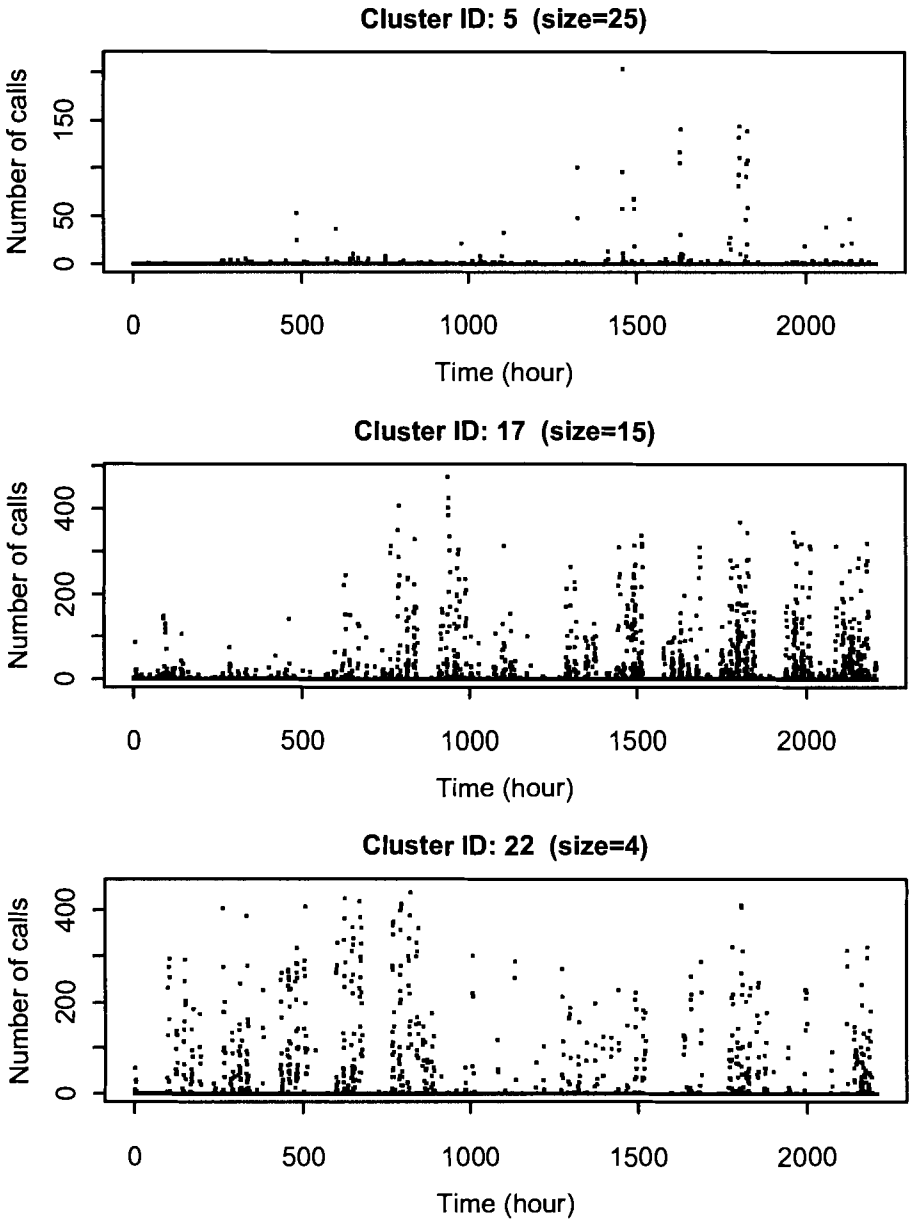

Figure 4.4: Number of calls for three AutoClass clusters with IDs 5 (top), 17 (middle), and 22 (bottom).

object similarity function are two input parameters to the K-means algorithm. Cluster similarity is measured by the average distance from cluster objects to the mean value of the objects in a cluster, which can be viewed as the cluster's center of gravity. The algorithm is well-known for its simplicity and efficiency. It is relatively efficient and stable. The use of various similarity or distance functions makes it flexible. It has numerous variations and it is applicable in areas such as physics, biology, geographical information system, and cosmology. However, its main drawback is its sensitivity to the initial seeds of clusters and outliers, which may distort the distribution of data. In addition, user sometimes may not know a priori the desired number of clusters K, which is the most important input parameter to the algorithm.

The distance between two points is taken as a common metric to assess the similarity among the components of a population. The most popular distance measure is the *Euclidean Distance*. The Euclidean distance of two data points $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ is:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}.$$

We use a variation of K-means, PAM (Partitioning Around Medoids) [10] and our own implementation of K-means to cluster the talk group data. The PAM algorithm searches for K representative objects or medoids among the observations of the data set. It finds K representative objects that minimize the sum of the dissimilarities of the observations to their closet medoids.

We also implemented the classical K-means algorithm using the Perl programming language [16]. The program first seeks K random seeds as cluster centroids in the data set. Based on the Euclidean distance of the object from the seeds, each object is assigned to a cluster. The centroid's position is recalculated every time an object is added to the cluster. This process continues until all the objects are grouped into the final specified number of clusters. Objects change their cluster memberships after the recalculation of the centroids and the re-assignment. Clusters become stable when no object is re-assigned. Different clustering results are obtained depending on the random seeds. However, clustering results for different runs with the same number K

are relatively stable when K is not large, i.e., the clusters converge and different runs result in almost identical cluster partitions.

Without knowing the actual cluster label for each talk group, we are unable to measure the clustering quality using objective measurement factor, such as the F-measure [17]. We use the inter-cluster and the intra-cluster distances to assess the overall clustering quality. The inter-cluster distance is defined as the Euclidean distance between two cluster centroids, which reflects the dissimilarity between clusters. The intra-cluster distance is the average distance of objects from their cluster centroids, expressing the coherent similarity of data in the same cluster. A large inter-cluster distance and a small intra-cluster distance indicate better clusters. The overall clustering quality indicator is defined as the difference between the minimum inter-cluster distance and the maximum intra-cluster distance. The greater the indicator, the better the overall clustering quality. Another measure for the clustering quality is silhouette coefficient [10], which is rather independent on the number of clusters, K. Experience shows that the silhouette coefficient between 0.7 and 1.0 indicates clustering with excellent separation between clusters.

The cluster size, distance measurement, overall quality, and silhouette coefficient of K-means clustering results for clusters with various number of K are shown in Table 4.4. Based on the overall quality and the silhouette coefficient, the best clustering result is obtained for $K = 3$ (in the top three rows). Figure 4.5 shows one week of traffic for each talk groups in the three clusters. The maximum, minimum, and average number of calls for each cluster are also shown. The plots demonstrate the distinct calling behavior of each cluster.

4.4 Comparison of AutoClass and K-means

To compare the clustering results of AutoClass and K-means, we enforce the number of clusters in AutoClass by specifying the parameter *fixed_j* to 3 in the search parameter file. The calling behavior properties for talk groups in the AutoClass clusters and in K-means clusters are compared in Table 4.5. The three clusters identified by K-means are more reasonable than the clusters produced by AutoClass. With

No. of clusters	Cluster size	Average intra dist.	Average inter dist.	Maximum intra dist.	Minimum inter dist.	Overall quality	Silhouette coefficient
3	17, 31, 569	1882.14	4508.38	2971.76	1626.4	-1345.36	0.7756
3	17, 31, 569	1882.14	4508.38	2971.76	1626.4	-1345.36	0.7756
3	17, 31, 569	1882.14	4508.38	2971.76	1626.4	-1345.36	0.7756
4	17, 33, 4, 563	1863	3889.12	2971.76	1556.68	-1415.07	0.7684
4	1, 17, 27, 572	1436.08	3966.26	2971.76	1282.01	-1689.75	0.7632
4	17, 39, 552, 9	2155.46	3848.36	3730.61	1011.97	-2718.63	0.7691
6	13, 17, 22, 3, 34, 528	2059.67	3284.52	3299.43	594.21	-2705.21	0.7640
6	14, 17, 25, 4, 551, 6	2210.88	3353.47	3485.42	1051.92	-2433.49	0.7639
6	15, 17, 3, 42, 5, 535	1693.28	2984.82	3087.33	605.38	-2481.95	0.7635
9	...	1020.08	3520.04	3065.25	808.28	-2256.96	0.7492
9	...	1451.46	2661.29	3687.39	735.37	-2952.01	0.7491
9	...	1478.42	2867.43	3716.73	607.67	-3109.06	0.7483
12	...	1372.67	3582.98	3278.14	731.26	-2546.88	0.7435
12	...	1443.9	2271.58	3436.66	398.95	-3037.7	0.7459
12	...	1676.57	3225.75	3908.67	581.68	-3326.99	0.7456
16	...	983.63	1815.79	3571.27	248.19	-3323.07	0.7337
16	...	1290.87	2154.53	3859.53	320.06	-3539.46	0.7387
16	...	1329.99	2275.42	3478.55	271.6	-3206.95	0.7412
20	...	1355.8	2458.39	3604.33	314.49	-3289.84	0.7386
20	...	1025.44	2296.45	3730.61	413.76	-3316.84	0.7390
20	...	924.43	2042.43	3661.58	343.15	-3318.43	0.7377

Table 4.4: K-means results: cluster size and distances.

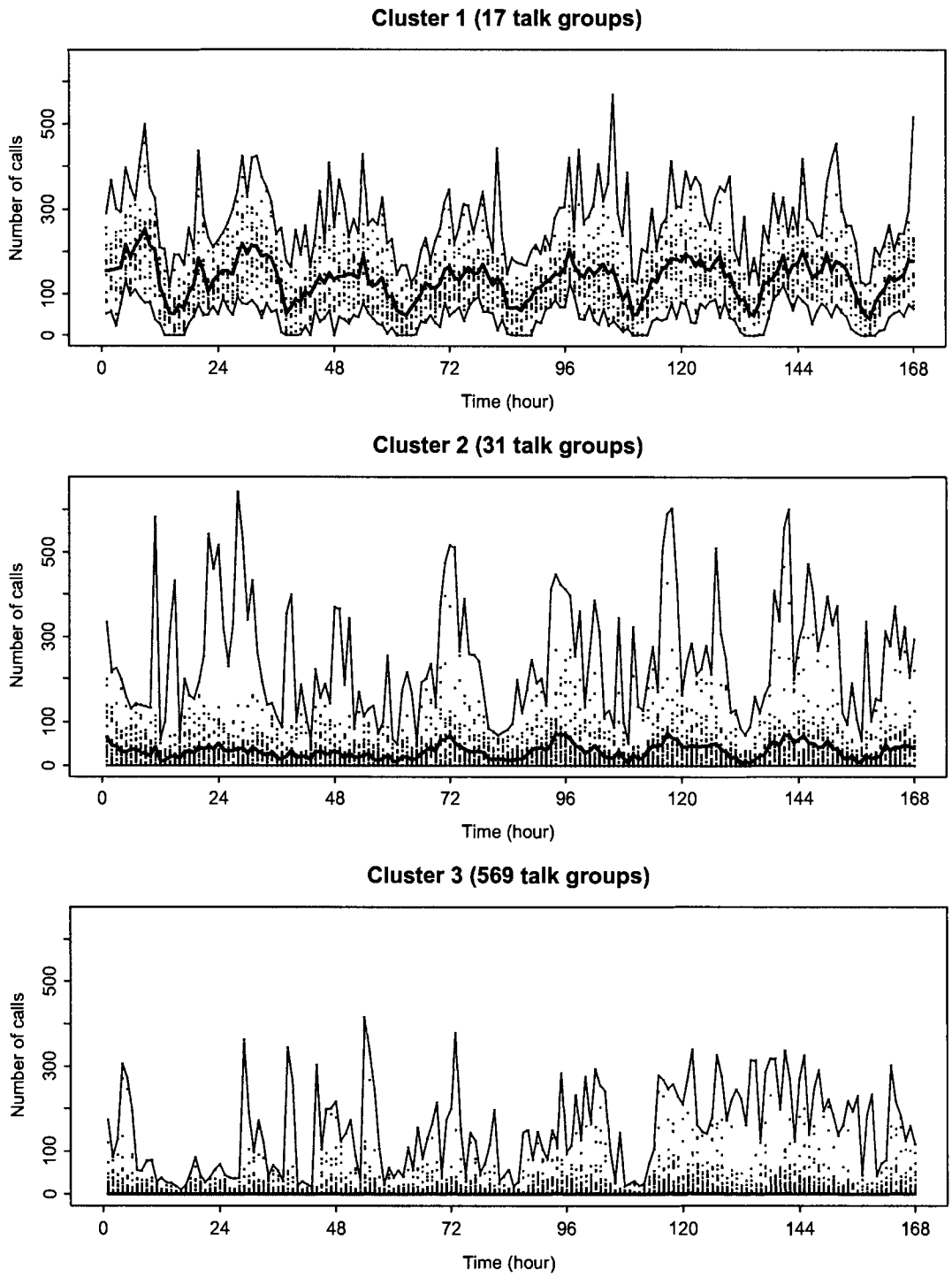

Figure 4.5: K-means result: number of calls in three clusters.

Alg.	Clu. size	Min. nc	Max. nc	Avg. nc	Total nc	Total nc (%)
AC	60	0	2 - 356	0 - 0.7	15,870	0.07
AC	202	0 - 6	7 - 1613	0.04 - 208	8,641,508	99.75
AC	355	0	1 - 243	0 - 0.8	6208	0.18
K	17	0 - 6	352 - 700	94 - 208	5,091,695	59
K	31	0 - 3	135 - 641	17 - 66	2,261,055	26
K	569	0	1 - 1613	0 - 16	1,310,836	15

Table 4.5: Comparison of talk group calling properties (AC: AutoClass, K: K-means, nc: number of calls).

K-means clustering, the first cluster has 17 talk groups, representing the busiest dispatch groups whose main tasks are coordinating and scheduling other talk groups for certain tasks. The second cluster contains 31 talk groups with medium network usage. The last cluster identifies a group of least frequent network users who made on average no more than 16 calls per hour. These interpretations of clusters have been confirmed by domain experts. On the contrary, it is difficult to provide reasonable explanations for group behavior for the three clusters identified by AutoClass. Thus, we use the three clusters identified by K-means in the prediction of network traffic.

4.5 Summary

Clustering analysis of the talk groups' calling behavior reveals hidden structure of talk groups by grouping the talk groups with similar calling behavior rather than by their organizational structure.

We used AutoClass tool and applied K-means algorithm to identify clusters of talk groups based on their calling behavior. Talk groups' behavior patterns are then categorized and extracted from the clusters. The optimal number of clusters is difficult to determine. By comparing the overall quality measurement and the silhouette coefficient measure, we found that three is the best number of clusters for K-means algorithm. Based on the domain knowledge, the three clusters identified by K-means

are more reasonable than clusters produced by AutoClass. Other clustering algorithms, such as hierarchical [18] and density based [19] clustering may also be used to cluster the user data.

Chapter 5

Data prediction

In this Chapter, we describe the time series data analysis and the Auto-Regressive Integrated Moving Average (ARIMA) models. We describe how to identify, estimate, and forecast network traffic using the ARIMA model. We also present the cluster-based prediction models and compare the prediction results with the results of traditional prediction based on aggregate traffic.

5.1 Time series data analysis

Performance evaluation techniques are important in the design of networks, services, and applications. Of particular interest are techniques employed to predict the QoS related network performance. Modeling and predicting network traffic are essential steps in performance evaluation. It helps network planners understand the underlying network traffic process and to predict future traffic. Analysis of commercial network traffic is difficult because the commercial network traffic traces are not easily available. Furthermore, there are privacy and business issues to consider.

The Erlang-C model [20], currently used by the E-Comm staff, was developed based on individual user's calling behavior in wired networks. It considers no-group call behavior in trunked radio systems. Network traffic may also be considered as a series of observations of a random process, and, hence, the classical time-series prediction ARIMA models can be used for traffic prediction.

We employ the Seasonal Autoregressive Integrated Moving Average (SARIMA) model [21], a special case of ARIMA, to predict the E-Comm network traffic. SARIMA models have been applied to modeling and predicting traffic from both large scale networks (NSFNET [22]) and from small scale sub-networks [23]. The fitted model is only an approximation of the data and the quality of the model depends on the complexity of the phenomenon being modeled and the understanding of data.

5.2 ARIMA model

The ARIMA model, developed by Box and Jenkins in 1976 [21], provides a systematic approach to the analysis of time series data. It is a general model for forecasting a time series that can be stationarized by transformations such as differencing and log transformation. Lags of the differenced series appearing in the forecasting equation are called *auto-regressive* terms. Lags of the forecast errors are called *moving average* terms. A time series that needs to be differenced to be made stationary is said to be an *integrated* version of a stationary series. Random-walk and random-trend models, autoregressive models, and exponential smoothing models (exponential weighted moving averages) are special cases of the ARIMA models [24]. ARIMA model is popular because of its power and flexibility.

5.2.1 Autoregressive (AR) models

Regression model is a widely applied multivariate model used to predict the target data based on observations and to analyze the relationship between observations and predictions. Autoregressive model is conceptually similar to the regression model. Instead of the multi-variative observed data, the previous observations of the univariate target data are used as the effective factors of the target data. The regression model assumes the future value of the target variable to be determined by other related observed data, while the autoregressive model assumes the future value of the target variable to be determined by the previous value of the same variable. An AR model closely resembles the traditional regression model.

An AR(p) model X_t can be written as

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + Z_t,$$

where Z_t denotes a random process with zero mean and variance σ^2. Using the backward shift operator B, where $BX_t = X_{t-1}$, the AR(p) model may be written as

$$\phi(B)X_t = Z_t,$$

where $\phi(B) = (1 - \phi_1 B - ... - \phi_p B^p)$ is a polynomial in B of order p.

Figure 5.1: Definition of the autoregressive (AR) model.

A time series X_t is said to be a **moving-average process of order q** if

$$X_t = Z_t + \theta_1 Z_{t-1} + ... + \theta_q Z_{t-q},$$

where $Z_t \sim WN(0, \sigma^2)$ denotes a random process with zero mean and constant variance σ^2 and $\theta_1, ..., \theta_q$ are constants.

Figure 5.2: Definition of the moving average (MA) model.

5.2.2 Moving average (MA) models

A moving average model describes a time series whose elements are sums of a series of random *shock* values. The process that generates a moving average model has no memory of past values. For example, a time series of an MA(1) process might be generated by a variable with measurement error or a process where the impact of a shock takes one period to fade away. In an MA(2) process, the shock takes two periods to completely fade away.

An ARIMA(p, d, q) model X_t can be written as

$$\phi(B)(1-B)^d X_t = \theta(B)Z_t,$$

where $\phi(B)$ and $\theta(B)$ are polynomials in B of finite order p and q, respectively. The backward shift operator B is defined as $B^i X_t = X_{t-i}$. A SARIMA $(p,d,q) \times (P,D,Q)_S$ model exhibits seasonal pattern and can be represented as:

$$\phi(B^s)\phi(B)(1-B^s)^D(1-B)^d X_t = \theta(B^s)\theta(B)Z_t,$$

where $\phi(B)$ and $\theta(B)$ represent the AR and MA parts, and $\phi(B^s)$ and $\theta(B^s)$ represent the seasonal AR and seasonal MA parts, respectively. B is the back-shift operator $B^i X_t = X_{t-i}$.

Figure 5.3: Definition of the ARIMA/SARIMA model.

5.2.3 SARIMA $(p,d,q) \times (P,D,Q)_S$ models

The ARIMA model includes both autoregressive and moving average parameters and explicitly includes in the formulation of the model differencing, which is used to stationarize the series. The three types of parameters in the model are: the autoregressive order (p), the number of differencing passes (d), and the moving average order (q). Box and Jenkins denote it as ARIMA (p, d, q) [21]. For example, a model ARIMA (0, 1, 2) means that it contains 0 (zero) autoregressive (p) order, 2 moving average (q) parameters, and the model fits the series after being differenced once (1). A SARIMA model is a ARIMA model plus seasonal fluctuation. It comprises normal orders (p, d, q) and seasonal orders (P, D, Q), and the seasonal period S. A general SARIMA model is denoted as SARIMA $(p,d,q) \times (P,D,Q)_S$.

5.2.4 SARIMA model selection

The general ARIMA model building process has three major steps:

- model identification
- model estimation
- model verification.

Model identification is used to decide the orders of the model, i.e., to determine the value of orders p, d, q, seasonal orders P, D, Q, and the seasonal period S. The $\phi(x)$ and $\theta(x)$ coefficients are computed in the model estimation phase, using minimum linear square error method or maximum likelihood estimation methods. Models are verified by diagnostic checking on the null hypothesis of the residual or by various tests, such as Box-Ljung and Box-Pierce tests [21], [24], [25].

The major tools used in the model identification phase include plots of the time series, correlograms of autocorrelation function (ACF), and partial autocorrelation function (PACF). Model identification is often difficult and in less typical cases requires not only experience but also a good deal of experimentation with models with various orders and parameters. The relation of the ACF with the MA(q) model, and the relation of the PACF with the AR(p) model, are shown in Figure 5.4.

We use three measurements to find the best models and check the validity of the model parameters. A smaller value of the measurement indicates a better selection of model.

- Akaike's Information Criterion (AIC)
  $AIC = -2ln(max.likelihood) + 2p$
- Akaike's Information Criterion Corrected (AIC_C)
  $AIC_C = AIC + 2(p+1)(p+2)/(N-p-2)$
- Bayesian Information Criterion (BIC)
  $BIC = -2ln(max.likelihood) + p + plnN$

Let $\{Y_t\}$ be the MA(q) model, so the ACF $\rho(k)$

$$\rho(k) = \begin{cases} \sum_{i=0}^{q-|k|} \theta_i \theta_{i+|k|} / \sum_{i=0}^{q} \theta_i^2, & |k| \leq q, k \neq 0, \\ 1, & k = 0, \\ 0, & \text{otherwise.} \end{cases}$$

The PACF of a stationary time series is defined as

$$\begin{aligned} &\phi_{11} = \rho(1), \\ &\phi_{kk} = corr(Y_{k+1} - P_{\overline{sp}\{Y_2,\cdots,Y_k\}} Y_{k+1}, Y_1 - P_{\overline{sp}\{Y_2,\cdots,Y_k\}} Y_1), \\ &k \geq 2, \end{aligned}$$

where $P_{\overline{sp}\{Y_2,\cdots,Y_k\}} Y$ denotes the projection of the random variable Y onto the closed linear subspace spanned by the random variable $\{Y_2, \cdots, Y_k\}$.

Theorem [23]
For an AR(p), $\phi_{kk} = 0$ *for* $k > p$.

Figure 5.4: Auto-correlation function and Partial auto-correlation function.

$(p,d,q) \times (P,D,Q)_S$	m	$nmse$	AIC	AIC_C	BIC
$(2,0,1) \times (0,1,1)_{168}$	1512	0.173	20558.7	20558.8	20590.3
$(2,0,9) \times (0,1,1)_{24}$	1512	0.379	22744.6	22744.9	22826.8
$(2,0,1) \times (0,1,1)_{168}$	1680	0.174	23129.8	23129.8	23161.9
$(1,0,1) \times (0,1,1)_{168}$	1680	0.175	23145.1	23145.1	23170.8
$(2,0,9) \times (1,1,1)_{24}$	1680	0.5253	25292.1	25292.4	25382.1
$(1,0,2) \times (1,1,1)_{24}$	1680	0.411	25332.6	25332.6	25371.2
$(2,0,9) \times (0,1,1)_{24}$	1680	0.546	25345.9	25346.1	25429.4
$(2,0,1) \times (0,1,1)_{24}$	1680	0.537	25360.5	25360.6	25392.6
$(3,0,1) \times (0,1,1)_{24}$	1680	0.404	25361.2	25361.2	25399.7

Table 5.1: Summary of SARIMA models fitting measurement.

We test a series of SARIMA models selected based on the time series plot, ACF, and PACF. The measurement results for several SARIMA models are shown in Table 5.1. The rows are sorted in ascending order of the value of the measurement BIC. Based on the same amount of training data 1,680, the model $(2,0,1) \times (0,1,1)_{168}$ has the smallest BIC value. Thus, it may be the most suitable model for the data we tested.

Null hypothesis test was used to check a model's goodness-of-fit. They verify the randomness of the time series and may be applied to the residual of the model. If the identified/estimated model fits the training data well, the residual obtained by subtracting the fitted data from the original observation. should be a true random series. Usual null hypothesis test includes time plot analysis and ACF checks. In addition, two types of goodness-of-fit test, Box-Ljung and Box-Pierce tests may be used to check the null hypothesis of the model.

Figures 5.5 and 5.6 show the time plot of the residual series and their ACF function, for two SARIMA models $(3,0,1) \times (0,1,1)_{24}$ and $(1,1,0) \times (0,1,1)_{24}$, respectively. Also shown are the P-value [26] of the Box-Ljung test for these two models. P-value of the test represents the probability that the sample could have been drawn from the population(s) being tested given the assumption that the null hypothesis is true. Thus, a higher P-value implies that the model being tested are more likely to pass the null hypothesis test. Based on the plot and P-value, the model $(3,0,1) \times (0,1,1)_{24}$

passed the null hypothesis test, while the model $(1,1,0) \times (0,1,1)_{24}$ failed.

5.3 Prediction based on aggregate traffic

The correlograms of the autocorrelation function and the partial autocorrelation function of the E-Comm data are shown in Figures 5.7 and 5.8, respectively. By differencing the sample data with 24 hours lag, we estimate from the ACF shown in Figure 5.7 that the MA order could be up to 9. Based on the PACF shown in Figure 5.8, we estimate that the order of AR part is 2 because of the apparent cut-off at lag 2. Hence, the ARIMA models (2, 0, 1) and (2, 0, 9) are selected as model candidates.

The order (0, 1, 1) are commonly used for seasonal part (P, D, Q). It is selected because the cyclical seasonal pattern itself is usually a random-walk process and may be modeled as an MA (1) process after one time differencing. Thus, we use the order of (0, 1, 1) for seasonal pattern.

A useful metric called normalized mean square error (*nmse*) is used to measure the prediction quality by comparing the deviation of the predicted data and the observed data. The *nmse* of the forecast is equal to the normalized sum of the variance of the forecast divided by the squared bias of the forecast. It is defined as

$$nmse(a,b) = \sum_{i=m+1}^{m+n} \frac{(a_i - b_i)^2}{(a_i - \bar{a})^2} \quad ,$$

where a_i is the observed data, b_i is the prediction, and $\bar{a}$ is the mean value of a_i. Smaller values of *nmse* indicate better model performance.

An open source statistical tools R [27], [28], [29] was used to identify, estimate, and verify the SARIMA model and to forecast the traffic. The E-Comm network traffic possesses both daily and weekly patterns. Hence, both 24-hour and 168-hour (one week) intervals are selected as seasonal period parameters. Hence, in addition to the $(2,0,9) \times (0,1,1)_{24}$ and $(2,0,1) \times (0,1,1)_{24}$ models, two models $(2,0,9) \times (0,1,1)_{168}$ and $(2,0,1) \times (0,1,1)_{168}$ are also used to predict the network traffic. The four models and corresponding parameters fitted for the E-Comm network traffic are shown in Table 5.2. The model performance is tested with four groups of data (A, B, C, and

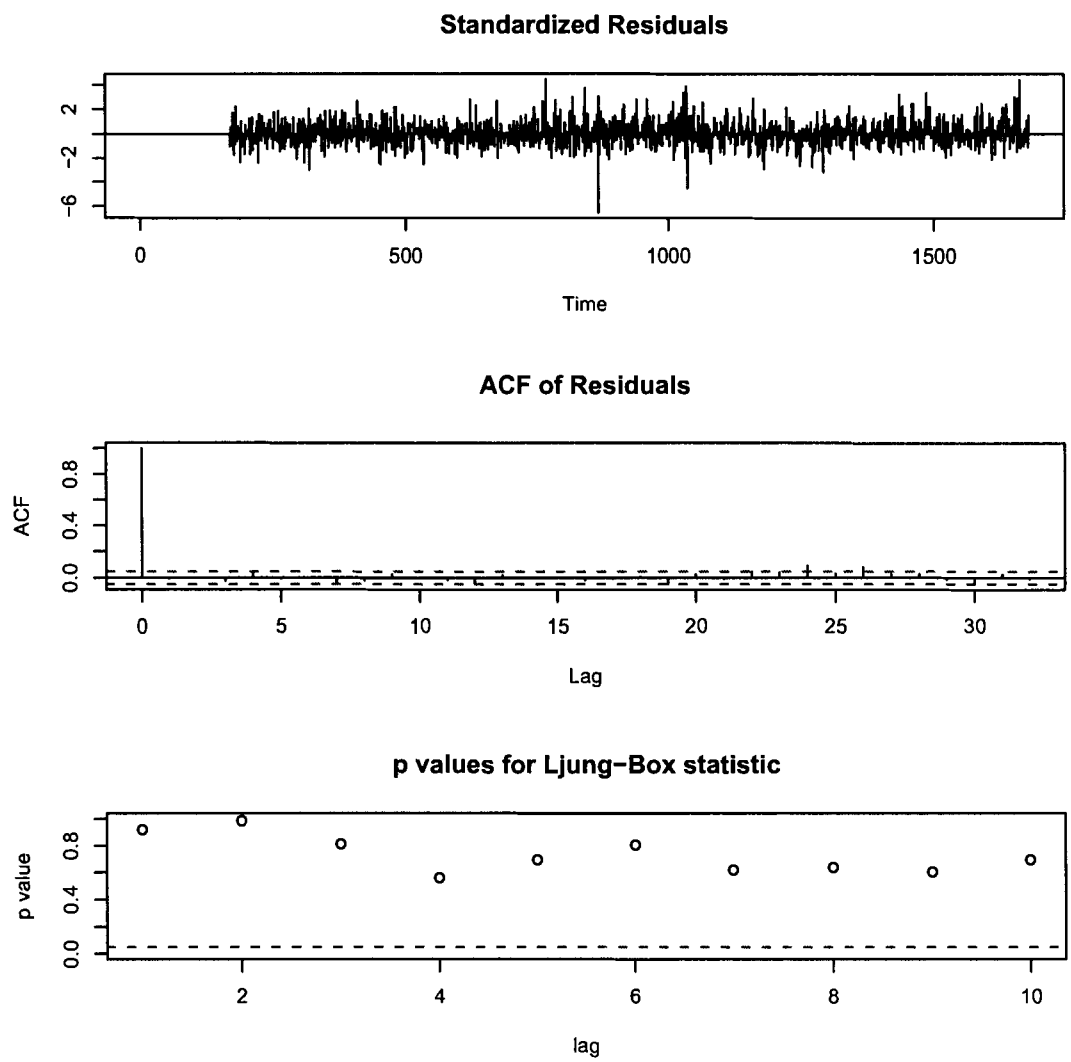

Figure 5.5: Residual analysis: diagnostic test for model $(3,0,1) \times (0,1,1)_{24}$.

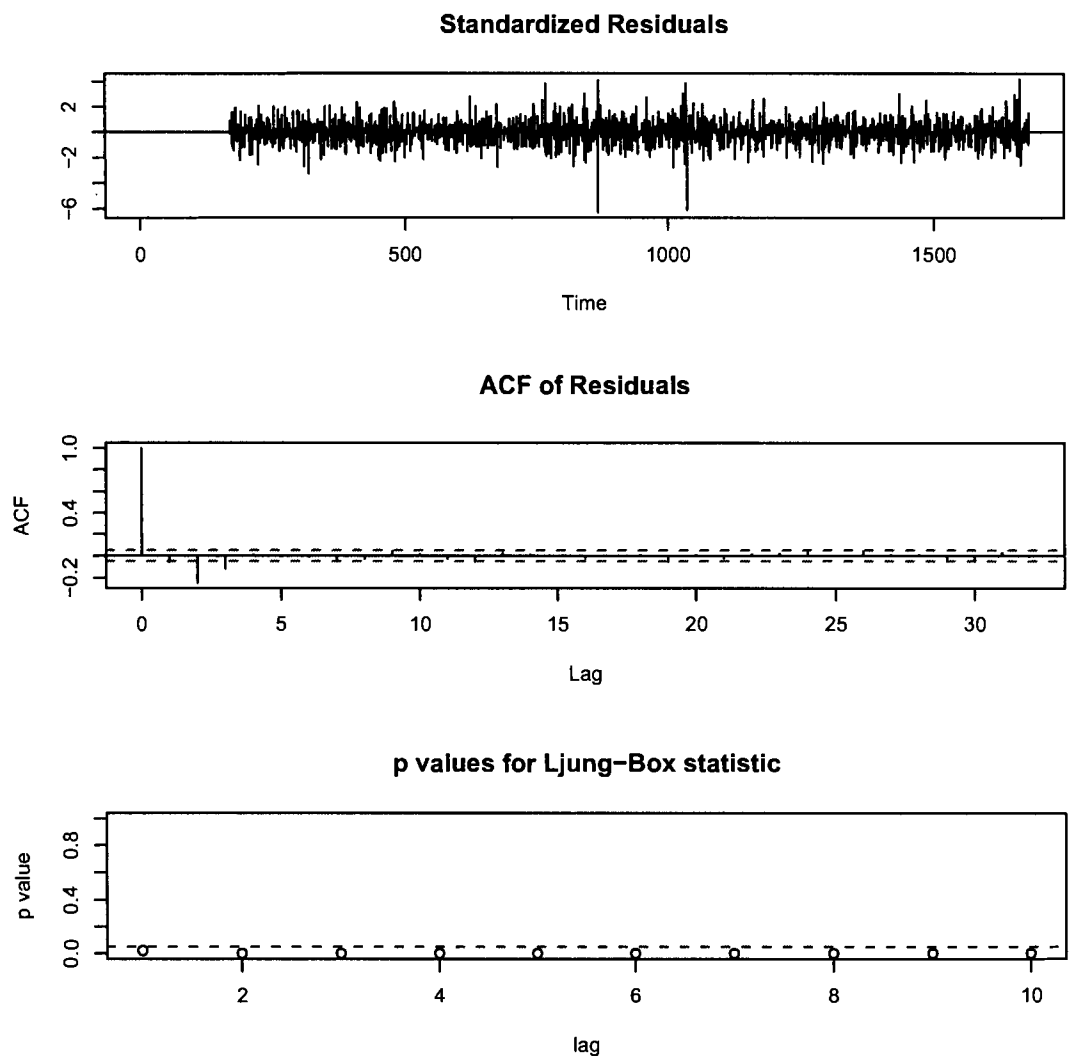

Figure 5.6: Residual analysis: diagnostic test for model $(1,1,0) \times (0,1,1)_{24}$.

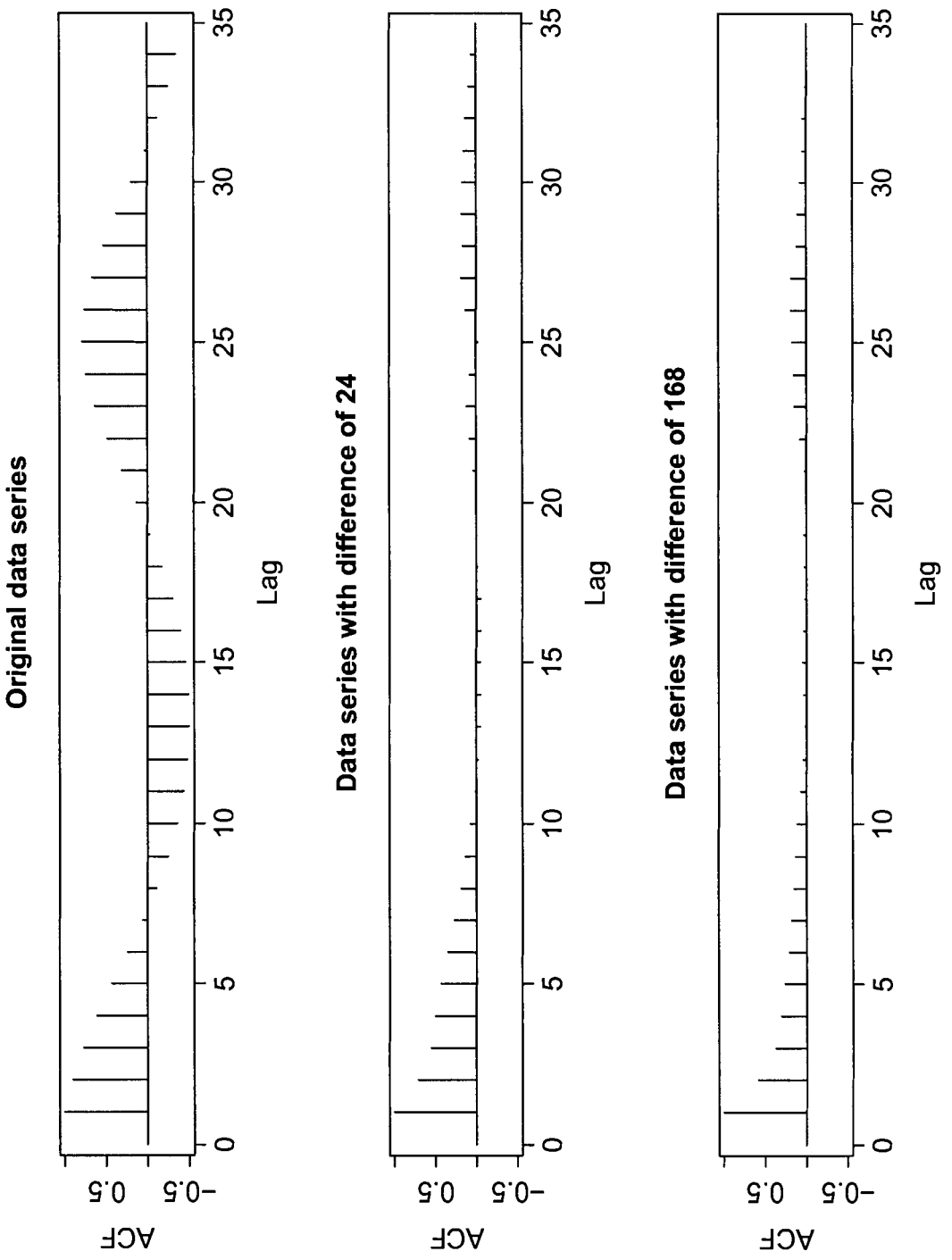

Figure 5.7: Number of calls: sample auto-correlation function (ACF).

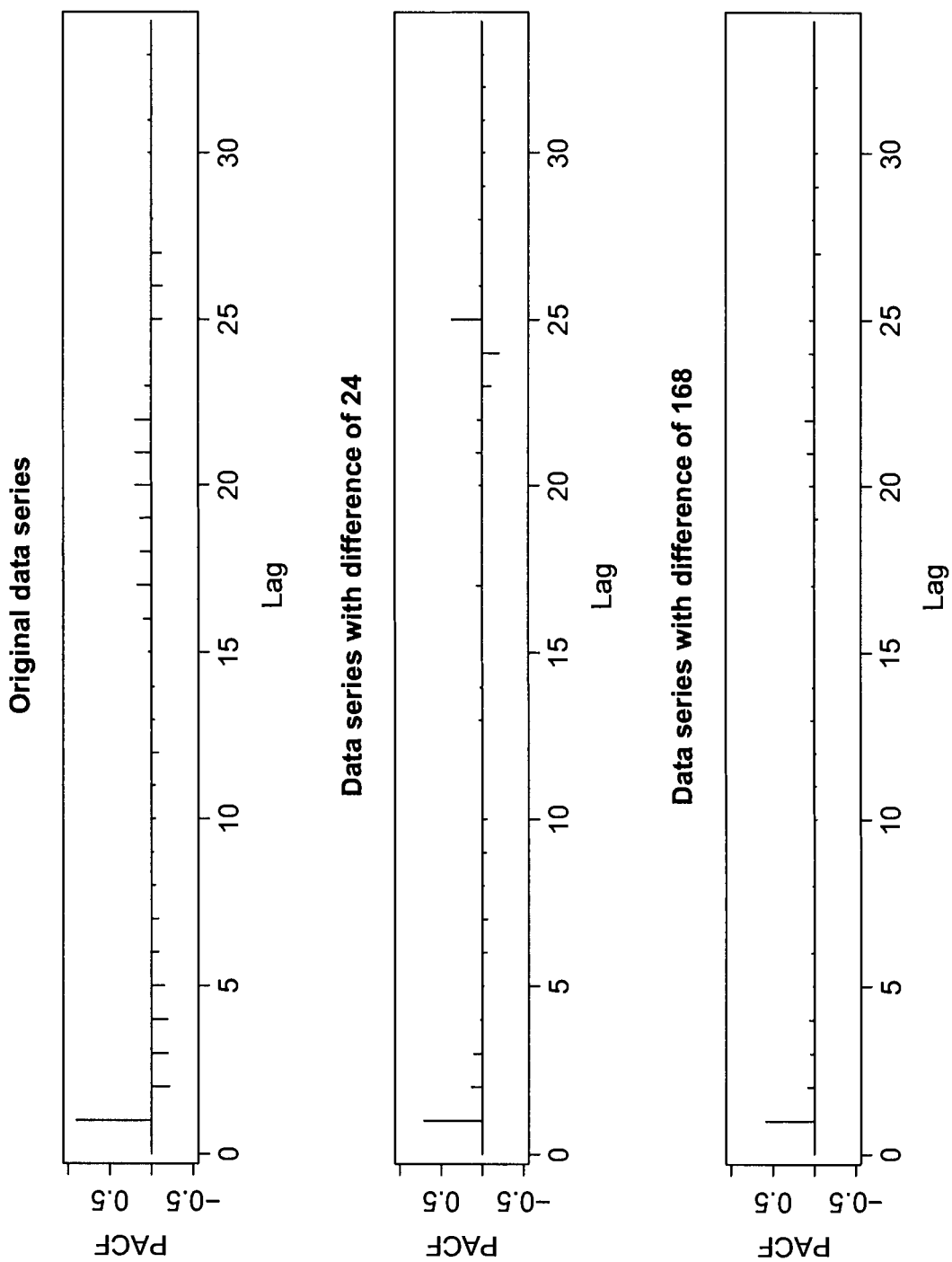

Figure 5.8: Number of calls: sample Partial auto-correlation function (PACF).

No.	p	d	q	P	D	Q	S	Trained (m)	Predicted (n)	*nmse*
A1	2	0	9	0	1	1	24	1512	672	0.3790
A2	2	0	1	0	1	1	24	1512	672	0.3803
A3	2	0	9	0	1	1	168	1512	672	0.1742
A4	2	0	1	0	1	1	168	1512	672	0.1732
B1	2	0	9	0	1	1	24	1680	168	0.3790
B2	2	0	1	0	1	1	24	1680	168	0.4079
B3	2	0	9	0	1	1	168	1680	168	0.1736
B3	2	0	1	0	1	1	168	1680	168	0.1745
C1	2	0	9	0	1	1	24	1920	24	0.3164
C2	2	0	1	0	1	1	24	1920	24	0.1941
C3	2	0	9	0	1	1	168	1920	24	0.1002
C4	2	0	1	0	1	1	168	1920	24	0.0969
D1	2	0	9	0	1	1	24	2016	168	0.3384
D2	2	0	1	0	1	1	24	2016	168	0.3433
D3	2	0	9	0	1	1	168	2016	168	0.1282
D4	2	0	1	0	1	1	168	2016	168	0.1178

Table 5.2: Aggregate-traffic-based prediction results.

D). We forecast the future n traffic data based on m past traffic data samples. In Table 5.2, p, d, q represent the order of the AR, difference, and MA model for the original data points, respectively. The P, D, Q represent the order of AR, difference, and MA model for the seasonal pattern, respectively. S is the seasonal period for the models.

Four SARIMA models with four groups of training data are shown in Table 5.2. The models differ in the order of moving average and the seasonal period.

- Model 1: $(2,0,9) \times (0,1,1)_{24}$ (rows A1, B1, C1, and D1) is the model with 24-hour seasonal period and moving average of order 9. The model performance does not depend on the number of training data, with *nmse* ranging from 0.3164 to 0.3790.

- Model 2: $(2,0,1) \times (0,1,1)_{24}$ (rows A2, B2, C2, and D2) is the model with 24-hour seasonal period and moving average of order 1. It exhibits similar

prediction effectiveness as Model 1. It performs better in row C2 than model 1 in row C1.

- Model 3: $(2,0,9) \times (0,1,1)_{168}$ (rows A3, B3, C3, and D3) is the model with a 168-hour period weekly cycle. It differs from Model 1 only in the seasonal period, but provides much better prediction results than Model 1.

- Model 4: $(2,0,1) \times (0,1,1)_{168}$ (rows A4, B4, C4, and D4), differs from Model 2 in the seasonal period. It performs better than Model 2.

The comparisons of rows A1 with A2, B1 with B2, and D1 with D2, indicate that Model 1 leads to better prediction results than Model 2. However, the prediction C1 is worse than C2. Furthermore, for all four groups of training data, Models 3 and 4 with 168-hour period always lead to better prediction results than Models 1 and 2 with 24-hour period. The 24-hour period models assume that the traffic is relatively constant for a weekday, while the 168-hour period models take into account traffic variations between between weekdays. To predict traffic on a Wednesday based on Tuesday's data not as accurate as predicting Wednesday's traffic based on the data of previous Wednesdays. However, the computational cost of identifying and forecasting 168-hour period models is much larger that for the 24-hour period models. Often, 168-hour models require over 100 times the CPU needed for 24-hour models. A comparison of the prediction results of the 24-hour model and the 168-hour model in predicting one future week of traffic based on the 1,680 past hours is shown in Figure 5.9. It is consistent with the *nmse* value. The 168-hour period model performs better than the 24-hour period model. The continuous curve shows the observation data. Symbol "o" indicate the predicted traffic based on the 168-hour season model. Symbol "*" denotes the prediction of the 24-hour season model. Based on the *nmse* values, the prediction of the 168-hour based model fits better the observations than the prediction based on the 24-hour model.

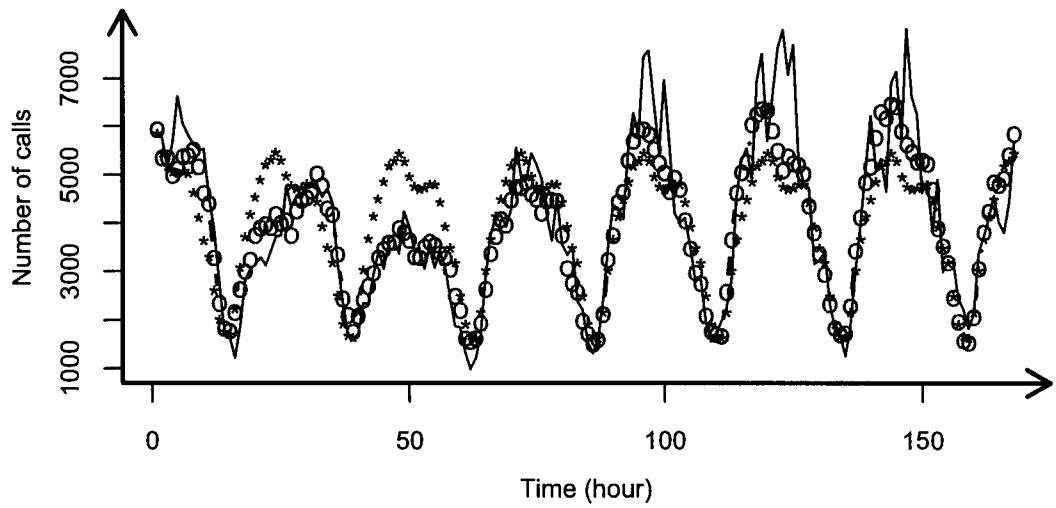

Figure 5.9: Predicting 168 hours of traffic data based on the 1,680 past data.

5.4 Cluster-based prediction approach

A key assumption of the prediction based on the aggregate traffic described in Section 5.3 is the constant number of network users and constant behavior patterns. However, this assumption does not hold in case of network expansions. Hence, it is difficult to use traditional models to forecast traffic of such networks. We propose here a cluster-based approach to predict the network traffic by aggregating traffic predicted for individual clusters.

Network users are classified into clusters according to the similarity of their behavior. It is impractical to predict each individual user's traffic and then aggregate the predicted data. With user clusters, this task reduces to predicting and then aggregating several clusters of users' traffic. For each clusters produced by K-means in Section 4.3, we predict network traffic using SARIMA models $(2,0,1) \times (0,1,1)_{24}$ and $(2,0,1) \times (0,1,1)_{168}$. Results of the cluster-based prediction are compared to the prediction based on aggregate traffic in Table 5.3.

In Table 5.3, rows marked **A** represent the prediction based on aggregate user

Cluster	(p,d,q)	(P,D,Q)	S	m	n	*nmse*
1	(2,0,1)	(0,1,1)	24	1680	48	1.1954
2	(2,0,1)	(0,1,1)	24	1680	48	2.4519
3	(2,0,1)	(0,1,1)	24	1680	48	0.3701
A	(2,0,1)	(0,1,1)	24	1680	48	0.6298
*	(2,0,1)	(0,1,1)	24	1680	48	0.6256
O	(2,0,1)	(0,1,1)	24	1680	48	**0.4231**
1	(2,0,1)	(0,1,1)	168	1,920	24	0.2241
2	(2,0,1)	(0,1,1)	168	1,920	24	0.3818
3	(2,0,1)	(0,1,1)	168	1,920	24	0.1163
A	(2,0,1)	(0,1,1)	168	1,920	24	0.0969
*	(2,0,1)	(0,1,1)	168	1,920	24	0.1175
1	(2,0,1)	(0,1,1)	24	1,920	24	0.2508
2	(2,0,1)	(0,1,1)	24	1,920	24	0.2697
3	(2,0,1)	(0,1,1)	24	1,920	24	0.3020
A	(2,0,1)	(0,1,1)	24	1,920	24	0.1941
*	(2,0,1)	(0,1,1)	24	1,920	24	0.2052
1	(2,0,1)	(0,1,1)	24	1,680	168	0.5477
2	(2,0,1)	(0,1,1)	24	1,680	168	0.6883
3	(2,0,1)	(0,1,1)	24	1,680	168	0.2852
A	(2,0,1)	(0,1,1)	24	1,680	168	0.4079
*	(2,0,1)	(0,1,1)	24	1,680	168	0.4093

Table 5.3: Summary of the results of cluster-based prediction.

traffic (without clustering of users) using the model shown in rows A2, B2, C2, and D2 in Table 5.2. Rows **1**, **2**, and **3** represent traffic prediction for user clusters 1, 2, and 3, respectively. Row * is the weighted aggregate prediction of network traffic based on the prediction for three user clusters. Row **O** stands for the optimized weighted aggregate prediction. Note that the $nmse > 1.0$ for clusters 1 and 2 implies that the prediction results are worse than prediction based on the mean value of past data. A better prediction shown in row **O** is obtained if the mean value prediction is adopted for clusters 1 and 2. We named it the optimized cluster-based prediction. Even with un-optimized clustered based prediction (row *), the prediction results are not worse than results of prediction based on aggregate traffic (rows **A**).

The advantage of the cluster-based prediction is that we could predict traffic in a network with variable number of users as long as the new user groups could be classified into the existing user clusters. The computational cost of forecasting the network traffic is reduced to the number of clusters times the prediction cost for one cluster.

5.5 Additional prediction results

Additional prediction results are presented in Tables 5.4 – 5.7. The experimental results show that 57% of the cluster-based prediction models perform better than the prediction models based on aggregate traffic when the seasonal period is 168 hours. Furthermore, 7 out of 8 optimized models give better prediction results when the model seasonal period is 24 hours.

5.5.1 Comparison of predictions with the $(2,0,1) \times (0,1,1)_{24}$ model

The results of cluster-based prediction and the prediction based on aggregate traffic are compared in Tables 5.4 and 5.5. In the tables, pdq, PDQ, and S are SARIMA model orders, seasonal orders, and season period, respectively. m is the number of model training data and n is the number of predicted data. Tables 5.4 and 5.5 also

show the *nmse* for prediction of each cluster, *nmse* for prediction based on aggregate user traffic, *nmse* for cluster-based prediction, and *nmse* for optimized cluster-based prediction, if any. Note that we use the same optimization method as used in Table 5.3: we use the mean value of training data m to replace the "bad" cluster predictions when $nmse > 1.0$. Rows marked "()" indicate that the cluster-based predictions perform better than the predictions based on aggregate traffic (8 out of 56). Rows marked "[]" show that the optimized cluster-based prediction performs better than the prediction based on aggregate traffic (7 out of 56). 7 out of 8 optimized predictions perform better than the aggregate-traffic-based predictions, which proves the effectiveness of the proposed optimization method.

5.5.2 Comparison of predictions with the $(2,0,1) \times (0,1,1)_{168}$ model

The results of cluster-based prediction and the prediction based on aggregate traffic using SARIMA model $(2,0,1) \times (0,1,1)_{168}$ are compared in Tables 5.6 and 5.7. In the tables, pdq, PDQ, and S are SARIMA model orders, seasonal orders, and season period, respectively. m is the number of model training data and n is the number of data predicted. Tables 5.6 and 5.7 also show the *nmse* for prediction of each cluster, *nmse* for prediction based on aggregate user traffic, *nmse* for cluster-based prediction, and *nmse* for optimized cluster-based prediction, if any. Note that we also applied the same optimization method as used in Table 5.3, which replaces "bad" prediction results ($nmse > 1.0$) with the mean value of training data m. None of the optimized cluster-based predictions performs better than the predictions based on aggregate user traffic. However, more than 57% cluster-based predictions perform better than the predictions based on aggregate traffic, which are shown in rows marked "()".

5.6 Summary

In this Chapter, we described the analysis of time series data, emphasizing the SARIMA models. The SARIMA model was used to fit the aggregate network traffic

No	pdq	PDQ	S	m	n	*nmse* cluster1	*nmse* cluster2	*nmse* cluster3	*nmse* aggregate	*nmse* clusters	*nmse* optimized
(1)	201	011	24	240	24	0.3237	0.5481	0.3084	0.2546	(0.2416)	n/a
(2)	201	011	24	240	48	0.3942	0.7123	0.4457	0.3431	(0.3324)	n/a
(3)	201	011	24	240	72	0.4367	0.6914	0.4596	0.3708	(0.3605)	n/a
(4)	201	011	24	240	96	0.4278	0.8055	0.3828	0.3559	(0.3456)	n/a
(5)	201	011	24	240	120	0.4992	0.8066	0.3556	0.3904	(0.381)	n/a
(6)	201	011	24	240	144	0.5073	0.7856	0.3316	0.3905	(0.3831)	n/a
(7)	201	011	24	240	168	0.4804	0.7793	0.3345	0.3806	(0.3748)	n/a
(8)	201	011	24	480	24	0.4249	0.322	0.1071	0.1203	(0.1187)	n/a
9	201	011	24	480	48	0.3189	0.3427	0.1661	0.1661	0.167	n/a
10	201	011	24	480	72	0.7208	0.5083	0.2042	0.3206	0.3419	n/a
11	201	011	24	480	96	0.712	0.5202	0.2449	0.3673	0.3933	n/a
12	201	011	24	480	120	0.5282	0.4922	0.3037	0.3422	0.3633	n/a
13	201	011	24	480	144	0.4408	0.4841	0.3116	0.3122	0.3223	n/a
14	201	011	24	480	168	0.3943	0.4817	0.3015	0.3016	0.3046	n/a
15	201	011	24	720	24	0.2699	2.029	0.3572	0.221	0.2535	0.2993
16	201	011	24	720	48	0.3063	0.6413	0.3788	0.2894	0.2986	n/a
17	201	011	24	720	72	0.3439	0.687	0.3901	0.3255	0.3321	n/a
18	201	011	24	720	96	0.3146	0.714	0.4147	0.3103	0.3151	n/a
19	201	011	24	720	120	0.3055	0.7431	0.3585	0.3081	0.3107	n/a
20	201	011	24	720	144	0.3482	0.7229	0.3551	0.334	0.3352	n/a
21	201	011	24	720	168	0.4586	0.7054	0.4089	0.4105	0.4117	n/a
22	201	011	24	960	24	0.1621	0.2336	0.1112	0.08545	0.09052	n/a
23	201	011	24	960	48	0.1636	0.3572	0.1209	0.09983	0.103	n/a
24	201	011	24	960	72	0.2262	0.4515	0.3382	0.2409	0.2449	n/a
25	201	011	24	960	96	0.3613	0.5711	0.4261	0.3387	0.3429	n/a
26	201	011	24	960	120	0.4374	0.6528	0.3934	0.3703	0.3738	n/a
27	201	011	24	960	144	0.4408	0.6533	0.3629	0.3469	0.3504	n/a
28	201	011	24	960	168	0.4625	0.6376	0.3332	0.3378	0.3414	n/a

Table 5.4: Comparison of predictions with $(2,0,1) \times (0,1,1)_{24}$ model: part 1.

No	pdq	PDQ	S	m	n	*nmse* cluster1	*nmse* cluster2	*nmse* cluster3	*nmse* aggregate	*nmse* clusters	*nmse* optimized
[29]	201	011	24	1200	24	2.413	1.579	0.2961	0.5975	0.5794	[0.3952]
[30]	201	011	24	1200	48	5.716	2.808	0.2886	1.131	1.125	[0.8403]
[31]	201	011	24	1200	72	1.774	1.976	0.2708	0.8846	0.886	[0.8463]
[32]	201	011	24	1200	96	1.319	0.8667	0.2602	0.6112	0.6138	[0.6106]
33	201	011	24	1200	120	0.8409	0.7031	0.2457	0.4637	0.4672	n/a
34	201	011	24	1200	144	0.6657	0.6472	0.2367	0.3966	0.3997	n/a
35	201	011	24	1200	168	0.7114	0.6709	0.2384	0.3863	0.3877	n/a
36	201	011	24	1440	24	0.4070	0.5568	0.2421	0.3257	0.3363	n/a
37	201	011	24	1440	48	0.5247	0.6054	0.2556	0.3803	0.3917	n/a
38	201	011	24	1440	72	0.6247	0.6268	0.23	0.4209	0.4328	n/a
39	201	011	24	1440	96	0.5938	0.6044	0.2755	0.4236	0.4326	n/a
40	201	011	24	1440	120	0.5915	0.6317	0.2728	0.4264	0.4334	n/a
41	201	011	24	1440	144	0.5472	0.6506	0.2810	0.4093	0.4139	n/a
42	201	011	24	1440	168	0.5222	0.6306	0.2715	0.3896	0.3943	n/a
[43]	201	011	24	1680	24	0.9441	1.522	0.4123	0.5372	0.5486	[0.4657]
[44]	201	011	24	1680	48	1.195	2.452	0.3701	0.6256	0.6298	[0.4231]
[45]	201	011	24	1680	72	0.9587	2.074	0.3459	0.5968	0.5968	[0.494]
46	201	011	24	1680	96	0.6411	0.9414	0.3347	0.4707	0.471	n/a
47	201	011	24	1680	120	0.5395	0.666	0.3172	0.4128	0.4139	n/a
48	201	011	24	1680	144	0.5155	0.6677	0.3011	0.4041	0.4057	n/a
49	201	011	24	1680	168	0.5477	0.6885	0.2853	0.4079	0.4093	n/a
50	201	011	24	1920	24	0.2509	0.2696	0.3013	0.1942	0.2050	n/a
51	201	011	24	1920	48	0.278	0.5794	0.3844	0.3227	0.3289	n/a
52	201	011	24	1920	72	0.2846	0.6823	0.3584	0.3761	0.3821	n/a
53	201	011	24	1920	96	0.2690	0.676	0.3253	0.3513	0.3573	n/a
54	201	011	24	1920	120	0.3312	0.6597	0.3935	0.3797	0.3848	n/a
55	201	011	24	1920	144	0.3696	0.6635	0.3944	0.3964	0.4005	n/a
56	201	011	24	1920	168	0.3678	0.6753	0.3694	0.3927	0.3957	n/a

Table 5.5: Comparison of predictions with $(2,0,1) \times (0,1,1)_{24}$ model: part 2.

No	pdq	PDQ	S	m	n	*nmse* cluster1	*nmse* cluster2	*nmse* cluster3	*nmse* aggregate	*nmse* clusters	*nmse* optimized
1	201	011	168	840	24	0.2540	1.033	0.1483	0.1014	0.103	0.1815
2	201	011	168	840	48	0.394	0.9021	0.2578	0.2408	0.2434	n/a
3	201	011	168	840	72	0.3337	0.9385	0.2478	0.2437	0.2471	n/a
4	201	011	168	840	96	0.2516	0.7673	0.25	0.2234	0.2246	n/a
5	201	011	168	840	120	0.2283	0.5695	0.2467	0.2022	0.2022	n/a
(6)	201	011	168	840	144	0.2486	0.5127	0.2358	0.1979	(0.1952)	n/a
(7)	201	011	168	840	168	0.2407	0.5249	0.2231	0.1873	(0.1839)	n/a
(8)	201	011	168	840	336	0.3393	0.5027	0.2191	0.2200	(0.2172)	n/a
(9)	201	011	168	840	504	0.4443	0.4479	0.2022	0.2179	(0.2153)	n/a
(10)	201	011	168	1008	24	0.4441	0.3912	0.2593	0.2659	(0.2468)	n/a
(11)	201	011	168	1008	48	0.6772	0.5235	0.3384	0.4131	(0.3706)	n/a
(12)	201	011	168	1008	72	0.5758	0.6342	0.2783	0.3556	(0.3229)	n/a
(13)	201	011	168	1008	96	0.566	0.6592	0.2435	0.3220	(0.2869)	n/a
(14)	201	011	168	1008	120	0.4547	0.5026	0.2216	0.2805	(0.2578)	n/a
(15)	201	011	168	1008	144	0.4166	0.4694	0.2071	0.2567	(0.2357)	n/a
(16)	201	011	168	1008	168	0.4749	0.4955	0.2109	0.288	(0.2656)	n/a
(17)	201	011	168	1008	336	0.6163	0.466	0.1906	0.2855	(0.2605)	n/a
(18)	201	011	168	1008	504	0.4398	0.4468	0.1900	0.2379	(0.224)	n/a
19	201	011	168	1176	24	3.401	0.7474	0.1688	0.3654	0.5072	0.4369
20	201	011	168	1176	48	2.292	0.8033	0.1279	0.2594	0.3225	0.3512
21	201	011	168	1176	72	3.302	0.8128	0.1615	0.3808	0.4249	0.4328
22	201	011	168	1176	96	1.566	0.9416	0.1529	0.3816	0.4087	0.4099
23	201	011	168	1176	120	1.555	0.5541	0.1549	0.3333	0.3505	0.3434
24	201	011	168	1176	144	1.007	0.4426	0.1438	0.2487	0.2607	0.3012
25	201	011	168	1176	168	0.8061	0.4015	0.1518	0.2237	0.2331	n/a
26	201	011	168	1176	336	0.4268	0.4028	0.1635	0.1905	0.1952	n/a
27	201	011	168	1176	504	0.3842	0.4013	0.1644	0.1871	0.1907	n/a

Table 5.6: Comparison of predictions with $(2,0,1) \times (0,1,1)_{168}$ model: part 1.

No	pdq	PDQ	S	m	n	*nmse* **cluster1**	*nmse* **cluster2**	*nmse* **cluster3**	*nmse* **aggregate**	*nmse* **clusters**	*nmse* **optimized**
(28)	201	011	168	1344	24	0.9896	0.5187	0.1794	0.1357	(0.1226)	n/a
(29)	201	011	168	1344	48	0.9125	0.4876	0.1836	0.1537	(0.1491)	n/a
(30)	201	011	168	1344	72	0.3931	0.5368	0.1684	0.1552	(0.1527)	n/a
(31)	201	011	168	1344	96	0.2521	0.4719	0.1551	0.1515	(0.1498)	n/a
(32)	201	011	168	1344	120	0.2002	0.4435	0.1651	0.1512	(0.1495)	n/a
(33)	201	011	168	1344	144	0.1996	0.4145	0.1717	0.1531	(0.1512)	n/a
(34)	201	011	168	1344	168	0.2695	0.3934	0.1746	0.1794	(0.1772)	n/a
(35)	201	011	168	1344	336	0.2752	0.3797	0.1711	0.1802	(0.1784)	n/a
(36)	201	011	168	1344	504	0.2890	0.3946	0.1559	0.1859	(0.1849)	n/a
37	201	011	168	1512	24	0.4997	1.202	0.1069	0.2106	0.2196	0.2569
38	201	011	168	1512	48	0.5958	1.251	0.1157	0.2078	0.2094	0.2418
(39)	201	011	168	1512	72	0.3954	1.072	0.1399	0.1910	(0.1909)	0.2535
(40)	201	011	168	1512	96	0.3059	0.5117	0.1427	0.1742	(0.1729)	n/a
(41)	201	011	168	1512	120	0.2713	0.3971	0.1393	0.1527	(0.151)	n/a
(42)	201	011	168	1512	144	0.2762	0.3509	0.1636	0.1597	(0.1560)	n/a
(43)	201	011	168	1512	168	0.2790	0.3498	0.1566	0.1633	(0.1589)	n/a
(44)	201	011	168	1512	336	0.2938	0.3742	0.1424	0.1739	(0.1716)	n/a
(45)	201	011	168	1512	504	0.3486	0.3755	0.1555	0.1808	(0.1780)	n/a
(46)	201	011	168	1680	24	0.3677	0.4447	0.1156	0.1321	(0.1298)	n/a
47	201	011	168	1680	48	0.3807	0.4671	0.095	0.1149	0.1168	n/a
48	201	011	168	1680	72	0.2827	0.4244	0.091	0.1068	0.1086	n/a
49	201	011	168	1680	96	0.2818	0.3341	0.1007	0.1094	0.1100	n/a
50	201	011	168	1680	120	0.2528	0.2676	0.1128	0.1236	0.1238	n/a
51	201	011	168	1680	144	0.2590	0.3575	0.1170	0.1627	0.1630	n/a
52	201	011	168	1680	168	0.3013	0.3779	0.1222	0.1745	0.1750	n/a
(53)	201	011	168	1680	336	0.3775	0.3559	0.1517	0.1809	(0.1805)	n/a
54	201	011	168	1680	504	0.3500	0.3455	0.1566	0.1645	0.1654	n/a

Table 5.7: Comparison of predictions with $(2,0,1) \times (0,1,1)_{168}$ model: part 2.

data and the traffic of user clusters. We compared the prediction based on aggregate traffic with cluster-based prediction. Based on our tests, we noted that 57% of the cluster-based prediction performed better than the aggregate traffic prediction with SARIMA model $(2,0,1) \times (0,1,1)_{168}$. With SARIMA model $(2,0,1) \times (0,1,1)_{24}$, cluster-based prediction performs better than prediction based on aggregate traffic in 8 out of 56 tests and 7 optimized cluster-based predictions gave better results too. The advantage of cluster-based traffic prediction is the flexibility of predicting variable number of users and the reduction of the computational cost.

Chapter 6

Conclusion

In this thesis, we proposed a new prediction approach by combining clustering techniques with traditional time series prediction modeling. The new approach has been tested to predict the network traffic from an operational trunked radio system. We analyzed the network traffic data and extracted useful data from the E-Comm network. We explored the effectiveness and usefulness of clustering techniques by applying AutoClass tool and K-means algorithm to classify network talk groups into various clusters based on the users' behavior patterns. To solve the computational cost problem of "bottom-up" approach and the inflexibleness problem of "top-down" approach, we proposed a cluster-based traffic prediction method. We applied the cluster-based SARIMA models and aggregate-traffic-based models to predict the network traffic. The cluster-based prediction method produced comparable prediction results as the prediction based on aggregate network traffic. In our tests with the 168-hour SARIMA model, the cluster-based prediction performs better than the aggregate-traffic-based prediction. With the 24-hour SARIMA model, cluster-based predictions (8 out of 56 tests) and optimized cluster-based prediction (7 out of 56 tests) perform better the aggregate-traffic-based predictions. Furthermore, the cluster-based prediction approach is applicable to networks with variable number of users where the prediction based on aggregate-traffic-based could not be applied. Utilizing the network user clusters indicates a possible prediction approach for operational networks. Our approach may also enable network operators to predict network traffic and may provide

guidance for future network expansion. Another contribution of this research project is the illustration how data mining techniques may be used to help solve practical real-world problems.

We developed database processing and analysis skills while dealing with the 6 Gbyte database. By applying unsupervised classification method on the traffic data, we learned that it is rarely possible to produce useful results without having the domain knowledge. The discovery of important clusters is a process of finding classes, interpreting the results, transforming and/or augmenting the data, and repeating the cycle. The cluster-based prediction model illustrates the application of clustering techniques to traditional network traffic analysis.

6.1 Related and future work

Prior analysis of traffic from a metropolitan-area wireless network and a local-area wireless network indicated the recurring daily user behavior and mobility patterns [6], [7]. Analysis of billing records from a CDPD mobile wireless network also revealed daily and weekly cyclic patterns [8]. The analysis of traffic from a trunked radio network traffic showed that the call holding time distribution is approximately lognormal, while the call inter-arrival time is close to an exponential distribution [11]. Channel utilization and the multi-system call behavior of trunked radio network have been also simulated using OPNET [30] and a customized simulation tool (WarnSim) [31].

We also experimented with a Bayesian network based approach to explore the causal and conditional relationships among the different characteristics of user behavior, such as call duration, number of systems in a call, caller id, and callee id. We used B-course [32], [33] and Tetrad [34], [35] and constructed Bayesian network from the user calling behavior data. Analysis results are presented in Appendix C.

Since we only have three months of traffic data, we were able to extract only the daily and weekly patterns of the user calling behavior. A larger volume of data may enable identifying the monthly behavior patterns. Traffic models could also be compared using simulation tools. This would help verify the prediction results.

Appendix A

Data table, SQL, and R scripts

A.1 Call_Type table

Id	Call_type
0	Group call
1	Individual call
2	Emergency call
3	System call
4	Morse code
5	Test
6	Paging
7	Scramble
8	Group set
9	System log
10	Start emergency
11	Cancel emergency
100	N/A

A.2 SQL scripts for statistical output

Script to compute the average resource consumption for each talk group, in descending order of the number of calls during the 92 days.

```
SELECT callee, agency, sum(n_calls) AS sn,
TRUNCATE( SUM( n_calls * avgRes ) / SUM( n_calls ), 2 ) AS aR,
TRUNCATE( SUM( n_calls * avgDur ) / SUM( n_calls ), 2 ) AS aD,
TRUNCATE( SUM( avgSys * n_calls ) / SUM( n_calls ), 2 ) AS aSys
INTO OUTFILE '/tmp/dump/tg.res.stat' FROM tgStat
GROUP BY callee ORDER BY sn desc;
```

A.3 R scripts for prediction test and result summary

A.3.1 R script for prediction test

```
pred.test.24<-function(data, p=2, d=0, q=1, P=0, D=1, Q=1,
start=240, end=1920, step=240, p.start=24, p.end=168, p.step=24, prefix)
{
result<-list();
counter<-0;
for (m in seq(start, end, step))
{
mm<-as.integer(m);
worked<-0;
for (n in seq(p.start, p.end, p.step))
{
f.name<-paste(prefix, "pred",mm,n,"(",p,d,q,P,D,Q,")-24",sep="_");
cat(counter,":checking file", f.name);
if (file.exists(paste("./pred.test.24/", f.name, sep=""))) {
cat(" ... ... tested already\n");
```

```
worked<-1;
if (file.info(paste("./pred.test.24/", f.name, sep=""))$size != 0) {
load(paste("./pred.test.24/",f.name,sep=""));
x.arima<-result$arima;
x.a.t<-result$m.t;
rm(result);
break;
}
}
}
if (worked == 0) {
cat("building model based on", mm, "data!\n");
x.a.t<-system.time(x.arima<-arima(data[1:mm], order=c(p,d,q),
seasonal=list(order=c(P,D,Q), period=24)));
}
for (n in seq(p.start, p.end, p.step))
{
counter<-counter+1;
cat(counter,":predict",n,"based on",mm, "data !\n");
f.name<-paste(prefix, "pred",mm,n,"(",p,d,q,P,D,Q,")-24",sep="_");
if (file.exists(paste("./pred.test.24/", f.name, sep=""))) {
cat("tested already\n");
next;
}
x.p.t<-system.time(x.pred<-predict(x.arima, n.ahead=n));
x.nmse<-nmse(x.pred$pred[1:n], data[(mm+1):(mm+n)]);
cat("nmse=",x.nmse,"for (",p,d,q,P,D,Q,")-24\n");
result.pred<-x.pred$pred[1:n];
result<-list(par=c(p,d,q,P,D,Q,24,mm,n), arima=x.arima,
pred=result.pred, nmse=x.nmse, m.t=x.a.t, p.t=x.p.t);
save(result, file=paste("./pred.test.24/", f.name, sep=""));
}
}
```

```
}

pred.test.168<-function(data, p=2, d=0, q=1, P=0, D=1, Q=1,
start=840, end=1680, step=168, p.start=24, p.end=168, p.step=24, prefix)
{
result<-list();
counter<-0;
for (m in seq(start, end, step))
{
mm<-as.integer(m);
worked<-0;
for (n in seq(p.start, p.end, p.step))
{
f.name<-paste(prefix, "pred",mm,n,"(",p,d,q,P,D,Q,")-168",sep="_");
cat(counter,":checking file", f.name);
if (file.exists(paste("./pred.test.168/", f.name, sep=""))) {
cat(" ... ... tested already\n");
worked<-1;
if (file.info(paste("./pred.test.168/", f.name, sep=""))$size != 0) {
load(paste("./pred.test.168/",f.name,sep=""));
x.arima<-result$arima;
x.a.t<-result$m.t;
rm(result);
break;
}
}
}
if (worked == 0) {
cat("building model based on", mm, "data!\n");
x.a.t<-system.time(x.arima<-arima(data[1:mm], order=c(p,d,q),
seasonal=list(order=c(P,D,Q), period=168)));
}
for (n in seq(p.start, p.end, p.step))
```

```
{
counter<-counter+1;
cat(counter,":predict",n,"based on",mm, "data !\n");
f.name<-paste(prefix, "pred",mm,n,"(",p,d,q,P,D,Q,")-168",sep="_");
if (file.exists(paste("./pred.test.168/", f.name, sep=""))) {
cat("tested already\n");
next;
}
x.p.t<-system.time(x.pred<-predict(x.arima, n.ahead=n));
x.nmse<-nmse(x.pred$pred[1:n], data[(mm+1):(mm+n)]);
cat("nmse=",x.nmse,"for (",p,d,q,P,D,Q,")-168\n");
result.pred<-x.pred$pred[1:n];
result<-list(par=c(p,d,q,P,D,Q,168,mm,n), arima=x.arima,
pred=result.pred, nmse=x.nmse, m.t=x.a.t, p.t=x.p.t);
save(result, file=paste("./pred.test.168/", f.name, sep=""));
}
for (n in 2:3*168)
{
counter<-counter+1;
cat(counter,":predict",n,"based on",mm, "data !\n");
f.name<-paste(prefix, "pred",mm,n,"(",p,d,q,P,D,Q,")-168",sep="_");
if (file.exists(paste("./pred.test.168/", f.name, sep=""))) {
cat("tested already\n");
next;
}
x.p.t<-system.time(x.pred<-predict(x.arima, n.ahead=n));
x.nmse<-nmse(x.pred$pred[1:n], data[(mm+1):(mm+n)]);
cat("nmse=",x.nmse,"for (",p,d,q,P,D,Q,")-168\n");
result.pred<-x.pred$pred[1:n];
result<-list(par=c(p,d,q,P,D,Q,168,mm,n), arima=x.arima,
pred=result.pred, nmse=x.nmse, m.t=x.a.t, p.t=x.p.t);
save(result, file=paste("./pred.test.168/", f.name, sep=""));
}
```

```
}
}
```

A.3.2 R script used to summarize prediction results

```
pred.summary<-function(path, cluster=0) {
options(digits=8);
path.len<-nchar(path);
output<-file(paste("output/", path, ".summary", sep=""), open="wt");
files<-list.files(path, full.names=TRUE);
cat(file=output, "no", "(p,d,q)x(P,D,Q)-s", "m", "n",
"nmse", "m. time", "p. time\n", sep="\t");
for (i in 1:length(files))
{
cat("loading", files[i], "\n", sep="..");

load(files[i]);
f.par<-result$par;
f.arima<-result$arima;
f.m.t<-result$m.t;
f.p.t<-result$p.t;
f.nmse<-result$nmse;
rm(result);
if (cluster) {
cat(file=output, substr(files[i], path.len+2, path.len+3),
sep="");
} else {
cat(file=output, i, sep="");
}
cat(file=output, "\t(", f.par[1], ",", f.par[2], ",", f.par[3],
```

```
")x(", f.par[4], ",", f.par[5], f.par[6], ")-", f.par[7],
"\t", f.par[8], "\t", f.par[9], "\t", f.nmse, "\t", f.m.t[3],
"\t", f.p.t[3], "\n", sep="");

cat("finished", files[i], "\n", sep="..");
}
flush(output);
close(output);
}

db.output<-function(dir) {
files<-list.files(dir);
output<-file("db.out", open="at");
type<-c("\'centroid\'", "\'medioid\'");

for (i in 1:length(files)) {
cat("working on", files[i], "\n");
file<-paste(dir, "/", files[i], sep="");
cluster<-length(grep("kc3", files[i]));
pdq<-substr(files[i], nchar(files[i]) - 10, nchar(files[i])-8);
season<-substr(files[i], 11, regexpr("-", files[i])[1]-1);
med<-type[length(grep("med", files[i]))+1];

cat("pdq:", pdq, "season:", season, "med:", med, "\n");

if (cluster) {
input<-scan(file, what=list('character', 'character', 'integer',
'integer', 'numeric', 'numeric', 'numeric'), skip=1);
} else {
input<-scan(file, what=list('integer', 'character', 'integer',
'integer', 'numeric', 'numeric', 'numeric'), skip=1);
}
for (i in 1:length(input[[1]])) {
```

```
if (cluster) {
cat(file=output, "INSERT INTO prediction (cluster, type, pdq,
season, m, n,
nmse, m_time, p_time) VALUES (",
paste(substr(input[[1]][i], 2, 2), med, pdq, season,
input[[3]][i], input[[4]][i], input[[5]][i], input[[6]][i],
input[[7]][i], sep=", "), ");\n", sep=" ");
} else {
cat(file=output, "INSERT INTO prediction (pdq, season, m, n,
nmse, m_time, p_time) VALUES (", paste(pdq, season,
input[[3]][i], input[[4]][i], input[[5]][i], input[[6]][i],
input[[7]][i], sep=","), ");\n", sep=" ");
}
}
}

flush(output);
close(output);
}
```

Appendix B

AutoClass files

B.1 AutoClass model file

The first column of the data file is talk group id. It should be ignored in finding the cluters. The remaining columns use single_normal_cn model.

```
#Leo Chen, 2003-Sep-15
#the model file for E-Comm data user clustering

model_index 0 2
ignore 0
single_normal_cn default

;; single_normal_cm
;; single_multinomial
```

B.2 AutoClass influence factor report

```
#DATA_CLSF_HEADER
#AutoClass CLASSIFICATION for the 617 cases in
#/e-comm/clustering/tg.h/2208/tg2208.db2
#/e-comm/clustering/tg.h/2208/tg2208.hd2
```

```
#with log-A<X/H> (approximate marginal likelihood) = -6548529.227
#from classification results file
#/e-comm/clustering/tg.h/2208/tg2208.results-bin
#and using models
#/e-comm/clustering/tg.h/2208/tg2208.model - index = 0

#DATA_SEARCH_SUMMARY
#SEARCH SUMMARY 1110 tries over  19 hours 17 minutes 49 seconds
#SUMMARY OF 10 BEST RESULTS
#PROBABILITY exp(-6548529.230) N_CLASSES 24 FOUND ON TRY 653 *SAVED* -1
#PROBABILITY exp(-6592578.320) N_CLASSES 18 FOUND ON TRY 930 *SAVED* -2
#PROBABILITY exp(-6619633.090) N_CLASSES 21 FOUND ON TRY 940
#PROBABILITY exp(-6622783.940) N_CLASSES 24 FOUND ON TRY 323
#PROBABILITY exp(-6626274.570) N_CLASSES 17 FOUND ON TRY 542
#PROBABILITY exp(-6637269.320) N_CLASSES 24 FOUND ON TRY 1084
#PROBABILITY exp(-6657627.910) N_CLASSES 18 FOUND ON TRY 677
#PROBABILITY exp(-6658596.390) N_CLASSES 19 FOUND ON TRY 918
#PROBABILITY exp(-6660040.920) N_CLASSES 18 FOUND ON TRY 528
#PROBABILITY exp(-6671271.570) N_CLASSES 12 FOUND ON TRY 385


DATA_POP_CLASSES
#CLASSIFICATION HAS 24 POPULATED CLASSES
(max global influence value = 10.988)
#Class Log of class    Relative     Class   Normalized
#num   strength    class strength   weight  class weight
00    -8.23e+03        0.00e+00     144     0.233
01    -2.16e+04        0.00e+00     67      0.109
02    -7.69e+03        0.00e+00     66      0.107
03    -7.54e+03        0.00e+00     31      0.050
04    -7.54e+03        0.00e+00     25      0.041
05    -1.48e+04        0.00e+00     23      0.037
06    -6.95e+03        1.00e+00     22      0.036
```

07	-1.73e+04	0.00e+00	21	0.034
08	-7.20e+03	0.00e+00	20	0.032
09	-1.13e+04	0.00e+00	20	0.032
10	-7.47e+03	0.00e+00	19	0.031
11	-7.62e+03	0.00e+00	19	0.031
12	-1.65e+04	0.00e+00	18	0.029
13	-1.16e+04	0.00e+00	18	0.029
14	-7.60e+03	0.00e+00	18	0.029
15	-7.47e+03	0.00e+00	17	0.028
16	-1.17e+04	0.00e+00	15	0.024
17	-6.98e+03	2.22e-13	13	0.021
18	-8.23e+03	0.00e+00	12	0.019
19	-1.84e+04	0.00e+00	10	0.016
20	-7.14e+03	0.00e+00	9	0.015
21	-1.05e+04	0.00e+00	4	0.006
22	-8.68e+03	0.00e+00	3	0.005
23	-7.37e+03	0.00e+00	3	0.005

DATA_CLASS_DIVS

#CLASS DIVERGENCES

#Class #num	(class cross entropy) (w.r.t. global class)	Class weight	Normalized class weight
00	1.11e+04	144	0.233
01	5.58e+03	67	0.109
02	1.17e+04	66	0.107
03	1.19e+04	31	0.050
04	1.19e+04	25	0.041
05	5.03e+03	23	0.037
06	1.25e+04	22	0.036
07	2.87e+03	21	0.034
08	1.22e+04	20	0.032
09	8.31e+03	20	0.032

10	1.20e+04	19	0.031
11	1.18e+04	19	0.031
12	3.88e+03	18	0.029
13	8.04e+03	18	0.029
14	1.19e+04	18	0.029
15	1.20e+04	17	0.028
16	8.03e+03	15	0.024
17	1.25e+04	13	0.021
18	1.13e+04	12	0.019
19	2.62e+03	10	0.016
20	1.24e+04	9	0.015
21	9.85e+03	4	0.006
22	1.18e+04	3	0.005
23	1.25e+04	3	0.005

DATA_NORM_INF_VALS

#ORDERED LIST OF NORMALIZED ATTRIBUTE INFLUENCE VALUES SUMMED OVER ALL CLASSES

#	num	description	I-*k
4335	Log NC[81]		1.000
2490	Log NC[1926]		0.999
2986	Log NC[1430]		0.998
4039	Log NC[377]		0.998
3732	Log NC[684]		0.998
3832	Log NC[584]		0.996
3184	Log NC[1232]		0.996
3831	Log NC[585]		0.995
4043	Log NC[373]		0.992
3927	Log NC[489]		0.992
2487	Log NC[1929]		0.992
4209	Log NC[207]		0.991
2506	Log NC[1910]		0.991
3804	Log NC[612]		0.990

```
2485 Log NC[1931]                    0.990
3829 Log NC[587]                     0.990
3948 Log NC[468]                     0.988
3588 Log NC[828]                     0.987
3013 Log NC[1403]                    0.987
3158 Log NC[1258]                    0.987
2700 Log NC[1716]                    0.986
3949 Log NC[467]                     0.986
2699 Log NC[1717]                    0.985
```

B.3 AutoClass class membership report

```
# CROSS REFERENCE CLASS => CASE NUMBER MEMBERSHIP
#DATA_CLSF_HEADER
# AutoClass CLASSIFICATION for the 617 cases in
# /e-comm/clustering/tg.h/2208/tg2208.db2
# /e-comm/clustering/tg.h/2208/tg2208.hd2
# with log-A<X/H> (approximate marginal likelihood) = -6548529.227
# from classification results file
# /e-comm/clustering/tg.h/2208/tg2208.results-bin
# and using models
# /e-comm/clustering/tg.h/2208/tg2208.model - index = 0


DATA_CLASS 0
#  CLASS = 0
#Case talkgroup NC[0] NC[1] NC[2] NC[3] NC[4] NC[5] NC[6] NC[7]
NC[8] NC[9] NC[10] NC[11] (Cls Prob)
008 1099.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
009 1100.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
015 1129.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
016 112.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
```

```
... ...

DATA_CLASS 1
#  CLASS = 1
#Case talkgroup NC[0] NC[1] NC[2] NC[3] NC[4] NC[5] NC[6] NC[7]
NC[8] NC[9] NC[10] NC[11] (Cls Prob)
001 0.hour.nc 26 25 24 24 24 25 25 26 26 25 25 23 1.000
003 1089.hour.nc 6 29 0 0 0 8 0 0 2 80 0 0 1.000
098 1429.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
102 1441.hour.nc 41 9 13 8 15 44 88 9 26 25 21 25 1.000
109 1474.hour.nc 31 13 34 35 10 19 30 8 17 7 4 14 1.000
... ...

DATA_CLASS 2
#  CLASS = 2
#Case talkgroup NC[0] NC[1] NC[2] NC[3] NC[4] NC[5] NC[6] NC[7]
NC[8] NC[9] NC[10] NC[11] (Cls Prob)
033 12118.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
039 12252.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
066 12858.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
069 12872.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
080 13405.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
087 13905.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
094 13931.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
... ...

DATA_CLASS 3
#  CLASS = 3
#Case talkgroup NC[0] NC[1] NC[2] NC[3] NC[4] NC[5] NC[6] NC[7]
NC[8] NC[9] NC[10] NC[11] (Cls Prob)
014 1123.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
059 12708.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
073 12913.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
```

```
081 13427.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
082 13473.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
096 139.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
116 1480.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
118 1491.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
11 7076.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
522 8013.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
524 8021.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
580 8705.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 0.964 4 0.036
... ...

DATA_CLASS 4
#  CLASS = 4
#Case talkgroup NC[0] NC[1] NC[2] NC[3] NC[4] NC[5] NC[6] NC[7]
NC[8] NC[9] NC[10] NC[11] (Cls Prob)
006 1097.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
022 1145.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
023 1146.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
058 126.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
074 13186.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
... ...

DATA_CLASS 5
#  CLASS = 5
#Case talkgroup NC[0] NC[1] NC[2] NC[3] NC[4] NC[5] NC[6] NC[7]
NC[8] NC[9] NC[10] NC[11] (Cls Prob)
005 1091.hour.nc 0 0 0 0 4 0 0 0 0 0 0 0 1.000
020 113.hour.nc 3 0 0 0 0 0 0 0 0 0 0 0 1.000
108 146.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
232 163.hour.nc 0 0 0 0 0 0 0 0 0 0 0 0 1.000
... ...
```

Appendix C

Bayesian network analysis

C.1 B-Course analysis

Conditional dependency analysis results from B-Course are shown in Figures C.1 and C.2.

C.2 Tetrad analysis

Bayesian network analysis results from Tetrad are shown in Figures C.3 and C.4.

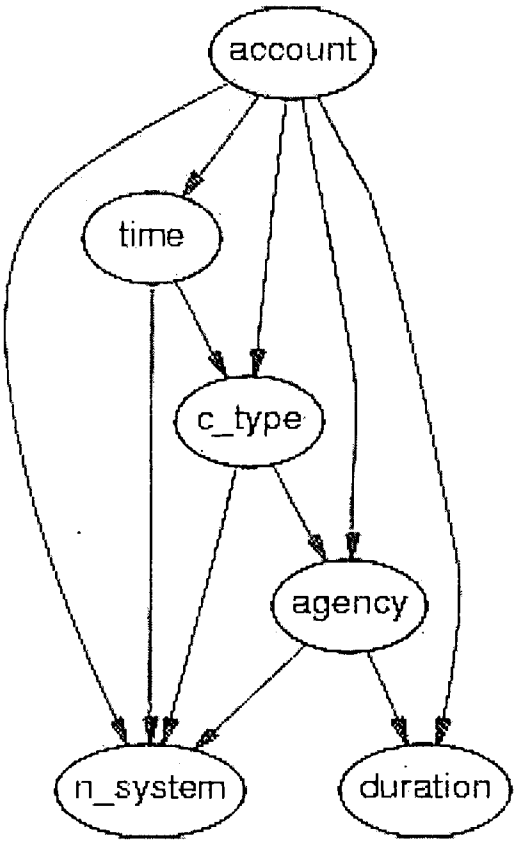

Figure C.1: B-Course analysis: result 1.

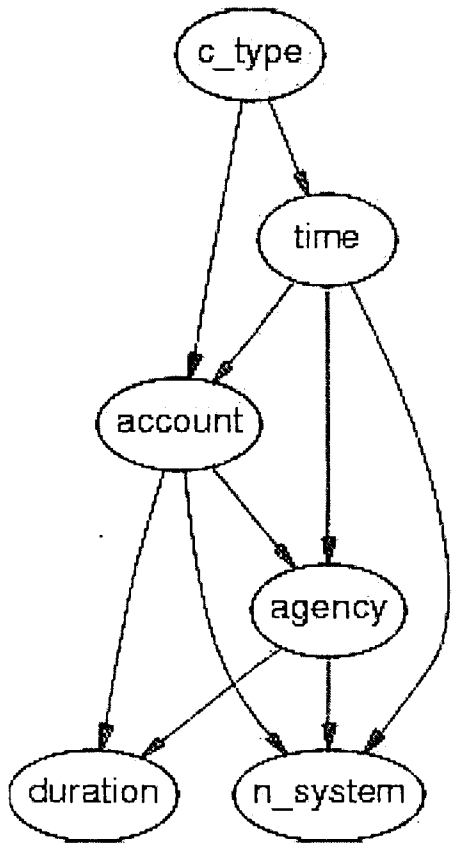

Figure C.2: B-Course analysis: result 2.

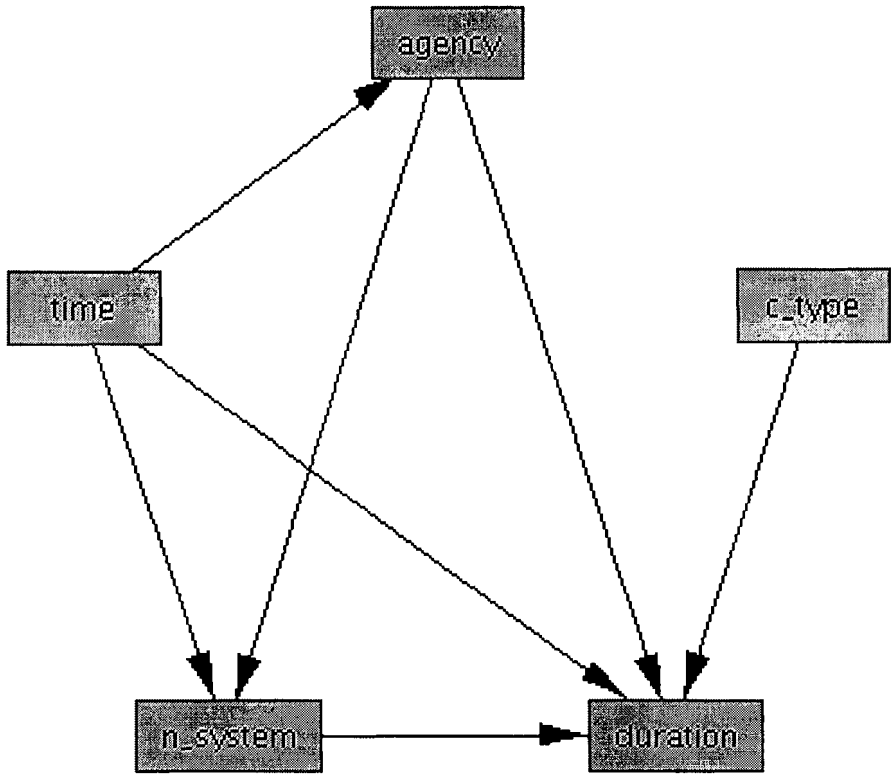

Figure C.3: Tetrad analysis: result 1.

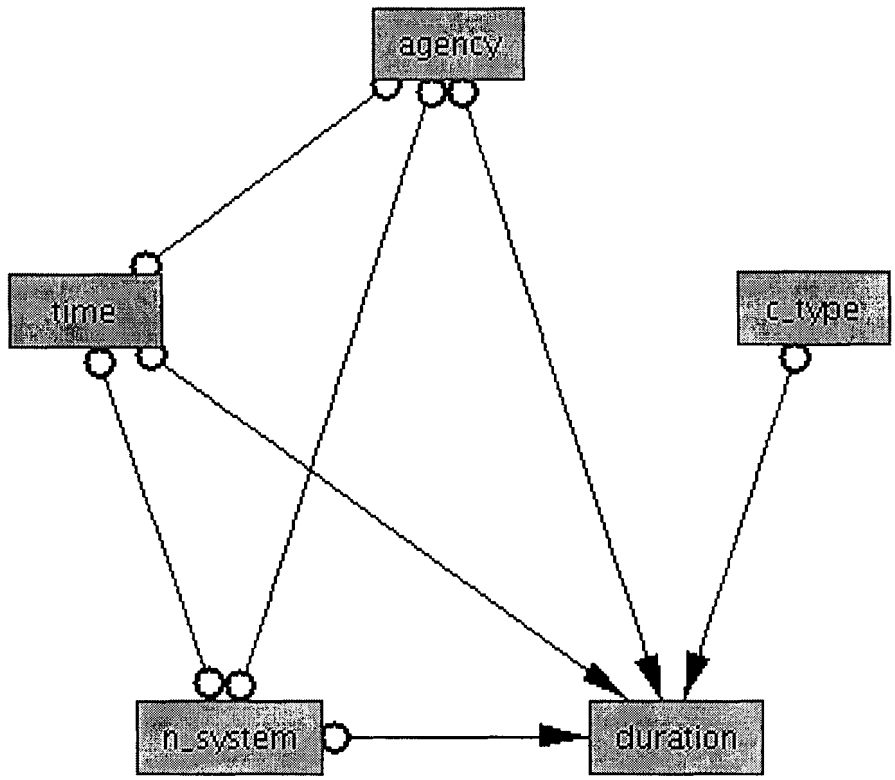

Figure C.4: Tetrad analysis: result 2.

Bibliography

[1] E-Comm - Emergency Communications for SW British Columbia. [Online]. Available: http://www.ecomm.bc.ca.

[2] Welcome to Translink. [Online]. Available: http://www.translink.bc.ca.

[3] M/A-COM - supplier of RF and microwave materials, devices, semiconductors, components and wireless systems. [Online]. Available: http://www.macom.com.

[4] EDACS Explained. [Online]. Available: http://www.trunkedradio.net/trunked/edacs/EDACS_Whitepaper.pdf.

[5] MySQL: The World's Most Popular Open Source Database. [Online]. Available: http://www.mysql.com.

[6] D. Tang and M. Baker, "Analysis of a metropolitan-area wireless network," *Wirel. Netw.*, vol. 8, no. 2/3, pp. 107–120, March-May 2002.

[7] ——, "Analysis of a local-area wireless network," in *Proc. of the 6th Annual International Conference on Mobile Computing and Networking.* ACM Press, 2000, pp. 1–10.

[8] L. A. Andriantiatsaholiniaina and L. Trajković, "Analysis of user behavior from billing records of a CDPD wireless network," in *Proc. of Wireless Local Networks (WLN) 2002*, Tampa, FL, November 2002, pp. 781–790.

[9] P. Cheeseman and J. Stutz, "Bayesian classification (AutoClass): theory and results," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. AAAI Press/MIT Press, 1996, pp. 61–83.

[10] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.* New York, NY: Wiley-Interscience, 1990.

[11] D. Sharp, N. Cackov, N. Laskoviċ, Q. Shao, and L. Trajkoviċ, "Analysis of public safety traffic on trunked land mobile radio systems," *JSAC Special Issue on Quality of Service Delivery in Variable Topology Networks*, vol. 22, no. 7, pp. 1197–1205, September 2004.

[12] The AutoClass Project. [Online]. Available: http://ic.arc.nasa.gov/ic/projects/bayes-group/autoclass.

[13] G. McLachlan and D. Peel, *Finite Mixture Models.* Wiley-Interscience, 2000.

[14] NASA - Computational Sciences Division. [Online]. Available: http://ic.arc.nasa.gov/ic/projects/bayes-group/index.html.

[15] R. Hansaon, J. Stutz, and P. Cheeseman, "Bayesian classification theory," NASA Ames Research Center, Artificial Intelligence Branch, Tech. Rep. FIA-90-12-7-01, 1991.

[16] The Perl Directory. [Online]. Available: http://www.perl.org.

[17] C. J. Van Rijsbergen, *Information Retrieval*, 2nd ed. London, Butterworths: Department of Computer Science, University of Glasgow, 1979. [Online]. Available: http://citeseer.ist.psu.edu/vanrijsbergen79information.html

[18] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstruction of phylogenetic trees," *Mol. Biol. Evol.*, vol. 8, pp. 406–425, 1987.

[19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*, August 1996, pp. 226–231.

[20] What is an Erlang. [Online]. Available: http://erlang.com/whatis.html.

[21] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control.* San Francisco, CA: Holden-Day, 1976.

[22] N. K. Groschwitz and G. C. Polyzos, "A time series model of long-term NSFNET backbone traffic," in *Proc. of the IEEE International Conference on Communications (ICC'94)*, vol. 3, May 1994, pp. 1400–1404.

[23] N. H. Chan, *Time Series: Applications to Finance.* New York, NY: Wiley-Interscience, 2002.

[24] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, 2nd ed. New York, NY: Springer-Verlag, 2002.

[25] C. Chatfield, *The Analysis of Time Series: An Introduction*, 6th ed. Boca Raton, FL: Chapman and Hall/CRC, 2003.

[26] P-Value. [Online]. Available: http://www.isixsigma.com/dictionary/P-Value-301.htm.

[27] R Development Core Team, *R: a language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2004. [Online]. Available: http://www.R-project.org.

[28] R. Ihaka and R. Gentleman, "R: a language for data analysis and graphics," *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, pp. 299–314, 1996.

[29] B. D. Ripley, "The R project in statistical computing," *MSOR Connections. The newsletter of the LTSN Maths, Stats & OR Network.*, vol. 1, no. 1, pp. 23–25, February 2001. [Online]. Available: http://ltsn.mathstore.ac.uk/newsletter/feb2001/pdf/rproject.pdf

[30] N. Cackov, B. Vujičić, V. S., and L. Trajković, "Using network activity data to model the utilization of a trunked radio system," in *Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04)*, San Jose, CA, July 2004, pp. 517–524.

[31] J. Song and L. Trajković, "Modeling and performance analysis of public safety wireless networks," in *Proc. of The First IEEE International Workshop on Radio Resource Management for Wireless Cellular Networks (RRM-WCN)*, Phoenix, AZ, April 2005, pp. 567–572.

[32] P. Myllymäki, T. Silander, H. Tirri, and P. Uronen, "B-course: A web-based tool for Bayesian and causal data analysis," *International Journal on Artificial Intelligence Tools*, vol. 11, no. 3, pp. 369–387, December 2002.

[33] B-Course. [Online]. Available: http://b-course.hiit.fi.

[34] Tetrad Project Homepage. [Online]. Available: http://www.phil.cmu.edu/projects/tetrad.

[35] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, 2nd ed. Cambridge, MA: MIT Press, 2000.

[36] J. Han and M. Kamber, *Data Mining: Concepts and Techniques.* San Francisco, CA: Morgan Kaufmann Publishers, 2001.

[37] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions.* New York; Chichester: Wiley-Interscience, 1997.

[38] R. Scheines, P. Spirtes, C. Glymour, C. Meek, and T. Richard, "The tetrad project: Constraint based aids to model specification," *Multivariate Behavioral Research*, vol. 33, no. 1, pp. 65–118, 1998.

[39] Y.-W. Chen, "Traffic behavior analysis and modeling of sub-networks," *International Journal of Network Management*, vol. 12, no. 5, pp. 323–330, September 2002.

[40] P. Cheeseman, J. Kelly, M. Self, J. Stuze, W. Taylor, and D. Freeman, "Autoclass: A Bayesian classification system," in *Proc. of the 5th International Conference on Machine Learning*, Ann Arbor, MI, June 1988, pp. 54–64.

[41] Making Networks and Applications Perform. [Online]. Available: http://www.opnet.com.