Network mining for managing a broadband network

M. Chartrand\textsuperscript{1} & J. Johnson\textsuperscript{2}
\textsuperscript{1}Department of Mathematics and Computer Science, Laurentian University, Canada
\textsuperscript{2}Department of Mathematics and Computer Science, Laurentian University, Canada

Abstract

By "Network Mining" we mean Data Mining of data available within a telecommunications network environment. Real world data obtained from the local Cable Internet provider are analysed to find rules to help better manage the network. This large dataset helps validate the use of data mining techniques for rule generation. Of the rules generated, to be passed on to the network managers, some proved to be interesting.

1 Introduction

Data mining is a step within Knowledge Discovery in Databases. KDD will be used to permit the network system to provide the human system administrator with information, to permit him or her to more effectively manage the system. Our goal is to use data available from the system and get some sense of it; to show that, from a humanly unmanageable data store, useful knowledge can automatically be obtained to manage the network.

2 Managing a broadband network

Our problem is the management of a broadband network (at the time, that of Regional CableSystems, now known as Persona Communications Inc.) The system is charged with providing Internet or point to point services to business and residential users. The system spans a variety of urban and rural areas, using
both new and old equipment and must withstand a variety of temperature and environmental changes while still providing a reliable service. The available data include, but are not limited to:

- unit IP address
- unit type
- unit uptime
- unit total downstream bytes transferred
- unit total upstream bytes transferred
- unit total downstream error bytes
- unit total upstream error bytes
- modem status
- modem transmit power level
- modem receive power level
- modem signal to noise ratio
- modem maximum upstream bandwidth
- modem maximum downstream bandwidth
- modem CMTS (which router it is attached to)

Application of data mining (DM) to this particular network environment yields the following additional benefits:

1. Data are organized into a data warehouse format that makes the data more accessible and consistent.
2. A better understanding of the system and relationships between data are provided by a discretization process sufficient for soft computing.
3. A fine-tuning of the data cleaning process has led to a better data warehouse. Some data were accurate and complete enough to be reliable while other data were not usable since they required too much cleaning.

At Regional Cablesystems and at companies in general, network problems occur outside of regular business hours. If technicians who maintain these networks can be forewarned by the system, problems can be solved at more appropriate and cost effective times. When a system fault occurs, it would be helpful if the system had already generated rules that led the technician to know where problems are occurring in the network. Time to diagnose and fix problems is greatly reduced because the system has already diagnosed itself. The techniques advanced here provide a step towards this goal.

2.1 How are rough sets used?

Soft computing techniques include rough sets and fuzzy sets. We use a well known data mining technique based on rough sets. A rough set is a pair of sets (lower approximation, upper approximation) used to approximate a concept. Consider Figure 1 and Table 1 to roughly define the concept of a modem being online.
Table 1: Decision table.

<table>
<thead>
<tr>
<th>Modem</th>
<th>Transmit power level</th>
<th>Receive power level</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>Good</td>
<td>Good</td>
<td>Online</td>
</tr>
<tr>
<td>M1</td>
<td>Good</td>
<td>Bad</td>
<td>Offline</td>
</tr>
<tr>
<td>M2</td>
<td>Fair</td>
<td>Good</td>
<td>Online</td>
</tr>
<tr>
<td>M3</td>
<td>Good</td>
<td>Fair</td>
<td>Online</td>
</tr>
<tr>
<td>M4</td>
<td>Good</td>
<td>Fair</td>
<td>Offline</td>
</tr>
<tr>
<td>M5</td>
<td>Fair</td>
<td>Good</td>
<td>Offline</td>
</tr>
<tr>
<td>M6</td>
<td>Bad</td>
<td>Bad</td>
<td>Offline</td>
</tr>
<tr>
<td>M7</td>
<td>Bad</td>
<td>Good</td>
<td>Offline</td>
</tr>
<tr>
<td>M8</td>
<td>Fair</td>
<td>Fair</td>
<td>Online</td>
</tr>
<tr>
<td>M9</td>
<td>Fair</td>
<td>Fair</td>
<td>Offline</td>
</tr>
<tr>
<td>M10</td>
<td>Fair</td>
<td>Bad</td>
<td>Offline</td>
</tr>
<tr>
<td>M11</td>
<td>Bad</td>
<td>Fair</td>
<td>Offline</td>
</tr>
</tbody>
</table>

Figure 1: Graphical representation of a rough set: Domain: — — —; Set: - - - - -; Element of set: Mx (white text); Non-element of set: Mx (black text); Negative region: ; Lower approximation (positive region): ; Upper approximation: & .

Certain or precise rules are generated from the lower approximation and uncertain or imprecise rules from the upper approximation. A rule is interpreted as condition attributes $\Rightarrow$ decision attributes.
In our implementation it is possible to experiment by choosing combinations of attributes from an information table as the condition attributes and the remaining attributes as decision attributes.

A key factor in the success of the Rough Set technique is, as discussed in [2, pp.108-109; 4, pp.102-103], discretization. To maintain objectivity, it is necessary to move away from the use of an expert’s subjective judgements in forming an information table from the input data [4, pp.102]. A preferable technique would involve having the data themselves define the discrete regions. Further issues, which form the key to Rough Set theory’s success, can be found in [5, pp.92].

The rules, such as the following, which were produced by the RS1 implementation proved very easy for us to understand however such would probably not be the case for those who would normally require or use the produced information such as managers, administrators and technicians.

Rules generated for class [start ].
Rule [upcappartofmonth=start ’] coverage [0.14] support [0.13], boundary width [0.010000000000000009].

The creation of an interpreter to translate these rules into predicate calculus format, (e.g.,)

\[ \text{UpCapPartOfMonth(start) } \Rightarrow \text{DownCapPartOfMonth(start)} \]

92.9% degree of certainty

that is much easier to understand, could be done very easily and would make the produced knowledge more attainable to the average user. In this paper, we will present the interpreted rules as much as possible.

3 Methodology - network mining

Knowledge Discovery in Databases (KDD) will be used with particular attention on Data Mining using a Rough Sets approach to obtain knowledge toward better management of Regional Cablesystems through the use of its network data. By the term "Network Mining we mean Data Mining of data available within a telecommunications network environment. To apply Rough Set theory within our environment it was decided to use an existing implementation of the RS1 inductive learning algorithm [9] written in Java by Warren [6, 7].

Data cleaning and pre-processing were done as part of the data acquisition process to ensure that the data were correct and complete. Due to the size of the dataset and number of sample readings, instances with missing data were simply discarded. Since the network is strictly defined by DOCSIS (Data Over Cable System Interface Specification), outliers represent erroneous readings which must also be discarded since meaning cannot be taken from their fallacious nature. Our Data Mining algorithm, by its very nature, provides data reduction by minimizing the number of attributes from the information table used in the rules.
A Data Warehouse was created to alleviate the problem of data being kept on a variety of non-compatible systems. Data from modems and CMTS (Cable Modem Termination Systems), Cisco’s proprietary modem management software: CNR (Cisco Network Registrar), the company’s client information database: RR, and expert data were all stored in one central data repository.

As mentioned the cable modem network is regulated by DOCSIS. Furthermore, to manage traffic on the network, each modem is given a QoS (Quality of Service). Data which fall in these two areas are therefore discretized according to the discrete regions formed within these rules. For example, modem TX (Transmit Power), RX (Receive Power), and SNR (Signal to Noise Ratio) are either good or bad. A modem has either exceeded its maximum allowable traffic, or it has not.

We wish to find rules for predicting the state of a network based on data about the network. The rules originate from the minimal set of attributes for an information table including Table 1 (third page of this article) and those containing attributes QOS, ISP, SNR, Uptime which are described in Appendix A. For deterministic rules our results will be of the form:

\[
\text{TransmitLevel(bad)} \land \text{ReceiveLevel(bad)} \Rightarrow \text{ModemStatus\text{(offline)}}
\]

_modems having a bad transmit power level and a bad receive power level will be offline._

\[
\text{QOS\text{(Business entry)}} \Rightarrow \text{ISP\text{(Vianet)}}
\]

_modems having a “Business Entry” quality of service will have “Vianet” as their Internet Service Provider._

\[
\text{SNR\text{(bad)}} \Rightarrow \text{Uptime\text{(<hour)}}
\]

_modems having a bad transmit power level and a bad receive power level will be offline._

For non-deterministic or uncertain rules, we wish to calculate a weight to measure their influence on the environment. We will show how the weights (or strength of a rule) are calculated shortly, but first observe a few examples of uncertain rules which are of the form:

\[
\text{TransmitLevel(bad)} \land \text{ReceiveLevel(bad)} \Rightarrow \text{SNR\text{(bad)}}\text{, 50\% degree of certainty}
\]

_we can be 50\% confident that a modem having a bad transmit power level and a bad receive power level will have a bad signal to noise ratio._

\[
\text{QOS\text{(residential Gold 1PC)}} \Rightarrow \text{ISP\text{(AOL)}}\text{, 40\% degree of certainty}
\]

_we can be 40\% confident that a modem having a “Residential Gold” quality of service will have America Online as its Internet Service Provider._

\[
\text{SNR\text{(good)}} \Rightarrow \text{Uptime\text{(month)}}\text{, 20\% degree of certainty}
\]

_we can be 20\% confident that a modem with a good signal to noise ratio will have been operational for more than one month._
To calculate weights we need to define a few terms. We use coverage and support as defined in the literature. [6, pp.561 From there we define “False Support” which measures the number of examples within the decision table that are covered by the rule, but whose class is different from the one that the rule is supposed to represent” [7]:

\[
\text{False Support} = \text{Coverage} - \text{Support}
\]

False Support is a ratio of the elements of a set and a given attribute value which, for a given rule, falls within the boundary area. Deterministic rules, whose elements are wholly contained within the lower approximation and thus have no elements present within the boundary area, will therefore have a false support value of zero. Non-deterministic rules will be represented by false support ratios other than zero.

We can therefore, using a rule’s coverage, support, and false support values, establish the level of importance each rule has with respect to the entire set. Rules with small coverage can be considered in the context of affecting only a small portion of the set. A false support value of zero or near zero denotes a rule that can be used with relative certainty. Conversely false support values closer to one are indicative of rules which are not very, if at all, reliable.

3.1 RS1 preparation and implementation

The RS1 implementation by Warren [7] was written in Java and uses JDBC which should have made it portable from the UNIX / Postgres environment to the Windows 2000 / SQL Server environment in which our data are present. This however was not the case. Although the Java code was oblivious to the transition, many of the SQL statements used to interact with the database were incompatible. This is due to the fact that Postgres implements an extended subset of the SQL92 and SQL3 languages while SQL Server implements Transact-SQL. Since the program could not be brought to the data it was decided to bring the data to the program. The dataset was therefore re-created in a different environment. The SQL Server dataset was exported to a text file and subsequently imported to Postgres for rule generation by Warren's implementation. The program’s operation requires three parameters:

- the name of the table we wish to analyse
- the name of the column which we wish to use as our decision attribute
- the name of a single column which forms a unique identifier for each row/element

The first two parameters could easily be found within our present tables. The third was simply added to datasets in SQL Server, since it would have been needed there if the program had been used in that environment, and so is now also present in the mirrored tables.
The program outputs, to the console, messages to indicate its progress along with the rules it finds and their coverage, support and false support values. Rule generation varied from a few minutes for individual tables with a relatively small number of records, to over 8 hours for joins of various tables. Generated output also varied from a few megabytes of data, to over 300 megabytes. Program output was therefore redirected to files for analysis. The program was run on a DELL Inspiron 7500 with a Pentium III 450 and 128 Megs of RAM. The operating environment consisted of Linux RedHat 7.1 beta, Postgres 7.0, JDK 1.2.2 R4, JDBC 1.2 and Perl 5.6.0.

The RS1 implementation was run on all of the tables in the dataset. Furthermore, to broaden our search for knowledge, each of the columns were used in turn as the decision attribute for all tables but DS_all (a join of all of the tables). See the Appendix A for a description of all of the tables. Although many columns were used as decision attributes in this last table, the prohibitively long processing time caused by its many attributes and examples prevented each column individually from being treated as the decision attribute.

3.2 Rough sets results

The inductive algorithm has provided us with a tremendous number of rules about the communications network environment. The task of sifting through these rules to discover the knowledge they contain is left to the final stages of KDD as it is not part of the Rough Set process.

Examples, from the DS_levels table described in Appendix A, using the transmit power level (Tx) as the decision attribute, as output by the program in raw form are:

Rules generated for class [good ].

***Rule [uptime='month'] total coverage [0.10808656036446469] with support [0.10808656036446469].

***Rule [uptime='day' AND levelweekyear='8'] total coverage [0.046810933940774485] with support [0.046810933940774485].

***Rule [uptime='month' AND levelweekyear='8'] total coverage [0.018223234624145785] with support [0.018223234624145785].

***Rule [uptime='week' AND levelweekyear='8'] total coverage [0.09874715261958998] with support [0.09874715261958998].

The large number of rules output from the RS1 implementation made it virtually impossible to analyse all of them. Moreover, our goal is not to obtain all possible knowledge from the environment, but to obtain knowledge that can help in the management of the system. To simplify this task, our search for knowledge was confined to rules with relatively high coverage and low false support values.

A redundant attribute is one that provides no additional information that the remaining attributes do not already express. From these rules we know that a modem with an Uptime of a month, or more, will have a good Tx (transmit
power level). Since the rule is deterministic, confirmed by the identical coverage and support values, no weight value needs to be given to the rule.

\[ \text{Uptime(month)} \Rightarrow \text{Tx(good)} \]

Modems that have been operational for more than one month will have a good transmit power level.

From RS1 we know that all attributes other than Uptime are redundant for defining the concept Tx when the Uptime value is \( \text{month} \). For other values of Uptime, more attributes are necessary. For example we know that if a modem’s Uptime is a week during the \( 8^{th} \) week of the year, the Tx is good.

\[ \text{Uptime(week)} \land \text{LevelWeekYear(8)} \Rightarrow \text{Tx(good)} \]

Modems that have been operational for more than one week and where the data was obtained on the \( 8^{th} \) week of the year will have a good transmit power level.

Rules of all lengths covering a multitude of attributes were obtained as can be seen below by this raw rule generated from our implementation.

\[ \text{***Rule \{uptime='week' AND levelhour='21to23' AND leveldayweek='7' AND levelweekyear='9' AND rx='good' AND snr='good' AND levelpartofmonth='start' AND levelmonth='3'} \] total coverage \[0.0036446469248291574\] with support \[0.0036446469248291574\].

In fact the majority of the rules produced were of this size and length. However, with their minimal coverage values, such are to be considered exceptions rather than rules. Although knowledge could be obtained from some of them, this is left to further work as we look for more relevant knowledge evidenced by greater coverage.

Thus far we have considered only deterministic rules. Failing to consider non-deterministic rules would be to ignore the vast power that Rough Set theory. The following are examples of rules that have relatively high coverage values, and although they are not deterministic, their low false support (boundary width) values assure us that we can use them with great certainty.

The degree of certainty is obtained as the ratio between support and coverage. By this computation from parameters of the raw rule, uncertain rules such as the following were obtained.

\[ \text{Uptime(day)} \Rightarrow \text{Tx(good)} \quad 93.8\% \text{ degree of certainty} \]

Modems that have been operational for more than one day will have a good transmit power level.

\[ \text{Uptime(week)} \Rightarrow \text{Tx(good)} \quad 97.6\% \text{ degree of certainty} \]

Modems that have been operational for more than one week will have a good transmit power level.

We see here that an Uptime value of a week is sufficient in predicting a good transmit power level particularly when we look at the high degree of certainty and relatively high coverage value (to three decimal places 0.322). This simpler
rule can therefore be more practically used than the complex raw rule above that required eight attributes to come to the same conclusion.

Knowledge was also obtained by using other columns, such as a modem's receive power level, as decision attributes.

***Rule \[\text{uptime} = \text{<hour}>\] total coverage \[0.010592255125284738\] with support \[0.010592255125284738\].

Referring to the rule above, since coverage and support are the same, we know with certainty that modems with an Uptime of < hour have a bad Rx (receive power level). In the following, if a degree of certainty is not given, the rule is certain.

\[\text{Uptime(<hour)} \Rightarrow \text{Rx(bad)}\]

Modems that have been operational for less than one hour will have a bad receive power level.

The following rules give us more examples of knowledge that was obtained using the Rx decision attribute

\[\text{Uptime(hour)} \Rightarrow \text{Rx(good)} \quad 99.2\% \text{ degree of certainty}\]

Modems that have been operational for more than one hour will have a good receive power level.

\[\text{Uptime(month)} \Rightarrow \text{Rx(good)} \quad 99.6\% \text{ degree of certainty}\]

Modems that have been operational for more than one month will have a good receive power level.

The use of Signal to Noise Ratio (SNR) also provided good rules:

\[\text{Uptime(month)} \Rightarrow \text{SNR(good)}\]

Modems that have been operational for more than one month will have a good signal to noise ratio.

\[\text{Uptime(day)} \Rightarrow \text{SNR(good)} \quad 99.1\% \text{ degree of certainty}\]

Modems that have been operational for more than one day will have a good signal to noise ratio.

The DS_details table provides knowledge about modems' Internet Service Provider and Quality of Service. Rules were generated while using ISP as the decision attribute. A translation of the rules generated for class \[105\] follows:

\[\text{QOS(Residential Gold 1 PC updown restrict)} \Rightarrow \text{ISP(Vianet)}\]

Modems that have a "Residential Gold 1 PC" quality of service rated limited in the upstream and downstream directions will have Vianet as their Internet Service Provider (ISP).

\[\text{QOS(Business Entry active)} \Rightarrow \text{ISP(Vianet)}\]

Modems that have a "Business Entry" quality of service and which have not been rate limited in any direction will have Vianet as their ISP.
The next group of rules was obtained by using the Quality of Service (QOS) as the decision attribute. The knowledge obtained from rules generated from class [213] are as follows:

**ISP(AOL) ⇒ QOS(Residential Gold 1 PC active)**

95.2% degree of certainty

Modems whose Internet Service Provider is America Online will have a “Residential Gold 1 PC” quality of service which has not been rate limited in any direction.

**ISP(ON-Link) ⇒ QOS(Residential Gold 1 PC active)**

89.5% degree of certainty

Modems whose Internet Service Provider is ON-Link will have a “Residential Gold 1 PC” quality of service which has not been rate limited in any direction.

The DS_ totals table provides us with knowledge of modems’ total traffic in the upstream and downstream directions for an entire month. It also provides us with the dates when the modem quality of service was changed for exceeding the limits of its former quality of service. The following rules were generated while using the modems total downstream usage as the decision attribute:

**UpCapDayWeek(Sunday) ⇒ DsUsage(over_limit)**

Modems that have been rate limited in the upstream direction on a Sunday will have a downstream usage that exceeds their allotted limit.

**UpCapDayWeek(Monday) ⇒ DsUsage(under_limit)**

80.0% degree of certainty

Modems that have been rate limited in the upstream direction on a Monday will have a downstream usage that does not exceed their allotted limit.

Knowledge was also obtained by using the change of quality of service date. Here is an example of "when the change in QOS occurred" was used as the decision attribute.

**UpCapPartOfMonth(end) ⇒ DownCapPartOfMonth(end)**

66.7% degree of certainty

Modems that have been rate limited in the upstream direction at the end of the month will have been rate limited in the downstream direction at the end of the month.

Other rules of interest are as follows:

**DownCapPartOfMonth(end) ⇒ UpCapPartOfMonth(end)**

Modems that have been rate limited in the downstream direction at the end of the month will have been rate limited in the upstream direction at the end of the month.

**UpCapPartOfMonth(start) ⇒ DownCapPartOfMonth(start)**

92.9% degree of certainty
The DS_client and DS_modem tables contain data that identify the client and modem, respectively. Due to their nature these data remain relatively the same and thus we would only obtain a rule to echo the data for each individual client or modem.

The DS_traffic table held promise to predict trend in usage with respect to time. Such was unfortunately not the case. Rules obtained from this table were generally either boundary cases with a very low degree of certainty, or exceptions with low confidence levels. The lack of available knowledge from this table seems to be due to a lack of sufficient discretization or possibly even improper discretization. Observing the sums of traffic on a given day for a given modem may prove more fruitful than using eight separate values obtained every three hours during that day. Dividing traffic in a manner other than multiples of ten would also surely provide better results.

The DS_all table (not shown) contained data from all the tables except for DS_totals. The data within the DS_totals table did not have an attribute which made it possible to logically link it with the other tables in the dataset. DS_all represents all of the client, modem, level and traffic data at a particular reading instance.

DS_all was a very large which meant that processing the data within it took very long. Processing time to generate rules for this table took upwards of eight hours each, if the program at all managed to complete. This meant that only a few of the attributes were attempted as decision attributes. Those attributes which were attempted are: day_hour, partOfMonth, isp, qos, snr, and uptime. However no new knowledge was provided from this table. The rules that were returned merely echoed the knowledge that was already obtained from the smaller tables.

4 Evaluation

Examples of network management data include when and where traffic is occurring and modems in particular as good indicators of hardware problems. In this study, we found that a modem’s operational time was not a good indicator of bad transmit power levels despite our original beliefs to the contrary. On the other hand, the operational time was shown to be a good indicator of bad transmit power levels. Hence, to our surprise, operational time a good parameter by which to judge the status of the network. Further work is required on presenting the information generated for managing the network in a user-friendly format.

The discovered knowledge will be passed on the managers of the Regional Cablesystems network. It will either provide previously unknown facts or confirm former assumptions. In either case it will help administrators to better understand the communications network and thus help them to better manage it.

The method used generates rules but does not indicate which are interesting or useful to a given problem. Human interpretation is still required to distinguish which are the interesting rules. Most of the rules obtained validated our
intuitions, but such rules were not very useful. Nevertheless some rule, like the following, did provide insightful knowledge.

\( \text{Uptime(<hour)} \Rightarrow \text{Rx(good)} \)

Modems that have been operational for less than one hour will have a good receive power level.

\( \text{Uptime(day)} \Rightarrow \text{SNR(good)} \quad 99.1\% \text{ degree of certainty} \)

Modems that have been operational for more than one day will have a good signal to noise ratio

\( \text{QOS(Residential Gold 1PC updown restrict)} \Rightarrow \text{ISP(Vianet)} \)

Modems that have a "Residential Gold 1 PC" quality of service that has been rate limited in the upstream direction will have Vianet as their ISP

\( \text{QOS(Business Entry active)} \Rightarrow \text{ISP(Vianet)} \)

Modems that have a "Business Entry" quality of service that has not been rate limited in any direction will have Vianet as their Internet Service Provider.

\( \text{UpCapPartOfMonth(start)} \Rightarrow \text{DownCapPartOfMonth(start)} \quad 92.9\% \text{ degree of certainty} \)

Modems that have been rate limited in the upstream direction at the beginning of the month will have been rate limited in the downstream direction at the beginning of the month

A complete list of rules can be found in [1].

Rough Sets theory proved to be the correct choice for our Data Mining algorithm. Given their nature, the data in most attributes were easily discretized. This permitted data from tables with as many as 20000 elements or as many as 23 attributes to be analysed with relative ease and speed.

Further work has involved attempts to make accessible other important sources of data within the Regional Cablesystems environment such as RR and CNR. Our effort was hindered by the lack of any real decision attributes within the data we possessed and by our limited ability to geographically situate hardware within the system. Both of these problems would be resolved with access to these other data sources. Writing an implementation from scratch (pseudocode) of the RS1 algorithm in a faster language such as C++ should be considered as it would yield further time savings.

Although the use of the RS1 was very successful, the use of other complementary algorithms, heuristics or implementations may also prove to be fruitful such as those based on Fuzzy Sets [3] and statistical methods [8].

5 Conclusion

An in depth understanding of the application domain was acquired. The environment now has a central repository of cleaned and processed data in the form of a data warehouse. We have analysed our goal with respect to the system in which new and interesting predictive rules are to be achieved, come up with a
proper algorithm and process, and this process remains to extract further knowledge from this environment. We have obtained rules that either confirm our intuitions or were insightful regarding the telecommunications environment.

We have however, after all this work, only scratched the surface of what is possible. Only a small subset of the entire network, less than 3%, was used to demonstrate the viability of Network Mining. Observing the entire network would give a better representation of it and thus provide even better results. The data that were used only span a period of two months. Data accumulated over a greater period of time would permit trends to be more predominant and reduce the effect of isolated cases having too much effect on the outcome. Rich sources of information, in the form of detailed client information, and geographical and environmental data, were not available at the time of this endeavour. These data relate to factors of great influence on the system. Their presence more than anything else would greatly improve the quality and quantity of knowledge that is obtained.

Appendix A: Metadatabase

Metadatabase which describes the subset of the network management environment at Regional Cablesystems Northern Division which was used for this study.

<table>
<thead>
<tr>
<th>Table</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS_modem</td>
<td>Time relevant modem values that do not relate to levels or traffic.</td>
</tr>
<tr>
<td>DS_levels</td>
<td>At a particular point in time a modem's level.</td>
</tr>
<tr>
<td>DS_traffic</td>
<td>Counter values for each modem for each time the counter was read.</td>
</tr>
<tr>
<td>DS_totals</td>
<td>The total usage in each direction for the current and each previous month.</td>
</tr>
<tr>
<td>DS_details</td>
<td>The current owner of a particular modem and the account's status.</td>
</tr>
<tr>
<td>DS_client</td>
<td>Client information. This is the central table of the database. It contains the information tied to a particular account.</td>
</tr>
<tr>
<td>DS_All</td>
<td>A join of all other tables to give the double benefit of having the largest possible variety of attributes(fields) to compare while also reducing processing speed by eliminating the processing time necessary to join data from various tables.</td>
</tr>
</tbody>
</table>

References


32 Data Mining IV


