Faculty of Engineering and Technology

Master of Computing (MCOM)

Master Thesis

Provenance-based Debugging and Drill-down Approach for Revenue Leakage Detection and Root Cause Analysis: An Application in The Telecom Domain

منهجية التتبع والتنقيب باستخدام مصادر البيانات وأصولها لاكتشاف تسرب الإيرادات وتحليل اسبابه: تطبيق في مجال الإتصالات

Author:

Wisam Al-Abbasi

Supervisor:

Dr. Adel Taweel

This Thesis was submitted in partial fulfilment of the requirements for the Master's Degree in Computing from the Faculty of Graduate Studies at Birzeit University, Palestine.

10th-June-2018

Provenance-based Debugging and Drill-down Approach for Revenue Leakage Detection and Root Cause Analysis: An Application in The Telecom Domain

By: Wisam Al-Abbasi

Approved by the thesis committee

_____

Dr. Adel Taweel, Birzeit University

_____

Dr. Ahmad Alsadeh, Birzeit University

_____

Dr. Samer El-Zain, Birzeit University

Abstract

Revenue Assurance (RA) is considered a top priority function for the telecommunication operators. Revenue leakage, if not prevented, depending on its severity, could cause a significant revenue loss of an operator. Detecting and preventing revenue leakage is a key process to assure the efficiency, accuracy and effectiveness of the telecom systems and processes. There are two general revenue leakage detection approaches: big data analytics and rule-based. Both approaches seek to detect abnormal usage and profit trend behavior and revenue leakage based on certain patterns or predefined rules, however both are mainly human-driven and fail to automatically debug and drill down for root causes of leakage anomalies and issues.

In this thesis, a rule-based RA approach that deploys a provenance-based model is proposed. The model represents the workflow of critical RA functions enriched with contextual and semantic information that may detect critical leakage issues and generate potential leakage alerts. A query model is developed for the provenance model that can be applied over the captured data to automate, facilitate and improve the current process of root cause analysis of revenue leakages.

The proposed approach has been implemented and tested on thirteen revenue leakage scenarios. Using defined root-cause gold standard datasets, these scenarios with 26 revenue leakage symptoms have been used to evaluate and validate the proposed approach in terms of completeness and accuracy. The evaluation results show that the proposed approach can automate the debugging and drill-down of the root causes of these scenarios, and achieves 100% completeness and accuracy for the evaluated root-cause scenarios. However, its accuracy is directly affected by the accuracy of the contextual information and thus must be accurately represented.

الملخص

يمثل توكيد الإيرادات أولوية عظمى لدى أغلب مشغلي الهواتف حول العالم ، فتسرب الإيرادات – إذا لم يتم منعه – قد يتسبب في ضياع إيرادات هامة للمشغل تبعاً لحدة التسرب، مما يؤثر على الربح والإستمرارية. تحديد ومنع تسرب الإيرادات هو عملية هامة لتأكيد عمل أنظمة الإتصالات بفعالية و دقة. هناك منهجان أساسيان للكشف عن تسرب الإيرادات أولهما المنهج القائم على تحليل البيانات الضخمة ، والثاني هو المنهج القائم على القواعد. هذان المنهجان تتم إدارتهما بشرياً ولا يمكن تشغيلهما أوتوماتيكياً للكشف عن جذور التسرب و أسبابه. من هنا استدعت الحاجة للتأكيد على أهمية تطوير منهج أوتوماتيكي للحصول على طريقة لا تقوم فقط بالكشف عن تسرب الإيرادات بل أيضا أسبابه و بسرعةٍ كبيرةٍ.

في هذا البحث ، تم اقتراح استخدام منحى يعتمد على توكيد الإيرادات و يطبق نموذجاً قائماً على مصادر البيانات وأصولها. هذا النموذج يمثل سير عمل مهام حرجة في مهام توكيد الإيرادات معززةً بالمعلومات السياقية و الدلالية اللازمة لتحديد مشاكل تسرب الإيراد و تولد تنبيهات عند وجود تسرب محتمل. تم تطوير نموذج استعلام للنموذج القائم على مصادر البيانات وأصولها والذي يأخذ بعين الاعتبار مشاكل تسرب الإيرادات المحتملة بسيناريوهات تم تعريفها، والتي من الممكن تطبيقها على البيانات التي تم التقاطها لأتمتة و تسهيل و تطوير العملية الحالية للفحص ، من أجل الوصول إلى السبب الرئيسي و التنقيب عن مشاكل تسرب الإيرادات.

تم تطبيق وفحص النموذج المقترح على ثلاثة عشر سيناريو يمثلون تسرب محتمل للإيرادات مع أسبابها الحقيقية كمعيار مثالي ، هذه السيناريوهات مع 26 أعراض على وجود مشاكل تسرب في الإيرادات والتي تم استخدامها لتقييم والتحقق من كفاءة النموذج المقترح من ناحية الإكتمال والدقة. تظهر نتائج الفحص والتحقق أن النموذج المقترح بالإضافة إلى نموذج الإستعلام تمكن و بشكل فعال من كشف السبب الرئيسي الكامن وراء مشكلة تسرب الإيرادات المحتملة لهذه السيناريوهات أوتوماتيكيا عن طريق التعرف على أصول البيانات و مصادرها. في المرحلة القادمة سيتم تطوير النموذج الحالي ليشمل سير عمل نقاط البيانات و معلومات سياقية و ودلالية للتعامل مع سيناريوهات أكثر تعقيداً. أثبت النموذج المقترح اكتماله ودقته من خلال تحقيق نسبة 100% من الاكتمال والدقة خلال مرحلة الفحص. ولكنه من جهة أخرى، يعتمد إلى حد كبير على دقة المعلومات السياقية المتوفرة ليوفر نتائج صحيحة ودقيقة.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Adel Taweel, who has been a role model for me in science. I am grateful for his understanding, academic guidance, and support.
I would like also to thank my parents for their support and encouragement over the years.

Table of Contents

Table of Figures

List of Tables

Abbreviations

ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
APN	Access Point Name
BRMS	Business Rules Management Systems
CCN	Charging Control Node
CDR	Call Detail Record
DISC	Data Intensive Scalable Computing
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
GUI	Graphical User Interface
KPI	Key Performance Indicator
MSC	Mobile Switching Center
MSISDN	Mobile Station International Subscriber Directory Number
OPM	Open Provenance Model
Panda	Provenance and Data
PSTN	Public Switch Telephone Network
RA	Revenue Assurance
RDF	Resource Description Framework
SASN	Service Aware Support Node
SDP	Service Delivery Platform
SGSN	Serving GPRS Support Node
SMS	Short Message Service
SMSC	Short Message Service Center
SPARQL	Simple Protocol And RDF Query Language
SQL	Structured Query Language
UMTS	Universal Mobile Telecommunications System

Chapter 1 Introduction

Current revenue assurance (RA) systems follow two approaches, either big data analytics [60] or rule-based [15]. Big data analytics are useful approaches for understanding streamlining and stored data, but they process only relevant data to specific cases based on patterns. However, these approaches do not consider revenue leakage as a whole from all its aspects and does not include all operational tasks and processes. An example of this approach, which has been developed to detect anomalies related to network issues, was proposed by Zoldi et al. [60]. Whereas there is a need in telecom industry for studying specific cases, beyond general ones, that may cause or lead to particular case of revenue leakage. There are many sources of revenue leakage in the telecom industry, which require multilevel complexity analysis. For example as shown Figure 1, a product that has many offers but includes one offer with a poor design that is comprised of too much details, which would make it very difficult to detect a revenue leakage using a general approach, but it requires approach which covers all these revenue leakage areas.

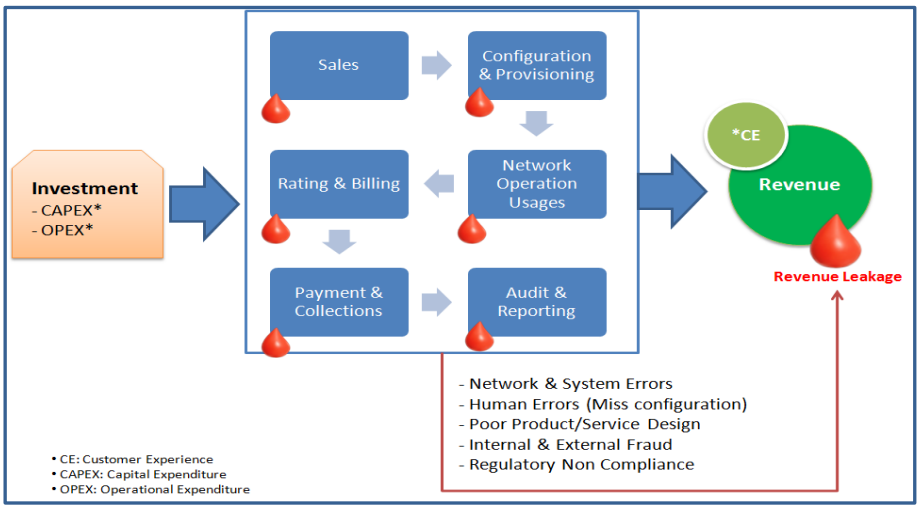


Figure 1 Revenue Leakage Sources in Telecom industry [46]

On the other hand, Rule-based RA systems work based on predefined business rules to capture certain traffic from multiple sources and make data comparison

measurements to detect revenue leakage issues or discrepancies [15]. Fine et al [15] presents a rule-based system in which the defined rules have been set based on the business logic using Business Rules Management Systems (BRMS) [6]. For example, a rule may define that only prepaid subscribers are charged for their usage traffic using the online charging system referred to as the Charging Control Node (CCN), while postpaid subscribers are charged for their usage traffic using the offline billing system. This rule implies that the switch generates voice call records for prepaid and postpaid subscribers, but the online charging system charges and generates records only for prepaid subscribers, while the offline billing system charges and generates records only for postpaid subscribers, breaking this rule indicates that a revenue leakage issue may have been introduced. The main problem with such approaches that they fail to both automatically debug and drill down for root-causes of leakage issues. This thus necessitates the development of an automated approach to address these issues.

The focus of this work is to extend the current general approach of rule-based RA architectures. This is done through the deployment of a data provenance-based approach that utilizes RA functionalities to generate provenance data, which can then be used to achieve debugging and drill down features using back tracing of the functional-workflow till the main source of input data. RA systems provide input data for call and usage, which provide detailed records, from which calls or usage attributes can be used to detect and derive revenue leakage causes.

1.1 *Introduction*

The main responsibility of RA analysts is to manage and prevent revenue leakage based on RA methodology, as shown in Figure 2, [57]. Current RA architectures support revenue leakage detection by applying a series of detective processes consisting of monitoring, summarization, auditing, and investigation [57].

But it is not usually an easy task to track back to the sources and root-causes of a leakage issue, since it may occur anywhere and almost everywhere in the business operations due to the wide variety of rate plans, products, offers, campaigns, incidents, changes, upgrades and millions or even billions of records in addition to the existence of tiered product plans and flat rates [45]. Moreover, in many cases like in any other business RA analysts may not be informed of such information, although they still have to study their effect and deduce what exactly has affected revenues negatively and positively. Another issue is related to making future decisions based on pervious events; current methods depend on past experience of RA analysts to take future decisions, where human error may occur in this process due to mistakes and staff rotations or replacement. Root-causes of usage traffic and revenue trends anomalies may be due to a change of a certain technical design of a product by an IT officer, a great revenue drop due to a network failure, or a launch of two conflicting campaigns may cause the estimated revenue of each of them to be much higher than the actual revenue. Therefore, such incidents must be recorded for accountability purposes. We argue that automating the debugging and drill down process would greatly increase performance, ease the auditing process, save operators' revenues, provide better analytical experience, better management of data, more accurate reports and leads to an informed future decision making process.

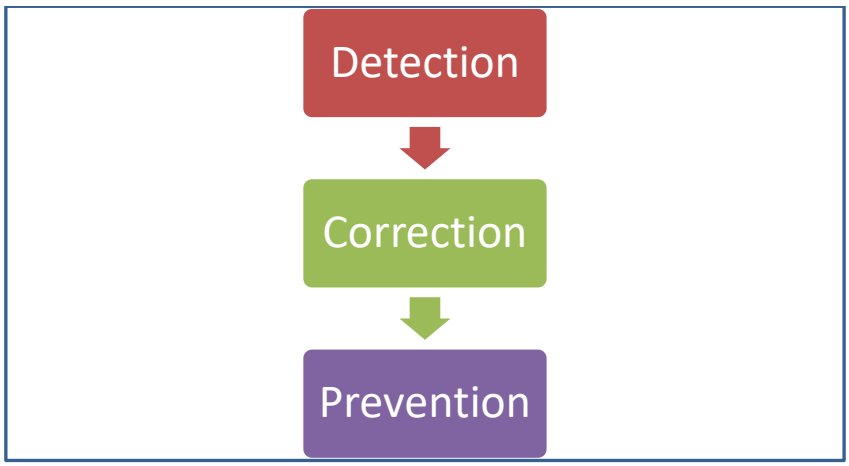


Figure 2 RA Methodology

On the other side, provenance is a global term appeared for the first time in art to relate to the creation and history of a specific artifact, such as the original painter and location of a specific paint and its ownership history, offering a contextual and circumstantial historical track for the artefact [56]. Afterwards, the term has been added to many domains and merged into multiple fields like archaeology, paleontology, archives and science and computing [56]. In the computing field, data provenance is being adopted by defining the origin of processes that have led to a specific state of data product within an information system, such as databases and workflows [40, 44]. Many studies and scientific research have been conducted in computer science for the purpose of data provenance application upon two major domains, science [19, 50] and business [11, 47]. Provenance modelling in the science domain has seen sufficient advancement with the aim of information sharing and validation in the scientific community while preserving copyright and authority aspects. In the business domain the attention was more focused on achieving data quality, reproducibility, auditability, validation, debugging, accountability, error backtracking, prediction, and forward tracking aspects [28, 52, 1].

Several earlier provenance representation and characterization models were proposed in literature, such as [5, 16]. Afterwards, research communities have made

huge efforts and contributions to improve provenance models, resulting in the emergence of the Open Provenance Model (OPM), which is a causal graph standard model representing entities and relationships among them. It describes the whole provenance of data elements including the whole transformation or production process in a workflow, consisting of artifacts, processes and agents [39]. Thereafter, W3C standards body has extended OPM to produce the PROV model to boost entities attribution and evolution over time, where entities may be physical or conceptual [38]. The workflow in OPM and PROV is queried using graph query languages, such as Simple Protocol And RDF Query Language (SPARQL) language to query Resource Description Framework (RDF) graphs [23].

1.2 *Motivation*

In the last two decades telecom industry have witnessed an exponential growth in the data being generated, influenced by information technology and telecommunication sectors advancement and rapid development. The data need to be controlled, quantified, correctly generated, stored, and rated. It has to be carefully analyzed, audited, integrated, and being able to provide knowledge in order to ease and simplify the process of decision making. On the other hand, Revenue leaks largely affect business profitability and continuity, the sources and root-causes behind these leaks need to be identified and repaired in order to recover revenue losses. EYs' Global telecoms revenue assurance survey on 2013 has reported global losses in the telecom industry of 1.1% as yearly revenue leakage, costing the approximately US$15 billion out of US$23 billion for fraud leakage and revenue leakage [18].

From the above, we conclude that there is a critical need in the market for an RA analytics debugging tool that would ease the process of finding root-causes in much

less effort, shorter time, limited need for human knowledge and experience, and limited human intervention by simply providing an alert of the erroneous value associated with possible reasons and sources of the error. Debugging and drill down in current RA architectures are done manually taking several hours or even days of analysis. This is due to the need to understand the complicated nature of data, drill down through hundreds of services, offers, and products to assure that the issue is a real problem or not and to figure out its root-cause and affected subscribers in real time to take correction actions, reduce the amount of human work and effort needed in this process, and make informed decisions in real time.

To our best knowledge, data provenance is not applied as a model in the telecom revenue assurance systems design and development. Therefore, we are motivated in this study to create a new uniquely provenance data model for this domain, since such a model would help revenue analysts audit their operator's traffic in a better way, simplify error tracing back to the source, better management of the data and data quality, and the provision of a historical record of data products and its origins aggregated from multiple sources.

1.3 *Problem statement and research objectives*

Problem statement

Current revenue leakage detection approaches are mainly human-driven and fail to automatically debug and drill down for root-causes of leakage anomalies and issues, requiring time and effort to find the reason(s) of the leakage and increasing the financial effect of the leakage issue. This, necessitates the development of an automated approach to automatically not only detect but also identifies the root-cause of the leakage problem speedily.

Research questions

- ➢ What data elements, business, contextual and semantic, are needed to identify revenue leakage root-cause within a set of defined general scenarios?
- ➢ How to develop a provenance model enriched with contextual and semantic information that captures all data elements needed for leakage root-cause analysis?
- ➢ How to develop a query model that enables automatic debugging for root-cause of revenue leakage detected issues from the underlying provenance model?

The proposed approach improves current RA rule-based architecture to be provenance based. It does this through the deployment of Provenance and Data's (Panda) system approach for provenance capturing and tracking [27]. The proposed provenance model extends Panda's approach by automatically and eagerly capture provenance information once the processes are executed, it stores semantic provenance information in the relationships of the automatically generated data oriented workflows. These workflows are stored in a graph database with the ability to connect to other nodes that may enrich the contextual provenance information.

For revenue leakage detection, alerts may be generated based on Key Performance Indicators (KPIs). Normally, usage and profit trends are expected to follow a specific behavior at each telecom operator and for each usage scenarios. Often, each operator defines its usage KPIs according to its market. For example, if voice calls count is expected to be lower than 100 call per subscriber per day for 98% of Zain Jordan subscribers. Once this KPI threshold is reached to be more than 150 calls per subscriber per day for 20% of Zain Jordan subscribers, indicating that an abnormal

behavior has been introduced, then the proposed model would automatically generate an alert with all possible reasons for the incident and the ability to drill down among raw data.

The proposed provenance model is expected to answer the questions of:

- Identify when, where, and why the issue has been introduced.
- Recognize what actions, events, offers, network failures, system failures, campaigns, incidents may introduce revenue leakages.
- Recognize how data has been changed from its original form.

1.4 *Research methodology*

- Carry out a detailed literature review of recent revenue assurance approaches and techniques to find if there is any approach that enhances the debugging and root-cause analysis capabilities for revenue leakage issues. Also, to find out how data provenance application offers the ability of debugging and drill down on different disciplines and previous studies. In our approach we try to deploy logical data provenance in current rules based RA model.
- Generate a simulated representative dataset, using a quantitative research approach that include a call detail record generation tool. The tool developed by Paul Kinlan [25] was found to generate sufficient call records for our initially defined scenarios. The generated data will be a near replica simulated dataset to a run environment will be utilized by our proposed model to study a set of revenue leakage scenarios in order to detect, analyze the revenue leakage issue, and investigate its root-cause automatically.
- Describe the use cases and scenarios to be carried and tested, which should be representative and descriptive of real cases that may represent fake (i.e false

positives) and real (i.e. true positives) alerts of revenue losses related to voice traffic usage assurance. Then to conduct a systematic analysis of a typical telecommunication environment to identify possible types of leakage scenarios and their respective root-causes.

- Develop a provenance model that includes provenance data elements, contextual and semantic representation of information. Provenance model will be linked to recording needed RA processes and measures integrated into RA data, such as incidents, logs, offers, public holidays, and news tables. It will define how provenance will be represented in the new model by defining the schema that will be used to store the dataset and the architecture used to store provenance information, what data represent provenance information, and the provenance data format.
- Develop a query model that can address the needs of the defined scenarios to run workflow traceability and derive root-causes.
- Use a data oriented workflow processing architecture (e.g. Neo4j graph database [37] and py2neo [50] to query and construct the workflow, and to store provenance information) to be used to represent the source nodes, the associated nodes, processes execution, and data transformation.
- Choose an implementation technology for provenance model data storage, and the query model in order to be able to drill down while back tracing and being able to query the data. Potential candidates include RDF and SPARQL, Relational database and Structured Query Language (SQL).
- Implement the proposed approach, including provenance and query models and carry out experiments to evaluate the richness of the dataset for the defined scenarios and upgrade as necessary.

- Evaluate the performance and correctness of our approach by applying the developed model on different near real cases, then evaluate results by comparing the results to the original designed scenario with leakage problem.

1.5 *Summary of progress and contribution*

- To the best knowledge of the author and based on extensive literature review, there is no research work that enhances current RA systems capabilities to solve the research problem of this study yet there is a critical need in the telecom industry and the research work for such research work.
- So far, a rule-based RA system model that supports backward tracing consisting of thirteen critical revenue assurance functions has been developed with a large dataset stored in a SQL database and connected to a graph database.
- Once, one of the functions is executed, its data workflow is automatically generated, consisting of nodes representing source nodes and SQL processes, and links representing the relationships between the nodes.
- Provenance information is stored in the relationships as mapped attributes and between the nodes and used filters. In addition to additional nodes such incidents, logs, offers, and public events are connected to source nodes and results nodes to enrich the provenance information.
- Once a threshold is reached the system generates an alert with possible reasons and additional information, a data workflow graph explaining how the data has changed and tables of all affected data at each step to give the analyst the ability to drill down among the raw data.

1.6 Organization of the thesis

Chapter 2 provides a background about the topic and discusses the literature and sources available.

Chapter 3 describes revenue leakage types through literature review and systematically.

Chapter 4 gives a general overview about the proposed approach and the scenarios to be used later on.

Chapter 5 gives a general overview of the implementation of the proposed model and testing the scenarios and the evaluation of the proposed model.

Chapter 6 explains the evaluation methodology followed, evaluation metrics, data collection, and discusses the results

Chapter 7 provides a detailed conclusion for the proposed approach contributions, results, limitations and future work.

Chapter 2 Background and literature review

2.1 *Background*

One of the most fascinating aspects about telecom industry is how the business exploits technology and data to extract knowledge, providing better services for their subscribers with appropriate targeting of their market while preserving revenue streams, quality, control, and security aspects [51]. There is a clear correlation between the basic organizational structure (business functions) and the core operational systems (computer systems) in the telecom industry, facilitating the process of identifying knowledge sources within the business and how they are relate to each other. Major information systems applied by telecom operators that may affect the revenue streams significantly in case a problem has occurred. These systems are the data collecting system, the billing system, the settlement system, and the operation and account system [8].

Data in telecom come from different nodes through the Mediation system where they are being parsed, classified and distributed among other systems as in Figure 3. The figure represents the nodes in the Global System for Mobile communications (GSM) and the Universal Mobile Telecommunications System (UMTS) networks. RA detection mechanism works by monitoring data movement and changes between different systems and within the same system using comparison, investigation and auditing of information and processes to find the root-cause behind detected errors [54]. To be more specific, telecom data vary from subscriber information, call details, Short Message Service (SMS) details, General Packet Radio Service (GPRS) usage details, subscription and refill transactions, and charging information.

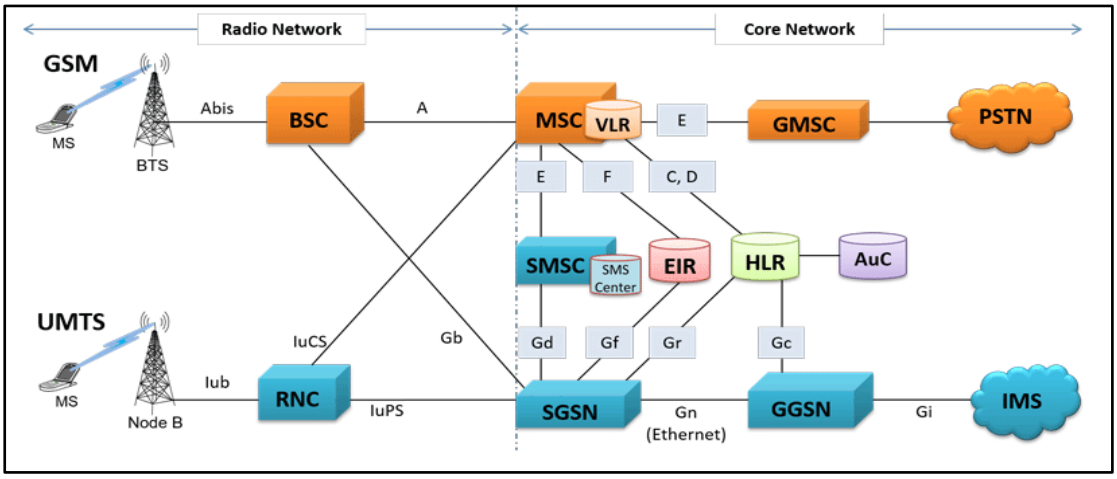


Figure 3 GSM and UMTS networks architecture [20]

Some of the main network nodes and systems are:

- MSC: Mobile Switching Center which is the network server that is responsible for call setup, routing, and release in addition to SMS routing, conference calls, and the interaction with other networks, such as Public Switch Telephone Network (PSTN). MSC records contains information, such as calling party, called party, traffic type, duration, switch ID, date and time, Roaming or local [54].
- SMSC: Short Message Service Center is a mobile telephone network node that is responsible for storing, forwarding, converting, and delivering short messages from mobile to mobile, mobile content provider, and application to mobile. SMSC records contains information, such as calling party, called party, traffic type, switch ID, date and time, Roaming or local [58].
- CCN: Charging Control Node, which is a signaling node that is responsible for the rating and charging of data packets and usage events in the communication network and the generation of call detail records for prepaid subscribers. CCN records contains information, such as calling party, called party, traffic type, duration, volume, date and time, roaming or local, charge [46].

- Billing: Billing system is responsible for loading all charged events of postpaid subscribers into the billing table. Billing table contains information, such as calling party, called party, traffic type, duration, volume, date and time, roaming or local, charge [59].
- SGSN: Serving GPRS Support Node is the node responsible for handling GPRS traffic within the communication network, SGSN is like MSC node for voice traffic. SGSN record contains information, such as calling party, used Access Point Name (APN), duration, downlink volume, uplink volume, total volume, date and time, local or roaming [53].

While there is a critical need in telecom industry for revenue leakage detection, debugging and drill down in real time. Data provenance has been widely used for auditing, root-cause analysis, and debugging purposes exploiting its tracing and information capturing capabilities to get the derivation history of data in database systems and workflow systems and determining the source of error.

Many studies and scientific research have been conducted on the application of data provenance following two main research domains, data provenance in database systems and data provenance in processes. Data provenance in database systems has been mainly focused on the executed query results [9]. Data provenance is captured to determine from where these results have been acquired, why these data were included in the results and why other data were excluded from the results [7]. The other research domain, data provenance in processes which are represented in workflow systems, where provenance is captured in the workflow after each step execution completion [12].

Several earlier provenance representation and characterization models were proposed in literature, but the first main widely used representation model was the Open

Provenance Model (OPM), a causal graph model representing entities and relationships among their transformation or production process in a workflow, consisting of artifacts, processes and agents [39]. Afterwards, the World Wide Web Consortium (W3C) standards body has produced the (PROV) model influenced by (OPM) model, PROV was produced to boost entities attribution and evolution over time, where entities may be physical or conceptual [38]. (PROV) model is represented by a directed graph consisting of entities, activities, and agents connected together to define the provenance. In this study, we take the transformation of revenue leakage detection functions and processes, as our domain, in determining provenance, and PROV model as our provenance representation standard.

In this work, we use Neo4j graph database [37] to build the data-oriented workflow, representing the data nodes included in the revenue leakage detection functions and processes, associated nodes for contextual information provision and executed SQL processing nodes. Where provenance semantic information are also being represented in the processing nodes. Neo4j is an open source non-relational, graph database developed in Java that uses graph structures for data storage purposes. Neo4j is considered one of the most popular and widely used graph databases due to its capabilities of full and incremental backups, Cypher language, high availability, Atomicity, Consistency, Isolation, Durability (ACID) based integrity, and its comprehensive graphical user interface [37]. In order to work with Neo4j using Python language, Py2neo Application Programming Interface (API) is used [50].

2.2 *Literature review*

Revenue assurance systems use the same concept of understanding the revenue management chain to detect and prevent revenue leakage cases, but they differ in the

techniques they use. These systems collect data from different network data sources, these data are being reconciled and matched based on automated rules or patterns, further investigation and reporting are being conducted based on the reconciliation results. KPIs and alerts are also used for notification purposes in the automated management.

Through literature review, we can observe the very limited research efforts performed on the area of revenue assurance understanding and enhancement in telecom domain, Rob Mattison is an internationally recognized RA expert and the president of GRAPA (The Global Revenue Assurance Professional Association), Mattison has a notable contribution in defining revenue assurance concept and standards in the telecom industry [36, 33]. He has published several white papers for assessing revenue assurance capabilities [34]. Mattison has also proposed a framework for revenue assurance decision making based on the cost and benefit equation, and elaborated on how to evaluate both of them and maximize benefits and minimize the cost [35]. But to our knowledge there is no research work that enhances current RA systems capabilities to solve the research problem of this research work.

Revenue leakage is likely to occur in any business regardless of its sector or size, and at any level in the revenue cycle due to errors happen at the organizational process or the technological side of the business. These errors usually take place as the complexity of the business itself increases, and when it comes to the field of telecom industry, these errors have a higher percentage to occur because of the high degree of complexity in the domain products and services, emerging technologies, introduction of new systems in the market, and integration among all existing systems. Therefore, the main possible sources of revenue leakage in telecom industry are fraudulent

activities, inappropriate processes, poor system integration, undiscovered discrepancies, human error, and inappropriate marketing.

Although several approaches appear in the literature that provide different methods for revenue leakage detection and root-cause analysis in other disciplines, such as the health and airline domains [27, 29, 31, 36, 42], however for the telecom domain, relatively very limited number of relevant approaches can be found in the literature [21, 15, 48, 60]. In this context we'll review some research works that have addressed this problem and the provided techniques for revenue leakage detection, their most important positive and negative aspects. Generally speaking revenue leakage detection can be classified into: analytical-based approaches and rule-based approaches. Provenance-based approaches are also discussed to bring relevance to the topic.

2.2.1 Rule-based approaches

Niall et al. [21] have proposed a system and method for performing offline revenue assurance activities on data usage in order to detect revenue leakage areas. This is done by auditing and investigating usage records from their sources in the network to identify the resulted discrepancies then take the suitable corrective actions. The system works by adding a revenue assurance component to the telecom network. This component is configured to receive consumption data from an online real time charging system, and receive usage data from the mediation system. Then the RA component compares consumption data to usage data for each service to identify inconsistencies, in addition to performing accuracy and completeness checks on the revenue streams by doing trend analysis, measuring trends, and dividing billing data into revenue streams and rising alerts based on certain predefined Key Performance Indicators KPIs. Discrepancies are determined through operations, such as time gap analysis, statistical

count analysis, and audit trail tracking that involves linking an identifier between usage records and a summary record. The inconsistencies are corrected by asking other network components to resend the missing data. If the discrepancies are not resolved, then the situation needs to be investigated by the RA admin manually to analyze the root-cause of discrepancies.

Fine et al. [15] have presented a patented revenue assurance tool for multimedia services. The proposed system works by collecting data from different network elements into a data repository configured tables in addition to vendor settlement data [15]. The system also includes a processor responsible for reconciliation, tracking, and reporting based on certain predefined rules, the reconciliation is done by linking at least one table to at least another one table to compare their data, then an alert rule is triggered in case the threshold that is associated to this rule has been exceeded. Moreover, tracking and trending of the data retrieved from at least one of the tables is done, but when the associated rule threshold of a certain type of traffic is exceeded an alert is raised. The alerts triggered by reconciliations and trending may indicate a possible real revenue discrepancy related to a revenue leakage problem, data anomaly, system defect, or an input error which are usually followed by a manual investigation phase and correction and recovery actions.

2.2.2 Analytical-based approaches

In the healthcare domain, according to a white paper by Opera solutions 54% of hospital executives have increased their employees in order to solve the problem of revenue leakage. This is done by addressing revenue integrity problems through manual revision of the revenue stream [24]. But due to the complex nature of the systems, data, policies, and billing codes in addition to the great amount of data, other executives have

implemented another solution, rule-based systems, which raise an alert for potentially missing bills, charges, diagnose, or procedures. The paper argues that both solutions have many limitations because of raising fake alerts and wasting time and money, since the manual revision requires highly skilled auditors, costly, and its effectiveness relies to a high extent on the labors and their professionalism. While rule-based systems may be too aggressive or conservative by detecting too many false positive for review or failing to detect real missing problems, and they need to much time to be adopted, maintained and reach the maturity level. Moreover, these systems need to be updated each time a change made to billing. Therefore, the authors propose a more efficient, effective, accurate, flexible, and comprehensive solution. This solution uses the analytic approach exploiting the concepts of machine learning, predictive modelling, and anomaly detection to detect outlier behavior and score the patients invoices according to incorrect charges and missing bills. Then these scores are ranked depending total impact in money. Highly prioritized missing invoices or charges are then investigated by nurse auditors to confirm whether they are actually missing to decide where the auditing resources should be focused. After investigation, nurse auditors find the root-cause for revenue leakage, start the correction actions, and add missing amounts to the patients' invoices. The proposed solution is effective and efficient in rising accurate and correct alerts for missing invoices, charges, and procedures. It also has the ability to continuously learn and adapt to find the source of leakage, but the main shortcoming for this solution that it has to be followed by manual effort and intensive investigation to find the root-cause for the missing problem, the solution finds out the source of missing but the auditors then have to analyze why this missing exists.

Another paper, by Schouten [48], explains how the ineffective revenue cycle management in healthcare industry has have a major impact on the bottom line for long

time ago due to the billing complexity. Also the lack of the right technology to detect and recover from missing and erroneous billed amounts has affected revenues. It discusses the shortcomings of other revenue leakage detection approaches; manual auditing and rule-based systems as in the previous study. Then the author has introduced the advanced analytics approach combined with machine learning, predictive modelling, anomaly detection, and pattern recognition to find hidden connections to filter out missing records and erroneous charges as in the previous study. Again this is a powerful approach but it needs too much of human effort and time to analyze the root-cause of the problem and take corrective actions.

While Zoldi et al. [60] have followed another approach by explaining that many revenue leakage problems occur due to network problems. These problems may cause calls not to be charged depending on the severity of the problem. Therefore, the authors argue that a network assurance analytics system that detect anomalies related to network issues that may lead to a revenue leakage, provides the telecom operators with the ability to apply fast fixes to the revenue leakage area and prevent great financial losses. The system works by using recursive variables in order to detect abnormal patterns, which may be a sign of a network failure or issue associated to a risk score. All scores are ranked according to their severity combined with reason codes captured by the recursive variables. Moreover, the system uses profiles for predictive analytics; these profiles are concise mathematical representations containing recursive variables of calling patterns occurring in the networks to study the calling behavior at four levels: the calling party (originating number), the network signaling node, the call route from the calling party to the called party, and the called party (terminated number).

From the above, we conclude that observed revenue leakage detection approaches lack the ability to automatically find out the source of error or leakage. It

also requires human knowledge and intervention in the investigation process. Manual auditing is not an option in this context for leakage detection. Moreover, big data analytics approach works on pattern basis and requires time to adapt and learn based on previously known issues and anomalies. Whereas rule-based approach in our opinion performs better in leakage detection, since they work based on certain predefined rules and must be highly efficient if it is well tuned. Therefore, RA rules based approach will be the main focus in this study to be extended by adding the ability of root-cause debagging through data provenance after leakage detection is completed.

2.2.3 Provenance-based approaches

Data provenance has been used for auditing, root-cause analysis, and debugging purposes, since it refers to the origins and the derivation history of data, making it easier to trace back to the sources and root-causes of faults and errors in database systems and workflow systems [27, 29, 31, 36, and 42]. Using data provenance has shown to be very helpful for debugging goals; it defines the relationships between inputs and outputs in addition to the processes involved in the flow.

According to Ikeda et al. [27] processing steps performed by the users on publically available data for analysis purposes can be managed using data oriented workflows. But the authors argue that these systems lacks two functionalities that may enhance the systems greatly, which are debugging and drill down features. Therefore, they have proposed the Panda system (Panda stands for Provenance and Data) [27], which employs logical data provenance concept to support both debugging and drill down by tracking input data, processing steps, transformations, and finally the output data. Logical

provenance supports tracking at the schema and processing node level, in addition to the provenance of the information stored at the processing node itself. Panda is used to integrate, process, and analyze data oriented workflows with multiple data sources, providing four functionalities: data oriented workflows creation and execution for relational processing nodes and python processing nodes, logical provenance generation automatically for relational processing nodes, and manually for python processing nodes through the use of attribute mappings and filters, provenance operations including backward tracing, forward tracing and refresh operations. The final fourth functionality is enabling the users to do the previous operations associated with viewing workflow graphs, and perform tracing using a comprehensive graphical user interface. Debugging and drill down features require performing provenance tracing first using Panda's Graphical User Interface (GUI) by specifying a tracing path. After running the workflow, if an abnormal behavior is detected, data provenance is investigated using backward tracing till the source of error is reached. While the drill down functionality is achieved via backward tracing to the source of error in addition to tracking down erroneous values through the browsing and exploration of the source.

Husted et al. [36] proposed a mobile devices system for debugging and profiling of mobile operating systems using data provenance. The system consists of the Android operating system, a provenance middleware, and Linux audit subsystem port. It provides the ability to detect performance problems and trace back to find out their root-causes. The authors have investigated two use cases that affect the battery life and low performance of mobile device. The first use case is related to wake locks and lags. Debugging is done by provenance

generation of workflow graphs, then tracing back on the provenance graph each use case to identify places that may have affected the problem to occur. The performance results of the system were promising, but it needs to be improved and provenance records obtained need to be increased.

Furthermore, many approaches were proposed in literature for debugging Data-Intensive Scalable Computing (DISC) systems using provenance, such as RAMP [42], that supports workflows tracing on MapReduce framework [13], which are targeted by Hadoops' higher level platforms [22]. The debugging process using provenance is done by backward tracing suspicious outputs back to their sources in the workflows, and the path is defined through the subsets that have contributed to the erroneous output element. The researchers have implemented the RAMP on an extension on Hadoop [22] to capture and trace fine-grained provenance on MapReduce [13] workflows using a wrapper based approach providing the ability to efficiently drill down through output results. The system consists of generic wrapper for provenance capturing based on MapReduce jobs, pluggable schemas for the storage of provenance as mappings between inputs and outputs with the assignment of IDs on output elements, and a stand-alone program for provenance tracing purposes. RAMP was tested on a real use case for movie sentiment analysis on MapReduce Pig script for movie ratings inference over Twitter data, then the researchers has conducted two drill down scenarios to find out using backward tracing until the original tweets why people didn't like two movies rated on Twitter, and one debugging scenario to figure out why people liked another movie.

Another Data Intensive Scalable Computing (DISC) debugging system called Newt [31] was presented as a scalable debugging system at record level for big data analytics working based on why provenance as RAMP, but it adds a new feature over RAMP since it can capture and trace data over multiple processes and granularities, and the capturing is timed based on data arrivals and departures in order to lower the amount of overhead. Moreover, Newt captures provenance from pipelined actor execution instead of annotations [31].

Interlandi et al. [29] argue that the previous DISC debugging approaches (RAMP and NEWT) have some limitations related to the use of external storage for provenance information, provenance queries are supported by a separate programming interface, and they lack the ability of viewing intermediate data or to replay the processing steps on the data. Therefore, they have proposed Titian framework that supports capture and query features in Apache Spark DISC system, minimizes overhead by improving the capturing design, scalable on large datasets with less overhead, and supports interactive provenance query.

From above mentioned studies we can conclude that current architecture of RA approaches lacks the ability to identify the root-causes of revenue leakages after leakage detection and alerting is completed, whereas data provenance has been widely adopted for root-cause analysis and debugging. Therefore this work is proposed to combine and extend the two approaches to develop a provenance-based debugging and drill-down approach for revenue leakage detection in telecom systems.

Chapter 3 Revenue leakage types

This chapter describes potential and possible revenue leakage types. It goes through a systematic analysis of a typical telecommunication environment under study. It aims to identify possible revenue leakage types on which our proposed approach could be evaluated and tested for.

3.1 *Network and system errors*

Issues related to the telecom network may occur due to many reasons, such as nodes signaling problems affecting service provision or traffic generation impacting mediation, rating, and billing systems. Other possible causes could be a failure in the Call Detail Records (CDRs) transportation or systems integration, such as MSC to Mediation CDRs transfer. For instance, if the CDRs were not sent to the mediation system, then the CDRs will not be charged resulting in great financial losses.

Revenue leakage occurring due to a failure at the network level or system level may cause complete or partial disconnection, resulting in uncharged transactions, mischarged transactions, charged but failed transactions, or fake revenue leakage alerts. Therefore, to check revenue leakage issues for this type, we have to consider all related network nodes and systems and identify possible leakage issues for each of them for postpaid and prepaid telecom usage traffic as below:

1. MSC node.
2. SMSC node.
3. Service Aware Support Node (SASN).
4. CCN node.
5. Mediation system.

6. Billing system.

We need to systematically trace revenue leakage sources and reasons due to a network or system errors, in order to map the needed checks per leakage issue reason and network node or system. In Figure 4, we have classified the network usage according to its type, then each type was connected with its originating source node, then it has been classified according to the subscriber profile to identify the used system or node for charging, CCN for prepaid subscribers and Billing for postpaid subscribers.

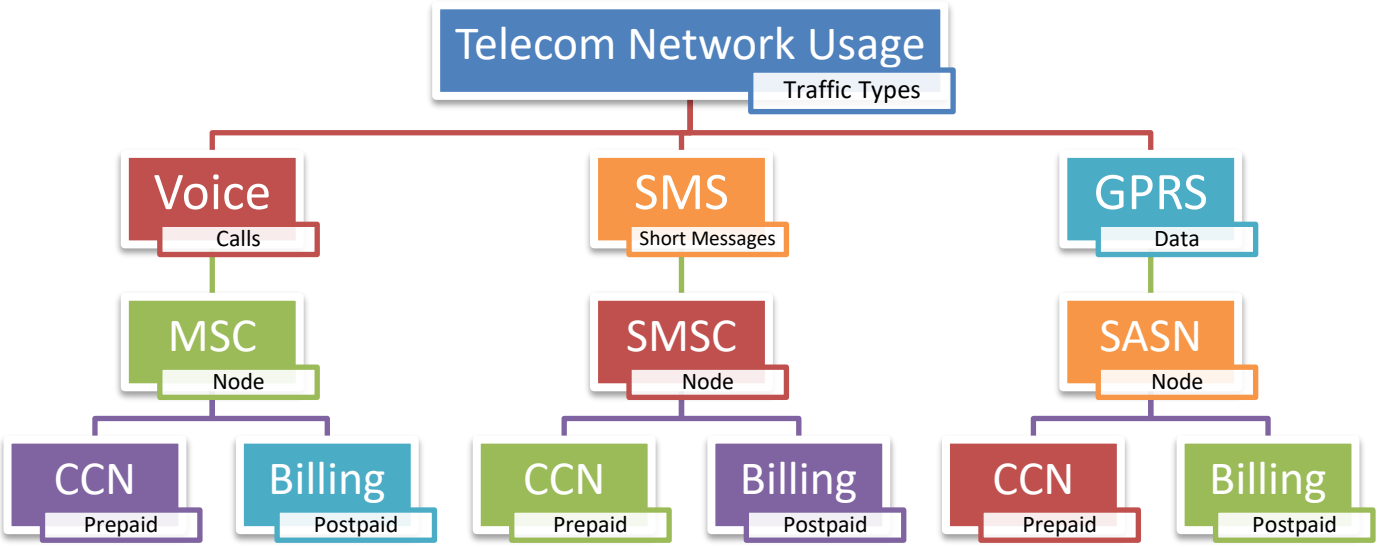


Figure 4 Telecom Network Usage

Figure 5 presents possible issues related to voice calls traffic with their related nodes and systems. For prepaid traffic, calls are originated using MSC, then they pass through the CCN to get charged, if for any reason these calls were not generated from any of these nodes or generated with wrong values due to a partial disconnection we will have three possible leakage scenarios:

1. Missing CDRs from CCN side causing calls not to be charged, in this case MSC calls count would be greater than CCN calls count, in this work we are interested to study the missing CDRs from CCN side symptom due to:
    - MSC disconnection caused by a switch upgrade.
    - CCN disconnection caused by a power failure in Service Deliver Platform (SDP) site.
2. Missing CDRs from MSC side causing calls to be charged without being provisioned to the subscribers (the subscribers in this regard do not get the service they have already paid for), in this case CCN calls count would be greater than MSC calls count, in this work we are interested to study the missing CDRs from MSC side symptom due to error in CDRs generation in MSC node caused by MSC node configurations change.
3. Mismatched calls duration between MSC and CCN due to a partial disconnection between the two nodes, causing calls to be undercharged, in this work we are interested to study mismatched calls duration between MSC and CCN due to CCN disconnection caused by power failure in SDP site.

Regarding postpaid voice calls traffic, the calls are originated by MSC, then calls are passed by the mediation system to the billing team in order to be loaded into the billing table. Therefore, in this context we may have two possible leakage scenarios to consider:

1. Missing records from Billing side due to a problem in the mediation system or any loading related problem which causes the calls not to be charged. In this work we are interested to study the missing CDRs from Billing side symptom due to a problem in the mediation system caused by an upgrade made to the mediation system.

2. Missing calls from both sides, MSC and Billing, meaning that if for any reason MSC node did not generate call records, these calls will not be billed as MSC is the only source for the Billing system. But we can identify this issue in two ways, the first by noticing a drop in the whole traffic in the usage monitoring trend. The second is to investigate prepaid voice calls traffic as MSC will not generate calls records for prepaid subscribers as well, but CCN will generate the records and we will get missing calls from the MSC side while they exist on CCN side. In this work we are interested to study the missing CDRs from MSC and Billing symptom due to error in CDRs generation in MSC node caused by MSC node configurations change.

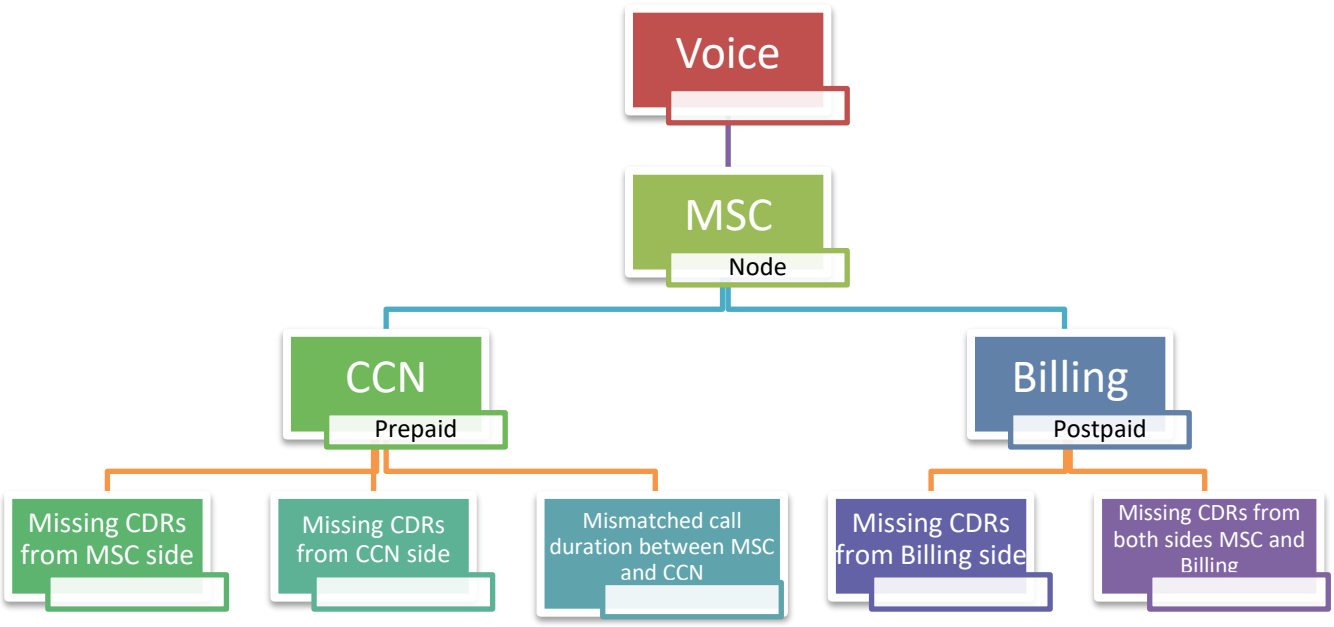


Figure 5 Voice calls traffic description

Figure 6 presents possible issues related to GPRS sessions traffic with their related nodes and systems. For prepaid traffic, GPRS sessions are originated using SASN, then they pass through the CCN to get charged. If for any reason these sessions were not generated from any of these nodes or generated with wrong values due to a partial disconnection we will have 3 possible leakage scenarios:

1. Missing CDRs from CCN side causing GPRS sessions not to be charged, in this case SASN GPRS sessions count would be greater than CCN GPRS sessions count, in this work we are interested to study the missing CDRs from CCN side symptom due to CCN disconnection caused by power failure at SDP site.
2. Missing CDRs from SASN side causing GPRS sessions to be charged without being provisioned to the subscribers (the subscribers in this regard do not get the service they have already paid for), in this case CCN GPRS sessions count would be greater than SASN GPRS sessions count, in this work we are interested to study the missing CDRs from SASN side symptom due to error in CDRs generation at SASN node caused by SASN node configurations change.
3. Mismatched GPRS sessions volume between SASN and CCN due to a partial disconnection between the two nodes, causing GPRS sessions to be undercharged, in this work we are interested to study mismatched GPRS sessions volume between SASN and CCN due to CCN disconnection caused by power failure in SDP site.

Regarding postpaid GPRS sessions traffic, the sessions are originated by SASN, then they are passed by the mediation system to the billing team in order to be loaded into the billing table. Therefore, in this context we may have two possible leakage scenarios to consider:

1. Missing records from Billing side due to a problem in the mediation system or any loading related problem which causes the GPRS sessions not to be charged, in this work we are interested to study the missing CDRs from Billing side symptom due to problem in the mediation system caused by an upgrade made to the mediation system.

2. Missing GPRS sessions from both sides SASN and Billing, meaning that if for any reason SASN node did not generate GPRS sessions records, these sessions won't be billed as SASN is the only source for the Billing system. But we can identify this issue in two ways, the first by noticing a drop in the whole traffic in the usage monitoring trend. The second is to investigate prepaid GPRS sessions traffic as SASN won't generate calls records for prepaid subscribers as well, but CCN will generate the records and we'll get missing GPRS sessions from SASN side while they exist on CCN side. In this work we are interested to study the missing CDRs from SASN and Billing symptom due to error in CDRs generation in SASN node caused by SASN node configurations change.

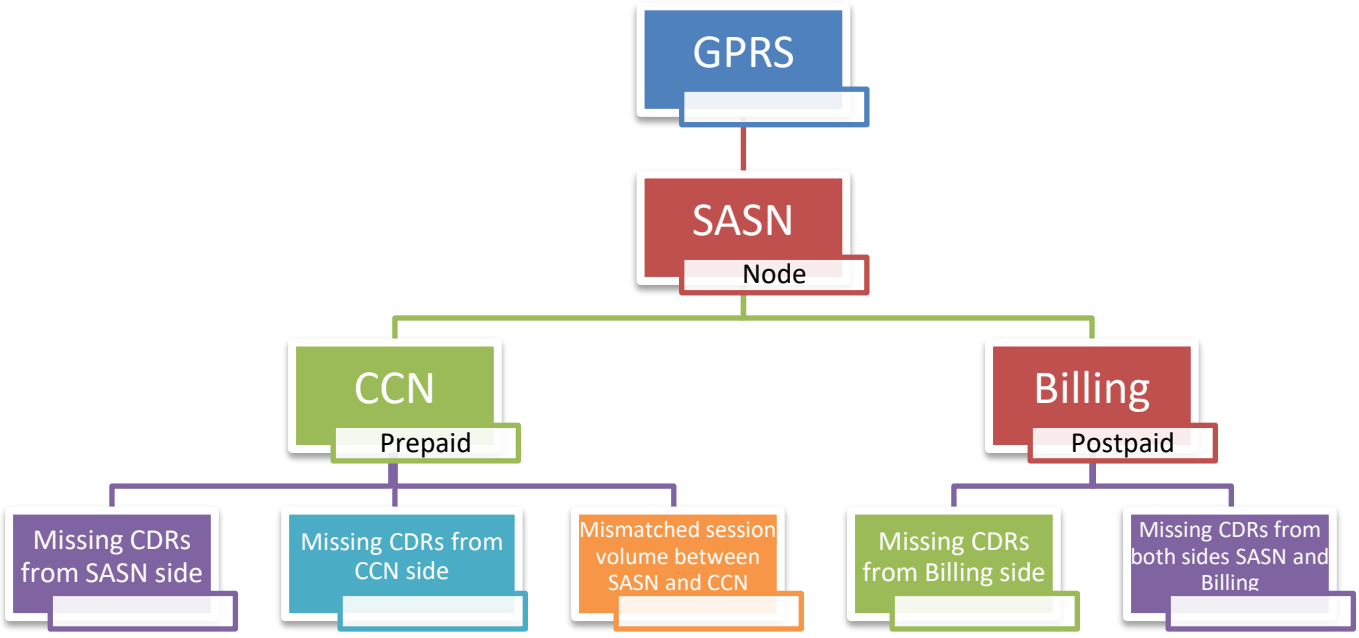


Figure 6 GPRS sessions traffic description

Figure 7 presents possible issues related to short messages traffic with their related nodes and systems. For prepaid traffic, SMSs are originated using SMSC, then they pass through the CCN to get charged, if for any reason these SMSs were not generated from any of these nodes we will have 2 possible leakage scenarios:

1. Missing CDRs from CCN side causing SMSs not to be charged, in this case SMSC SMS count would be greater than CCN SMS count, in this work we are interested to study the missing CDRs from CCN side symptom due to CCN disconnection caused by power failure at SDP site.
2. Missing CDRs from SMSC side causing SMSs to be charged without being provisioned to the subscribers (the subscribers in this regard do not get the service they have already paid for), in this case CCN SMS count would be greater than SMSC SMS count, in this work we are interested to study the missing CDRs from SASN side symptom due to error in CDRs generation at SASN node caused by SASN node configurations change.

Regarding postpaid short messages traffic, the SMSs are originated by SMSC, then SMS traffic is passed by the mediation system to the billing team in order to be loaded into the billing table. Therefore, in this context we may have two possible leakage scenarios to consider:

1. Missing SMS records from Billing side due to a problem in the mediation system or any loading related problem which causes the SMSs not to be charged, in this work we are interested to study the missing CDRs from Billing side symptom due to problem in the mediation system caused by an upgrade made to the mediation system.
2. Missing SMSs from both sides SMSC and Billing, meaning that if for any reason SMSC node did not generate SMS records, these SMSs won't be billed as SMSC is the only source for the Billing system. But we can identify this issue in two ways, the first by noticing a drop in the whole traffic in the usage monitoring trend. The second is to investigate prepaid voice SMS traffic as SMSC won't generate SMS records for prepaid subscribers as well, but CCN

will generate the records and we'll get missing SMSs from SMSC side while they exist on CCN side. In this work we are interested to study the missing CDRs from SMSC and Billing symptom due to error in CDRs generation in SMSC node caused by SMSC node configurations change.

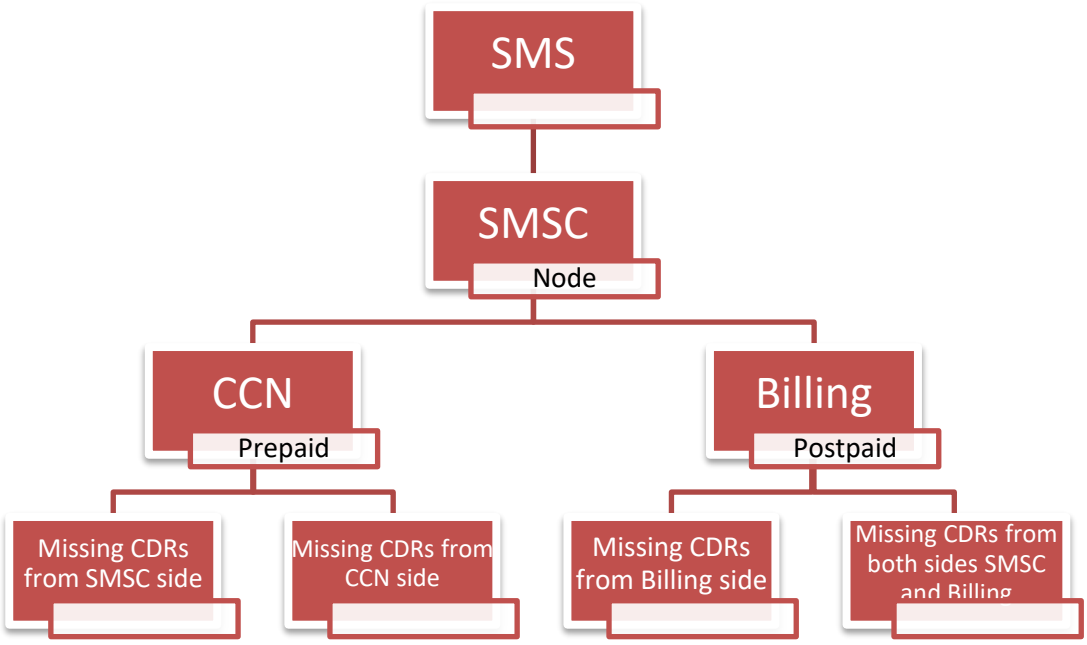


Figure 7 Short messages traffic description

Therefore, we need to test our approach on the following scenarios:

1. Missing CDRs from CCN side problem (MSC vs. CCN) for voice calls traffic: could be due to an MSC error or CCN error.
2. Missing CDRs from MSC side problem (MSC vs. CCN) for voice calls traffic: due an MSC error.
3. Mismatched calls duration problem (MSC vs. CCN) for voice calls traffic: could be due to an MSC error or CCN error.
4. Missing CDRs from CCN side problem (SASN vs. CCN) for GPRS traffic: could be due to a SASN error or CCN error.

5. Missing CDRs from SASN side problem (SASN vs. CCN) for GPRS traffic: due to a SASN error.
6. Mismatched GPRS session's volume problem (SASN vs. CCN) for GPRS traffic: could be due to an MSC error or CCN error.
7. Missing CDRs from CCN side problem (SMSC vs. CCN) for short messages traffic: could be due to an SMSC error or CCN error.
8. Missing CDRs from SMSC side problem (SMSC vs. CCN) for short messages traffic: due to an SMSC error.
9. Missing CDRs from Billing side problem (MSC vs. Billing) for voice calls traffic: could be due to a billing error or a mediation error.
10. Missing CDRs from both sides problem (MSC and Billing) for voice calls traffic: due to an MSC error.
11. Missing CDRs from Billing side problem (SASN vs. Billing) for GPRS traffic: could be due to a billing error or a mediation error.
12. Missing CDRs from both sides problem (SASN and Billing) for GPRS traffic: due to a SASN error.
13. Missing CDRs from Billing side problem (SMSC vs. Billing) for short messages traffic: could be due to a billing error or a mediation error.
14. Missing CDRs from both sides problem (SMSC and Billing) for short messages traffic: due to an SMSC error.

3.2 *Human errors (miss configuration)*

Human errors occurs when wrong configurations are set or incorrect tariffs are entered into the rating engine and billing system, illustrated in Figure 8. Such errors occur accidently without any intention to cause harm. Examples of such issues is to set a network node configurations to the default without making suitable adjustment, changing voice calls minute price to the wrong values, or entering wrong service tariffs to the rating engine. Human errors at network and system configurations level lead us to the previous type of revenue leakage issues (network and system errors) causing disconnection or improper systems integration.

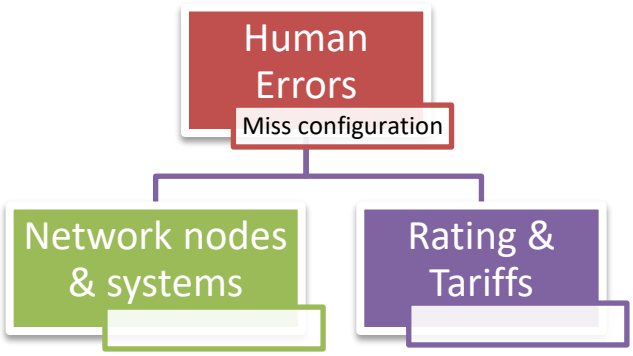


Figure 8 Human Errors

Therefore, we need to test our approach on scenarios related to the rating and tariffs errors as below, since network errors will be tested as in the previous section scenarios. Therefore, we will test the scenario of calls minute rate change due to prices change by an employee mistakenly.

3.3 *Poor product or service design*

Multiple products, services, and offers with various technical designs are being launched on monthly basis due to the market diversity and high competition. Any single human or systematic error in these designs may lead to significant financial losses. Poor designs may cause quality issues and raise fake revenue leakage alerts, or may affect

other products by launching two conflicting products for the same subscribers, since each product would have calculated budget and estimated revenues, illustrated in Figure 9.

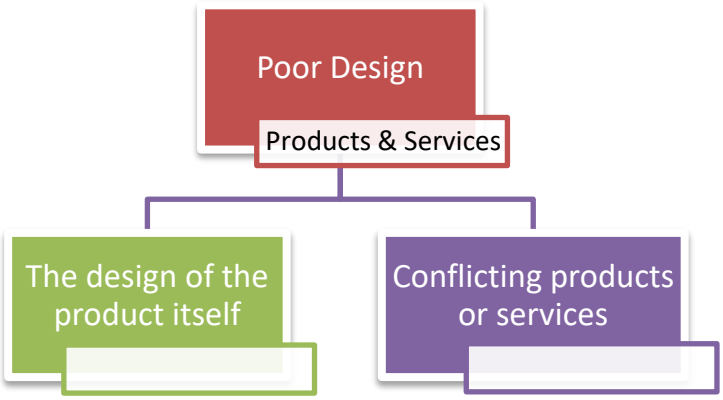


Figure 9 Poor product or service design

Therefore, we need to test our approach on the following scenarios:

1. Product or service poor design due to rounding criteria change.
2. Conflicting products design due to the launch of a conflicting offer.

3.4 *Internal and external fraud*

Although fraud detection is not an RA responsibility as there is a fully dedicated team for telecom fraud analysis and detection. But sometimes revenue assurance checks may indicate some internal or external fraudulent behavior as a revenue leakage alert affecting telecom usage and revenue streams. An example is illustrated in Figure 10. An internal fraud indicates that an employee with bad intention who has access to critical information and has the privileges to make changes, may intentionally change business rules and service tariffs causing great losses. On the hand, an external fraud may exploit a vulnerability in services and products technical implementation to use these services and products for free and provide for others too, consuming network resources and affecting its reputation and revenues.

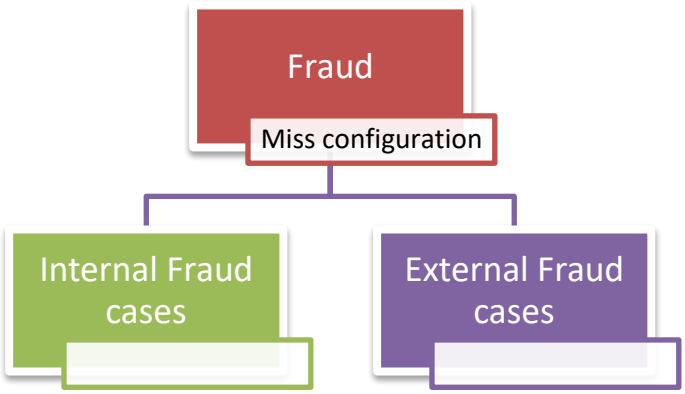


Figure 10 Fraud Types affecting revenue streams

Therefore, we need to test our approach on the two scenarios below:

1. Internal fraud scenario due to a price change by an employee with bad intent.
2. External fraud scenario due to using application to change GPRS traffic APN to appear as initiation traffic.

3.5 *Regulatory and non-compliance*

Related to the commitment to the international and statuary regulations to assure the accuracy of financial statements, this type of checks will not be considered in this work as it is not a systematic auditing process [3].

3.6 *Alert Leakage parameters and root causes*

For all above types we need some parameters to identify that a leakage may have occurred and generate an alert using rule-based controls consisting of audits and measures. Audits connect two relevant sources of data representing nodes and systems, do a comparison check between the data originated based on certain rules by the two sources. While measures extract data from data sources based on certain rules to

identify any abnormal behavior. Parameters needed to detect revenue leakages in both controls are:

- Audits: Calling Mobile Station International Subscriber Directory Number (MSISDN), called MSISDN (for calls and SMSs), APN (for GPRs), date, time of transaction, duration, and data volume.
- Measures: Date, total records count, total calls duration, total GPRS sessions volume, total charges.

Information needed to add a meaning to the detected leakage issue for root-cause identification are:

- Public events or holidays related to the date of the alert, since these events could greatly effect usage trends and revenue streams. Useful to identify root-causes of fake alerts related to public events or holidays that affect the monitoring trends and public news about the use fraudulent applications.
- Logs: logs occurred at the date of the alert and related to the type of traffic that was affected by the leakage issue. Useful to identify root-causes of human errors (miss configuration) and system errors resulted from systems and rates changes.
- Offers: offers launched at the date of the alert and related to the type of traffic that was affected by the leakage issue. Useful to identify root-causes of bad product or service design, such as conflicting campaigns
- Incidents: incidents occurred at the date of the alert and related to the type of traffic that was affected by the leakage issue. Useful to identify root-causes of network errors resulted from network upgrades and changes.

- Semantic information represented by mapped attributes and used filters are also required and useful to trace back through provenance diagram till identifying the root cause of the problem.

For instance, a great revenue drop may be due to an error at the network or systems level that may have caused a CCN disconnection to happen. It may also be due to a human error at the rating engine, a human error at network configurations level, an internal fraud error by changing prices to free, an offer was launched, or an external fraud that changes traffic to appear as free traffic to the charging system. Semantic and contextual information add a great help in this regard, if a network incident has occurred, an offer was launched to a certain type of traffic, system logs of changes, and public events would be automatically mapped to the leakage alert and all possible root-causes would be presented to the analyst.

Chapter 4 Proposed approach

4.1 *Introduction*

This chapter explains our proposed provenance based model and how it would be integrated into the current rule-based RA architecture to achieve debugging and drill down features. Provenance data represents semantic and contextual information related to the leakage issue or anomaly, these data are being created automatically based on the processes and sub-processes flow on the RA system. Each RA detective processes consists of a number of processing steps and at least one entity.

Provenance diagrams are used to graphically represent the data lineage by showing all the processing steps that were held during the execution of the function, the source nodes that were exploited as a source for data extraction, in addition to the entities that may provide contextual information related to the leakage issue. Provenance diagrams are designed to be backward looking, so arrows are originated from the final data item and targeted towards the process that has created it and keep going back till the first entity that was used for data extraction. In the proposed model entities that represent source nodes in the telecom network are associated with other entities to provide contextual information related to the RA function studied based on certain parameters and the final data item in the graphical workflow is associated with an entity named public holidays and events based on the date parameter to add more contextual information related to a specific date that may have affected the usage traffic or caused the trends anomalies.

Any process or entity in the provenance diagram is given a unique identifier and name and they are connected using relationships. The relationships are given the properties: the source node, the end node, and the relationship type. Semantic

information represented by mapped attributes, and used filters if any are stored at the processing node level.

4.1.1 Provenance models

Significant standard models have been developed with the aim of explaining the derivation history and contextual information of things in provenance graphs, the most important and influential models are the Open Provenance Model (OPM) and PROV Model [41, 38, and 4].

4.1.1.1 Open Provenance Model (OPM)

The Open Provenance Model is a model for provenance capturing and representation into annotated causality graphs. OPM was developed due to the critical need in the scientific domain for reproducibility of scientific work and analysis assurance. OPM specifications were set in 2008 for the first time, and many challenges were held to solve provenance special issues. The main requirements OPM was designed to meet are related to exchanging provenance information between different systems, enable developers build and share tools that have the ability to run on OPM, give an accurate specific technological definition for provenance, support the digital representation of provenance, allow the coexistence of multilevel provenance descriptions, and set the rules for valid provenance representation inferences identification [41].

OPM represents things whether they are physical, digital, real or imaginary things in directed graphs by determining nodes, dependencies, and roles. The node may be an artifact, a process, or an agent. While the dependencies represent the relationships

between the nodes, with 5 types of dependencies as in Figure 11, and roles represent the agents' or artifacts' responsibility for a process.

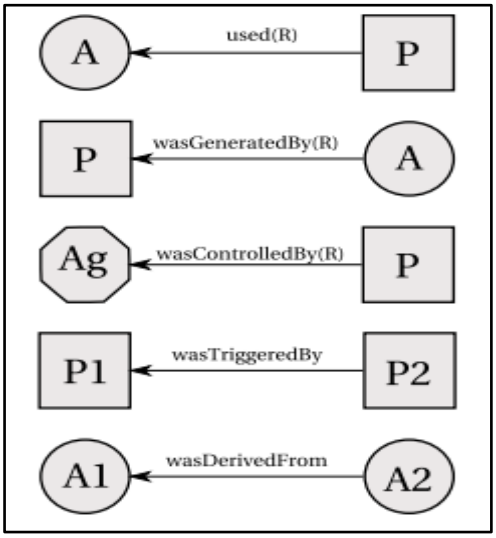


Figure 11 OPM Dependencies [41]

4.1.1.2 PROV model

PROV is a generic provenance model proposed by World Wide Web Consortium (W3C) in 2013 [32]. It has displaced OPM to consolidate provenance information interchange on the web and between heterogeneous systems. PROV at its core provides information about entities, activities, and agents involved in the usage or generation of an object as in Figure 12 [43].

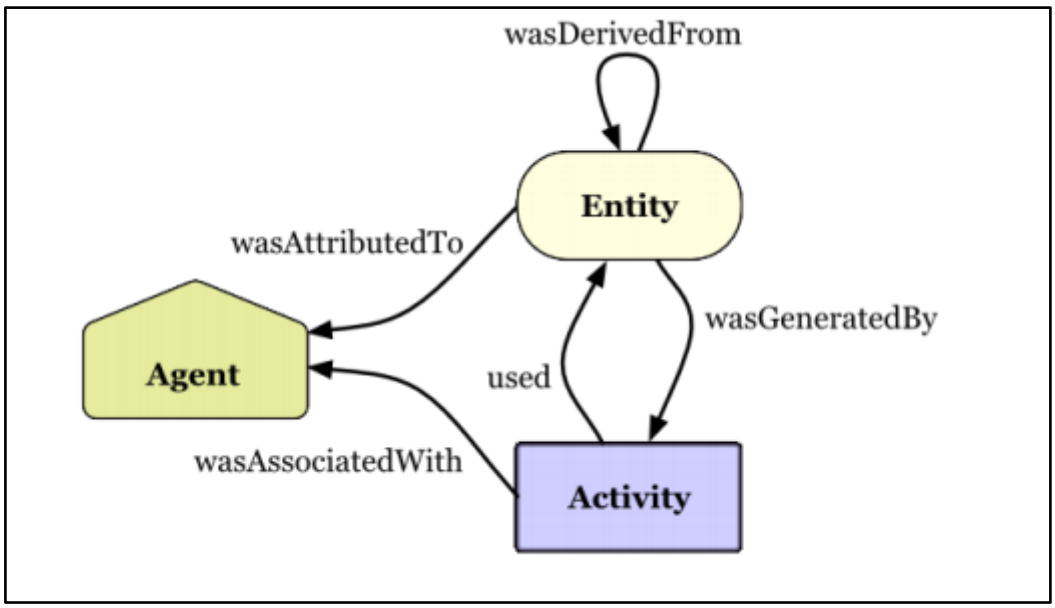


Figure 12 PROV Core Structures [43]

PROV Data Model (PROV_DM) presents exact definitions of the PROV core structures used in provenance representation explained below:

- Entities: represent any physical, digital, real, or any other kind of objects that can be used or produced by activities and described in provenance records [43].
- Activities: represent processes, actions or any kind of function that may produce an entity or change its attributes [43].
- Agents: agents are users, software pieces or any other kind of entity that may be assigned a role or responsibility of an activity [43].
- Roles: roles describe the nature of the relationship between an entity and an activity or between an agent and an activity [43].
- Usage: describes the usage of an entity by an activity [43].
- Generation: describes the generation of an entity by an activity [43].
- Derivation: when an entity is affected by another activity in terms of its existence, content, etc. then we say that it is derived from the other entity [43].

- Revision: when an entity get revised many times and one or more of its parameters or contents get changed, then there is more than one revision of this entity [43].
- Plans: predefined procedures being followed by the entity [43].

4.2 *Approach description*

The approach we are proposing in this work aims to improve the revenue assurance investigation detective process that is being executed to identify the root-cause of an anomaly observed by the rest of revenue assurance detective processes [57].

Figure 13 presents an overview of how the proposed provenance model works. Once an RA detective process starts execution the query model starts capturing semantic and contextual provenance information of each of its sub-processes, and store these information into data-oriented workflows in the Neo4j graph database.

Collected Data:

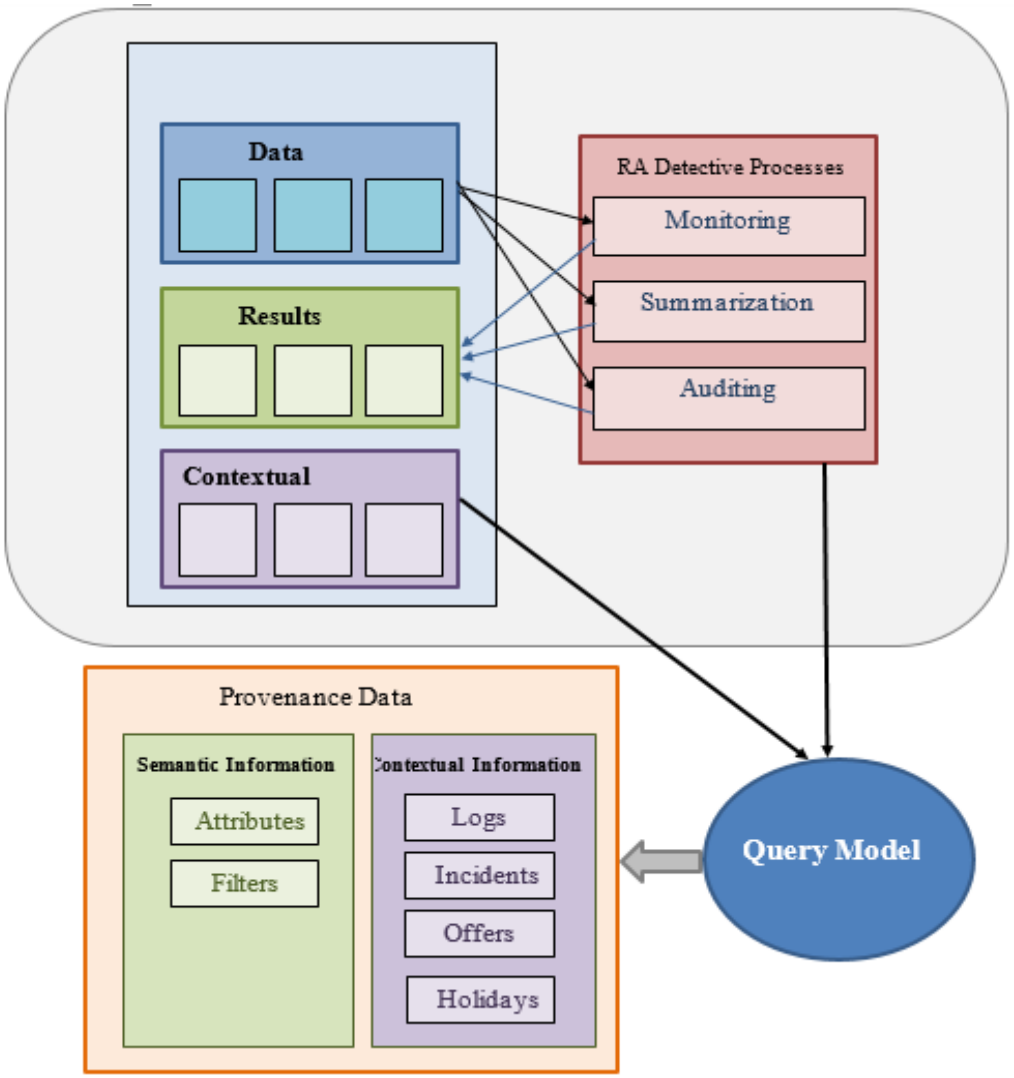


Figure 13 Proposed provenance-based approach

4.2.1 Conceptual model

This section presents an overview of the conceptual model developed in our approach as illustrated in Figure 14 with an aim to capture all provenance information during RA functions execution eagerly, meaning that the data is being collected during the execution of processes all the time, not only once a leakage issue occur. PROV standard model was our reference to design a model for revenue leakage detection in

telecom data workflows with the aim of representing data entities, traces, and workflows.

Workflow entities are represented as classes with the same Entity class specification in the PROV ontology, whereas the traces followed while executing the RA function are represented in a new class named Traces, and the involved processes during the execution of the function that requires taking input data and producing output data are represented by the class called Function. Therefore, the conceptual model is divided into three parts, each of them achieve one of the representation goals with associated classes and relation as in Table 1.

Entities	Workflows		Traces	
Classes	**Classes**	**Relations**	**Classes**	**Relations**
Entity	Function	hasInData	Traces	Used
				wasGeneratedBy
Data			Usage	qualifiedUsage
	Workflow	hasOutData		hasInData
Graph			Generation	hadEntity
				qualifiedGeneration

Table 1 Conceptual Model Parts and Element

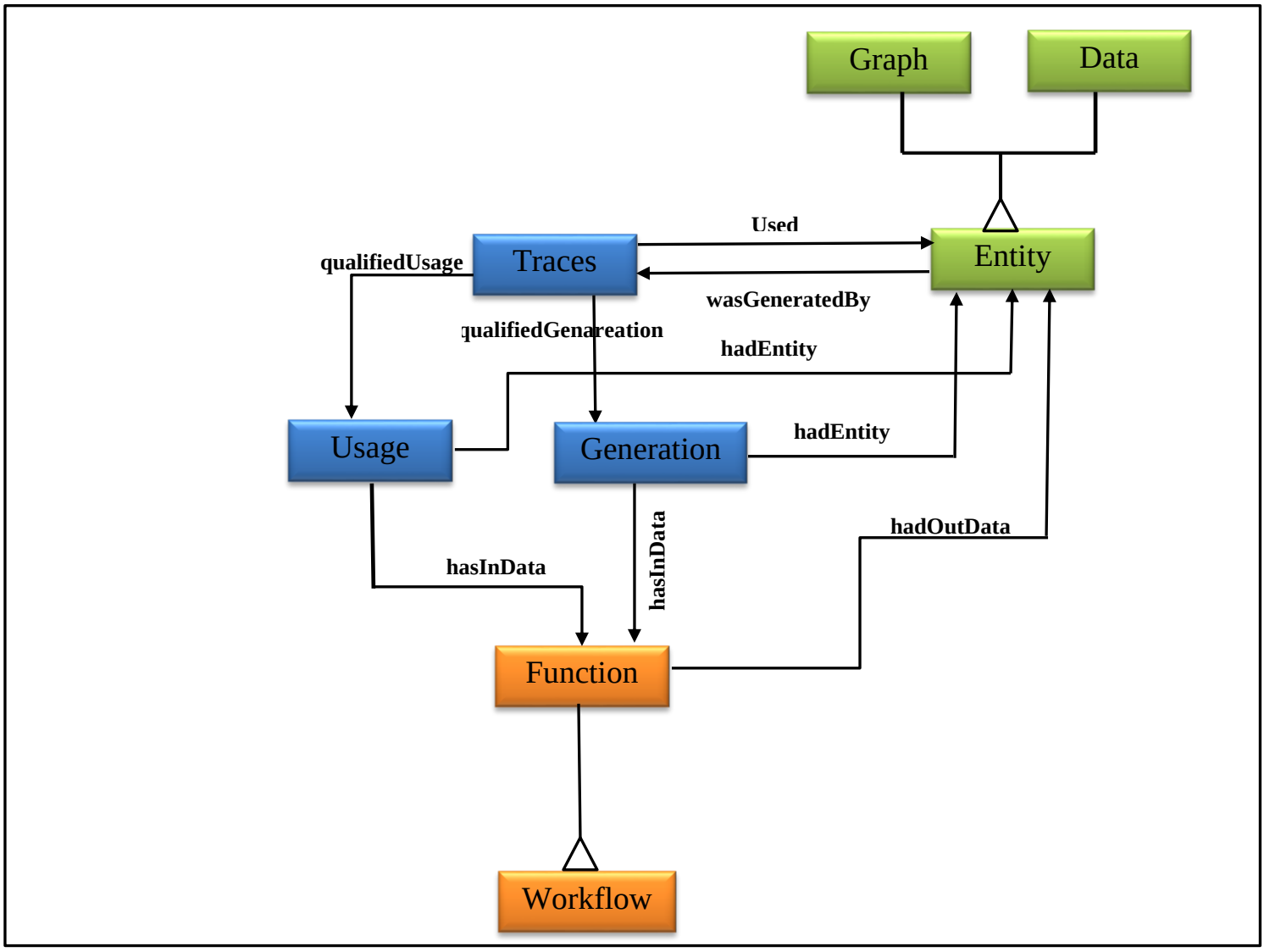


Figure 14 Conceptual Model

This section presents the model specifications in terms of classes and relations that forms the model.

- Entity class is specified in PROV ontology and represents anything with some fixed aspects, could be real or imaginary. In our model data represents tables, SQL attributes, SQL queries, and graph workflows.
- Data class represents the information to be processed or produced by the Function class.
- Graph class represents the output data as graph workflows by the Function class.

- Function class stands for a complete function or task completed by RA system requiring taking data as input and producing output data.
- Workflow class is considered as a Function, since it involves processing inputs and producing outputs.
- Traces class is the execution of a Function, and if the Function is a workflow then its execution would generate a trace.
- Usage class is specified in PROV ontology and it represents the start of entity utilization by an activity.
- Generation class is specified in PROV ontology and it represents a new entity production completion by an activity.
- Used property is specified in PROV ontology to declare that an Execution has used a particular Entity as input for its execution.
- wasGeneratedBy property is specified in PROV ontology to declare that an Execution has produced a particular Entity as output with its execution.
- qualifiedUsage property is specified in PROV ontology and represents a qualification of how an Activity has used an Entity.
- qualifiedGeneration property is specified in PROV ontology and represents a qualification of how an Activity has generated an Entity.
- hadEntity property determines the Entity used by a Trace.
- hasInData property determines the Entity used by a Function.
- hasOutData property determines the Entity generated by a Function.

4.2.2 Provenance workflow building algorithm

This algorithm pseudo code is used to build the provenance data-oriented workflow during the execution of any RA function based on the syntactic of analysis of all SQL processing nodes included:

```
For each SQL SELECT statement in RA function transformation:
    Do syntactic analysis of the SELECT statement
        Return (source-node, destination-node, rel_type, attributes, Filter)
    Create source-node if not exist
    Create destination-node if not exist
    Create processing-node (attributes, Filter)
        Create relationship (processing-node, source-node, rel_type='Uses')
        Create relationship (destination-node, processing-node, rel_type='Generated-From')
```

4.2.3 Provenance capturing

4.2.3.1 Capturing semantic provenance information

The query model does syntactic analysis on each SQL SELECT statement of sub-processes to determine its input and output tables, the mapped attributes between the two tables and used filter in the SELECT statement. The input and output tables are then represented as two data entities in the workflow connected by SQL processing entities and relationships among these entities. Semantic provenance information are represented by the mapped attributes and used filter in the SELECT statement and they are stored at processing node in the workflow. Attribute mapping between each input table I to the processing entity and output table O from to the processing entity means

that for each output table with attribute O.Y= v value is affected only by elements in the input table where I.X=v value. Therefore, the two tables are mapped by I.X↔O.Y. Another semantic to be preserved is the filter used at each processing entity during the transformation process. Denoting that for any output table O, input table I and filter C that is used to generate O from I. Each tuple in O is affected only by tuples in I that satisfies the filter C.

To clarify what semantic data in our model represent, the below SQL statement would be used to create two data entities for MSC and Duration_per_Destination in the workflow connected via a SQL processing nodes with semantic provenance information (Mapped_Attributes= [Date, Destination], Filter= "Traffic_Type='Voice' AND Record_Type='MOC' AND Profile='Prepaid'") stored at the processing node.

```
Create Table Duration_Per_Destination As
Select Date, Destination
From MSC
Where Traffic_Type='Voice' AND Record_Type='MOC' AND Profile='Prepaid'
Group By Destination
```

4.2.3.2 Capturing contextual provenance information

Contextual provenance information are meant to provide additional information about the data circumstances. For example, if our point of interest for a leakage problem is the MSC, then we have to check contextual information related to MSC on the same date in terms of incidents, changes, and upgrades. Therefore, in the approach we are proposing all source data entries in the workflow are to be associated with other three additional entities (Offers, System logs, and Incidents) for contextual information provision. Since source data entities represent source nodes and systems in the telecom

network as they are the source of data and at least one of them would always exist on the workflow assuring the association with contextual information provision entities described below would enrich the provenance data significantly once their relationship rules achieved.

- Offers entity is represented as a table in SQL database containing all launched offers, their launch date, end date, nodes that would be affected by each offer, affected traffic type, and the profile of affected subscribers. If all the affected rows generated while back tracing the leakage issue through the workflow satisfy the rules of the offers table for the day of the leakage problem, then a prediction is returned that the leakage issue or anomaly may be due to an offer launch.
- System logs entity is represented as a table in SQL database containing all systems logs, their execution start date and time, their execution end date and time, nodes that would be affected by each offer, affected traffic type, and the profile of affected subscribers. If all the affected rows generated while back tracing the leakage issue through the workflow satisfy the rules system logs table for the day of the leakage problem, then a prediction is returned that the leakage issue or anomaly may be due to system changes.
- Incidents entity is represented as a table in SQL database containing all occurred incidents, their start date, their end date, nodes that would be affected by each offer, affected traffic type, and the profile of affected subscribers. If all the affected rows generated while back tracing the leakage issue through the workflow satisfy the rules of the offers table for the day of the leakage problem, then a prediction is returned that the leakage issue or anomaly may be due to an incident occurred on the same date.

Furthermore, each final data item in the workflow has to be associated with a public holidays and events calendar table on the date attribute (Attribute mapping is done based on the date attribute), since such events would greatly affect the usage trends. For example, Eid holiday would double the voice and usage, while an industrial strike for political reasons would reduce the usage dramatically. Therefore, adding such contextual information is of great value for root-cause analysis, business understanding, and informed decision making.

4.2.4 Provenance tracing

Provenance tracing is performed backward through the tracing path in the generated provenance workflow during the RA function execution, which is specified according to the order of execution of the processing entities from the source dataset to the final destination dataset as in Figure 15. The tracing bath starts from the final destination entity going backward till the data source entity with the inclusion of all associated entities through the path. Given a record at the final dataset, to perform backward tracing, the model computes the join of all the tables in the path based on the logical provenance information consisting of attribute mappings and filters.

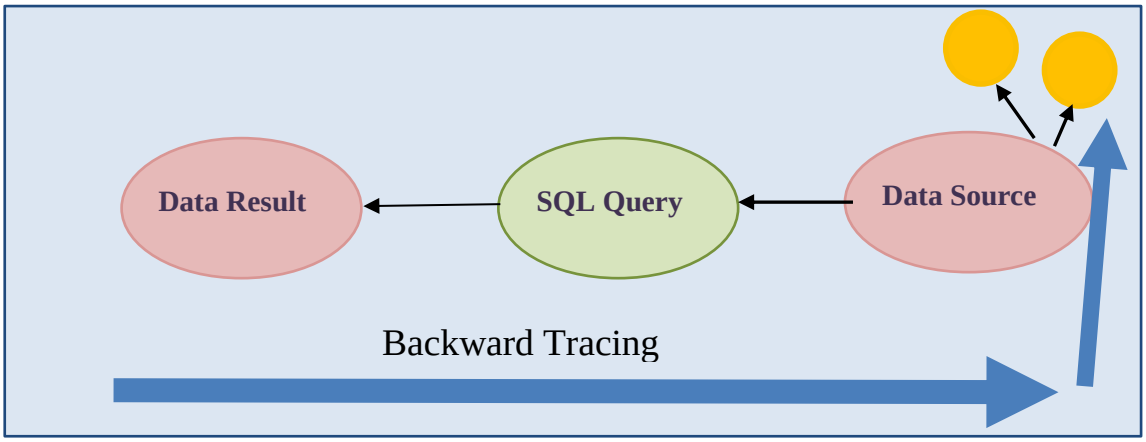


Figure 15 Backward tracing

4.2.4.1 Provenance tracing algorithm

After the execution of any detective RA process, the provenance data is being eagerly traced in our proposed model, by first getting all the SQL processing nodes and data nodes according to their execution order, reverse the order to get the final destination node and going backward through the nodes based on the algorithm below:

```
Get all provenance-workflow nodes
Reverse the nodes order
For each node in the reversed list:
    If node is a processing node:
        Get all associated nodes with 'Uses' type
For each associated node:
    If rel-type=='Uses':
        Get processing node attributes and filter
            Get affected records from input table and store in
        drill_down_table
If "select count(*) from drill_down_incidents where Filter" >0
    Then "An incident occurred"
Else if "select count(*) from drill_down_Offers where Filter" >0
    Then "An Offer was launched"
Else if "select count(*) from drill_down_Logs where Filter" >0
    Then "A change was introduced"
Else if "select count(*) from drill_down_Public_holidays_Events where Filter" >0
    Then "An event or a holiday occurred"
Return all drill_down tables for drill down
```

4.3 *Scenarios*

The review of the existing RA systems suggests that they lack the provenance capturing capability to answer the questions of when, where, and why the issue has been introduced and what reason(s) may have caused the revenue leakage issue. Noting that some usage monitoring trends may indicate abnormal behavior that may be due to a revenue leakage and needs to be immediately investigated, the human driven investigation may show that the alert of the revenue leakage is fake and that the abnormal behavior is due to some technical change or an offer launch, however it still needs to be investigated in order to ensure that no problem exists. On, the other hand, most of the times the alert indicates a real problem and needs an immediate investigation to find the reason of the leakage issue and take correction actions as soon as possible. Therefore, to assist us demonstrating our solution, we provide thirteen running examples for both cases (fake and real revenue losses alerts) with a description of the data collection process.

This first running example illustrates how provenance based approach in RA systems would easily and quickly find out the possible root-causes of an abnormal behavior in the usage monitoring trends that generates a fake alert of a revenue leakage issue providing better understanding of the usage traffic. Our second example illustrates how the provenance based approach would quickly indicates for possible reasons of a real problem caused missed charging of voice calls for prepaid calls due to an incident that caused network disconnection in order to immediately take corrective actions to stop the problem and take preventive actions in the future.

The third running example illustrates how the provenance based approach would quickly indicates for possible reasons of a real problem caused missed charging of voice calls, SMS calls, and GPRS sessions for prepaid subscribers due to CCN

disconnection caused by a power failure in an SDP site, in order to immediately take corrective actions to stop the problem and take preventive actions in the future. On the other hand, the fourth example explains how our approach would quickly predicts for possible reasons of a real problem caused miss-charging of voice calls, SMS calls, and GPRS sessions for prepaid subscribers due to CCN disconnection caused by a power failure in an SDP site.

The fifth example explains how our approach would quickly predicts for possible reasons of a real problem caused missed charging of voice calls, SMS calls, and GPRS sessions for postpaid subscribers due to a problem in the mediation system caused files not to be sent to the billing table. The sixth example explains how our proposed approach would easily and quickly find out the possible root-causes of an abnormal behavior in the usage monitoring trends that generates a fake alert due to a public holiday of a possible revenue leakage issue providing better understanding of the usage traffic.

The seventh example explains how our proposed approach would easily and quickly find out the possible root-causes of an abnormal behavior in the usage monitoring trends that generates an alert of a real revenue leakage issue due to prices change, providing better understanding of the usage traffic. The eighth and ninth examples explain how our proposed approach would easily and quickly find out the possible root-causes of an abnormal behavior in the usage monitoring trends that generates an alert of a real revenue leakage issue due to a wrong tariff configuration for different telecom traffic types, providing better understanding of the usage traffic.

The tenth example illustrates how our proposed approach would easily and quickly find out the possible root-causes of an abnormal behavior in the usage monitoring trends that generates an alert of a fake revenue leakage issue due to a new

product launch for rounding units of voice calls to 90 seconds instead of 60 seconds for the same price, providing better understanding of the usage traffic.

The eleventh example illustrates how our proposed approach would easily and quickly find out the possible root-causes of an abnormal behavior in the usage monitoring trends that generates an alert of a revenue leakage issue due to the launch of conflicting campaigns for international traffic that affect the estimated revenues for each other.

The twelfth example illustrates how our proposed approach would easily and quickly find out the possible root-causes of an abnormal behavior in the usage monitoring trends that generates an alert of a revenue leakage issue due to a great GPRS traffic increase and total charges decrease due to external fraud using fraudulent application.

4.3.1 Fake (i.e. false positives) revenue leakage alert

As an example of fake revenue leakage alerts, in the first scenario a great increase in voice calls duration with no additional increase in call counts or call charges has been used as an example of an abnormal behavior in the usage monitoring trends, since any additional voice usage must introduce additional fees or charges in the normal case, thus must be investigated by the RA team, by going back to the raw data and CDRs to find the root-cause for this behavior and communicate with other teams to assure their findings. In our proposed scenario the calls duration increase is due to an offer launched for local voice calls for prepaid subscribers to get 3 free minutes in each charged calls.

4.3.2 Real (i.e. true positives) revenue leakage alert

As an example of real revenue leakage alerts, in the second scenario a great missing of charged voice calls in one of the main RA system functions that connects the MSC source node representing the switch and the CCN source node representing the charging system has been used as an example of a real revenue loss example that must be investigated immediately by RA team to find out its root-cause by going back to the raw data and CDRs and extract affected calls and subscribers and communicate with other teams to assure their findings and make further investigations in order to take corrective actions and stop the problem. In our proposed scenario the calls are not being charged starting from 9 PM due to a disconnection occurred between the switch and the charging system after an upgrade was done to MSC3 with a change in its default settings that were not adjusted properly and caused the disconnection.

Chapter 5 Implementation

5.1 *Overview*

This chapter describes the implementation of the approach. We have built a prototype subsystem that uses logical provenance to ensure debugging and drill down aspects. Here we will demonstrate how our model would generate the provenance information, executes provenance tracing, detect leakage issue, generate usage trends for an abnormal usage behavior indicating fake revenue loss, find the root-causes behind these issues, and generate data to be drilled down at each processing node. The model has been developed to handle a large amount of data on a 10 GB DB on Microsoft SQL Server Express. The dataset is stored in a DB in a number of tables per node (MSC, SASN, SMSC, Billing and CCN) illustrated in Figures 16-18, which represent data instances from network nodes and contextual information from offers table. Semantic information are represented as used filters and mapped attributes in the provenance diagrams, discussed in more details in the next section 5.2.

SQLQuery1.sql - lo...KTIEGN\wisam (54))

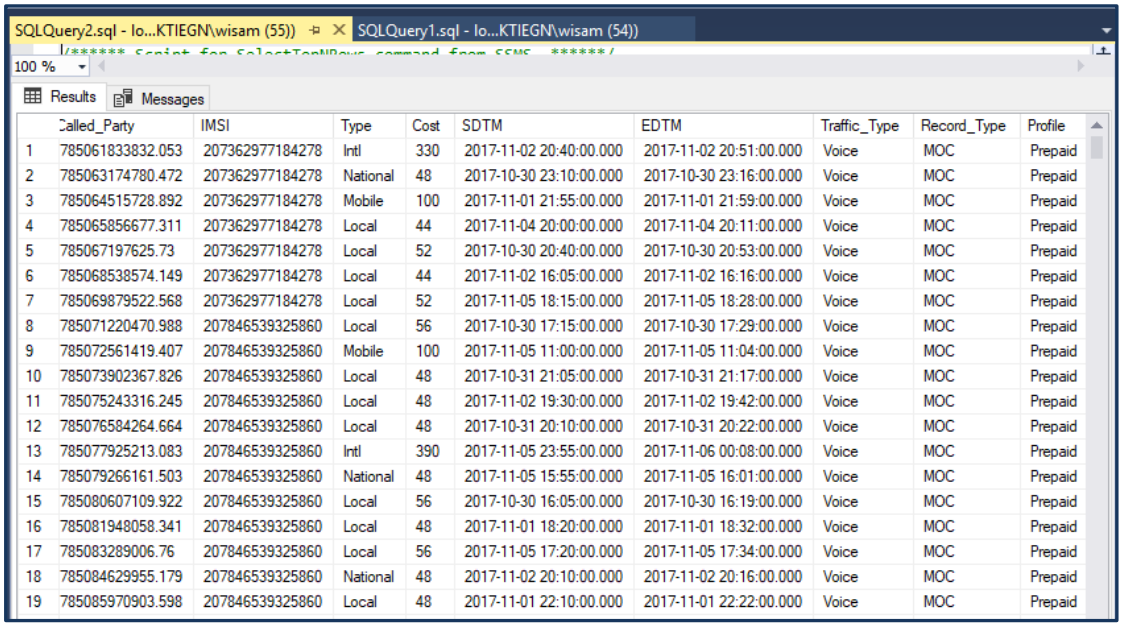
```
/****** Script for SelectTopNRows command from SSMS  ******/
```

100 %

Results | Messages

	ID	Calling_Party	Called_Party	IMSI	Type	SDTM	EDTM	Traffic_Type	Rec
1	4629187	2477111111	277743500303.149	207409465029788	National	2017-12-05 20:35:00.000	2017-12-05 20:37:00.000	Voice	MO
2	4629188	2477111111	277744841251.568	207409465029788	National	2017-12-09 05:05:00.000	2017-12-09 05:07:00.000	Voice	MO
3	4629189	2477111111	277746182199.987	207409465029788	Intl	2017-12-07 16:35:00.000	2017-12-07 16:42:00.000	Voice	MO
4	4629190	2477111111	277747523148.406	207409465029788	Local	2017-12-10 16:00:00.000	2017-12-10 16:02:00.000	Voice	MO
5	4629191	2477111111	277748864096.826	207409465029788	National	2017-12-07 21:30:00.000	2017-12-07 21:32:00.000	Voice	MO
6	4629192	2477111111	277750205045.245	207409465029788	Local	2017-12-06 21:30:00.000	2017-12-06 21:37:00.000	Voice	MO
7	4629193	2477111111	277751545993.664	207409465029788	Mobile	2017-12-10 20:40:00.000	2017-12-10 20:40:00.000	Voice	MO
8	4629194	2477111111	277752886942.083	207409465029788	Local	2017-12-07 19:15:00.000	2017-12-07 19:22:00.000	Voice	MO
9	4629195	2477111111	277754227890.502	207409465029788	Mobile	2017-12-06 00:10:00.000	2017-12-06 00:10:00.000	Voice	MO
10	4629196	2477111111	277755568838.921	207409465029788	Local	2017-12-05 21:05:00.000	2017-12-05 21:12:00.000	Voice	MO
11	4629197	2477111111	277756909787.341	207409465029788	Local	2017-12-04 14:40:00.000	2017-12-04 14:44:00.000	Voice	MO
12	4629198	2477111111	277758250735.76	207409465029788	Local	2017-12-04 18:50:00.000	2017-12-04 18:54:00.000	Voice	MO
13	4629199	2477111111	277759591684.179	207409465029788	Local	2017-12-04 21:50:00.000	2017-12-04 21:54:00.000	Voice	MO
14	4629200	2477111111	277760932632.598	207409465029788	Local	2017-12-06 18:55:00.000	2017-12-06 19:04:00.000	Voice	MO
15	4629201	2477111111	277762273581.017	207409465029788	Local	2017-12-06 21:10:00.000	2017-12-06 21:19:00.000	Voice	MO
16	4629202	2477111111	277763614529.436	207409465029788	National	2017-12-07 20:50:00.000	2017-12-07 20:53:00.000	Voice	MO
17	4629203	2477111111	277764955477.856	207409465029788	Local	2017-12-05 17:35:00.000	2017-12-05 17:44:00.000	Voice	MO

Figure 16 MSC Table



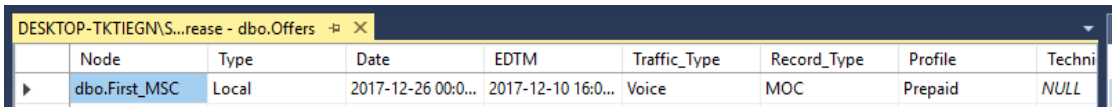
SQLQuery2.sql - lo...KTIEGN\wisam (55)) | SQLQuery1.sql - lo...KTIEGN\wisam (54))

100 %

Results | Messages

	Called_Party	IMSI	Type	Cost	SDTM	EDTM	Traffic_Type	Record_Type	Profile
1	785061833832.053	207362977184278	Intl	330	2017-11-02 20:40:00.000	2017-11-02 20:51:00.000	Voice	MOC	Prepaid
2	785063174780.472	207362977184278	National	48	2017-10-30 23:10:00.000	2017-10-30 23:16:00.000	Voice	MOC	Prepaid
3	785064515728.892	207362977184278	Mobile	100	2017-11-01 21:55:00.000	2017-11-01 21:59:00.000	Voice	MOC	Prepaid
4	785065856677.311	207362977184278	Local	44	2017-11-04 20:00:00.000	2017-11-04 20:11:00.000	Voice	MOC	Prepaid
5	785067197625.73	207362977184278	Local	52	2017-10-30 20:40:00.000	2017-10-30 20:53:00.000	Voice	MOC	Prepaid
6	785068538574.149	207362977184278	Local	44	2017-11-02 16:05:00.000	2017-11-02 16:16:00.000	Voice	MOC	Prepaid
7	785069879522.568	207362977184278	Local	52	2017-11-05 18:15:00.000	2017-11-05 18:28:00.000	Voice	MOC	Prepaid
8	785071220470.988	207846539325860	Local	56	2017-10-30 17:15:00.000	2017-10-30 17:29:00.000	Voice	MOC	Prepaid
9	785072561419.407	207846539325860	Mobile	100	2017-11-05 11:00:00.000	2017-11-05 11:04:00.000	Voice	MOC	Prepaid
10	785073902367.826	207846539325860	Local	48	2017-10-31 21:05:00.000	2017-10-31 21:17:00.000	Voice	MOC	Prepaid
11	785075243316.245	207846539325860	Local	48	2017-11-02 19:30:00.000	2017-11-02 19:42:00.000	Voice	MOC	Prepaid
12	785076584264.664	207846539325860	Local	48	2017-10-31 20:10:00.000	2017-10-31 20:22:00.000	Voice	MOC	Prepaid
13	785077925213.083	207846539325860	Intl	390	2017-11-05 23:55:00.000	2017-11-06 00:08:00.000	Voice	MOC	Prepaid
14	785079266161.503	207846539325860	National	48	2017-11-05 15:55:00.000	2017-11-05 16:01:00.000	Voice	MOC	Prepaid
15	785080607109.922	207846539325860	Local	56	2017-10-30 16:05:00.000	2017-10-30 16:19:00.000	Voice	MOC	Prepaid
16	785081948058.341	207846539325860	Local	48	2017-11-01 18:20:00.000	2017-11-01 18:32:00.000	Voice	MOC	Prepaid
17	785083289006.76	207846539325860	Local	56	2017-11-05 17:20:00.000	2017-11-05 17:34:00.000	Voice	MOC	Prepaid
18	785084629955.179	207846539325860	National	48	2017-11-02 20:10:00.000	2017-11-02 20:16:00.000	Voice	MOC	Prepaid
19	785085970903.598	207846539325860	Local	48	2017-11-01 22:10:00.000	2017-11-01 22:22:00.000	Voice	MOC	Prepaid

Figure 17 CCN Table



DESKTOP-TKTIEGN\S...rease - dbo.Offers

	Node	Type	Date	EDTM	Traffic_Type	Record_Type	Profile	Techni
▶	dbo.First_MSC	Local	2017-12-26 00:0...	2017-12-10 16:0...	Voice	MOC	Prepaid	*NULL*

Figure 18 Offers table contextual information

The provenance based debugging and drill-down for revenue leakage detection approach presented in this study has been developed using Python programming language implemented on Python 2.7.13 running on a personal computer with windows 10 Pro and 64-bit operating system with intel core (i7) and 2.4 GHZ processor and 8GB memory.

As shown in Figure 19, once an RA detective process starts, the execution of the query model starts capturing semantic and contextual provenance information of each of its sub-processes, and store these information into data-oriented workflows as provenance diagrams are built, stored, and queried using a data oriented workflow processing architecture (e.g. Neo4j graph database [37], py2neo [50]).

Simulated representative and descriptive datasets of real cases were generated using a call detail generation tool. Microsoft SQL Server Express 2017 for database

management, datasets storage and manipulation and SQL language were used for CDRs query. Microsoft Excel was used for analysis purposes.

Evaluation of the completeness and correctness of our approach is done by applying the developed model on different near real cases, then evaluating results using the gold standard approach.

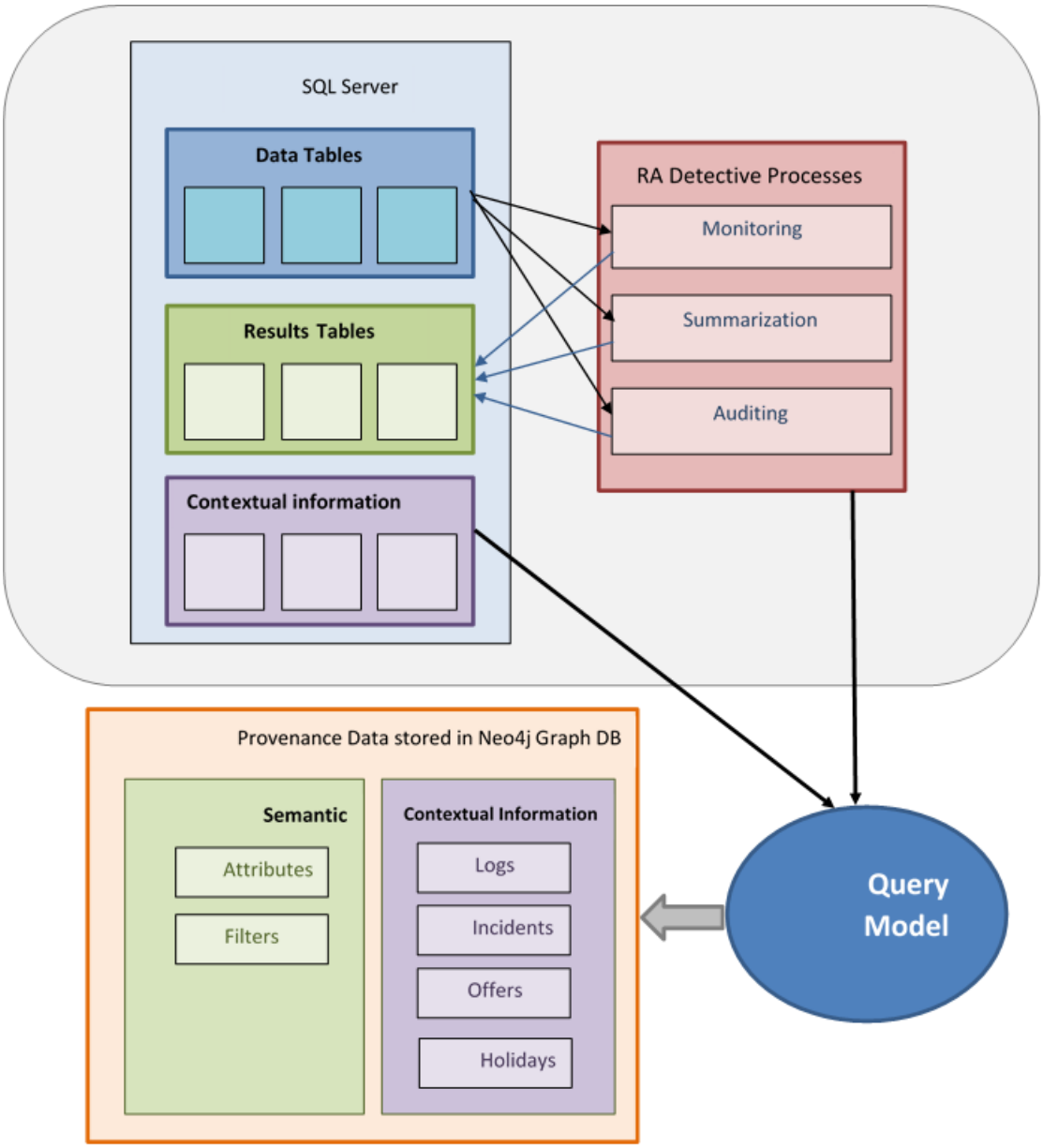


Figure 19 Proposed provenance-based approach

Regarding provenance diagrams, they are automatically built using Python programming language and Py2neo toolkit and stored in the Neo4j graph database, a code example is shown in Figure 20.

```
    for row in cursor.fetchall():
        print row
        cursor.execute("insert into "+ str(Destination)+" values (?,?,?)", (SDate1, row[0], row[1]))
        cursor.commit()
    SDate1 +=datetime.timedelta(days=1)
    SDate2 +=datetime.timedelta(days=1)

Incidents = Node("Incidents", name="dbo.Incidents", table="dbo.Incidents")
graph.create(Incidents)
Logs = Node("Logs", name="dbo.Logs", table="dbo.Logs")
graph.create(Logs)
Source = Node("MSC", name=Table.join(m[1][0][1].From[0]))
graph.create(Source)
Source_Incidents = Relationship(Source, "Associated With", Incidents, Attribute="Date", Filter="")
graph.create(Source_Incidents)
Source_Logs = Relationship(Source, "Associated With", Logs, Attribute="Date", Filter="")
graph.create(Source_Logs)
Offers = Node("Offers", name="dbo.Offers", table="dbo.Offers")
graph.create(Offers)
SQL_Processing1 = Node("Processing", name=Destination)
graph.create(SQL_Processing1)
Source_SQL = Relationship(SQL_Processing1, "Generated from", Source, Attribute=Columns, Filter=Final_Filter)
graph.create(Source_SQL)
Source_Offers = Relationship(Source, "Associated With", Offers, Attribute="Date", Filter="")
graph.create(Source_Offers)

Select_stmt2="select Date, SUM(Total_Duration)from dbo.Result_Per_Destination Group By Date"
```

Figure 20 Python code to build provenance diagram using Py2neo library

5.2 *Dataset generation*

Datasets are generated using CDR generation tool that contains many parameters to define such as the number of accounts to use, start date and end date, calls distribution among days of the week, call types, peak period, off peak period, and calls parameters, such as:

- Call cost.
- Call duration.
- Call probability.
- Standard deviation.
- Mean.

Since these parameters are defined for each dataset, then the datasets are considered repetitive for the same used parameters

5.3 *Scenarios*

To ensure the sufficiency of the implementation of the approach, different types of scenarios are considered in terms of their leakage alerts as well as root causes. Representative Datasets are then generated and implemented within the system. These scenarios are used to create gold standard datasets, in which their actual alerts and root-causes are known, which are then are used to evaluate the completeness and accuracy of the approach against generated datasets, see chapter 5 for details.

5.3.1 Fake alerts scenarios

5.3.1.1 First use case scenario (Major duration increase)

In this scenario, a major increase in total calls duration for prepaid subscribers is recorded with no additional increase in call counts or call charges. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since logically any additional voice usage must introduce additional fees or charges in the normal case which is not happening here, indicating that a revenue leakage may have occurred and revenue losses are still ongoing.

Figure 21 shows the usage abnormal behavior, since the prepaid voice calls duration per day follows a specific trend, high calls duration at Sunday and Monday, much lower calls duration during Tuesday to Thursday, and very low calls duration on weekends. But on this trend, we notice that there are much duration increase starting from 26th-Dec, which is a Tuesday compare to other Tuesdays. When we take a look at

the calls count and calls charge, we do not notice any proportional increase to this duration increase as in Figures 22-23, which is unrealistic in the normal case.

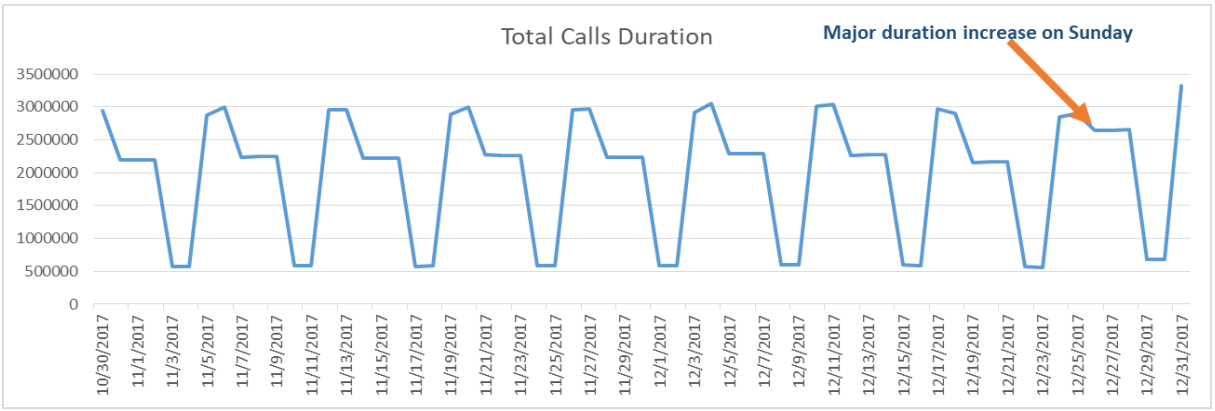


Figure 21 Calls Duration

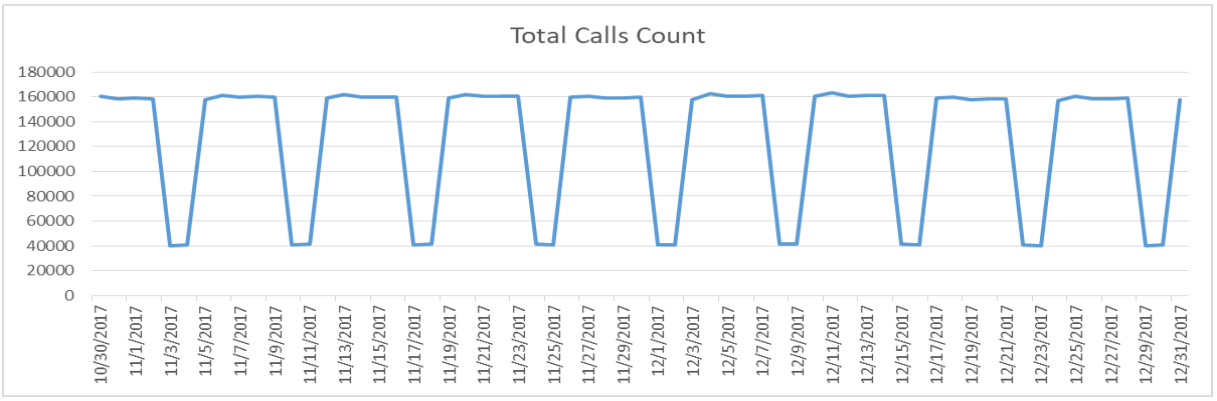


Figure 22 Calls Count

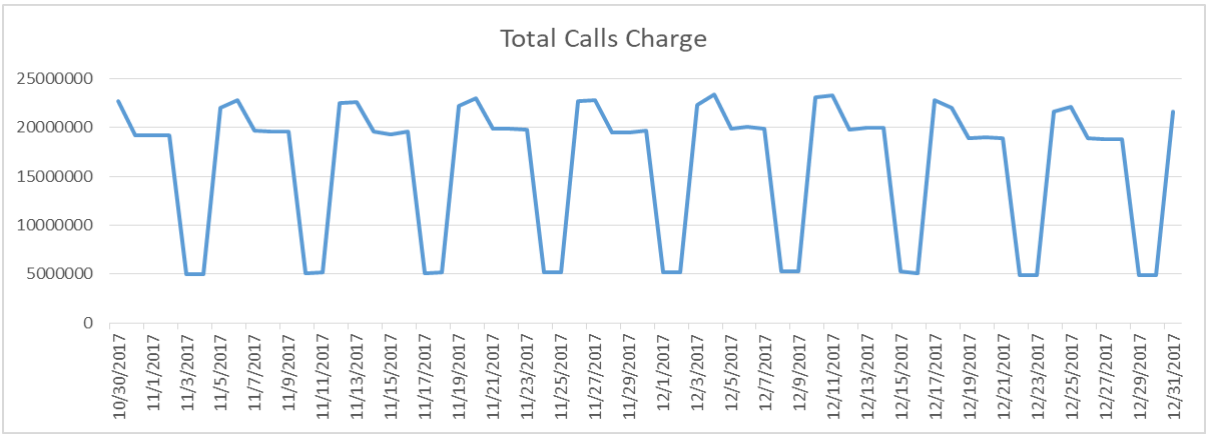


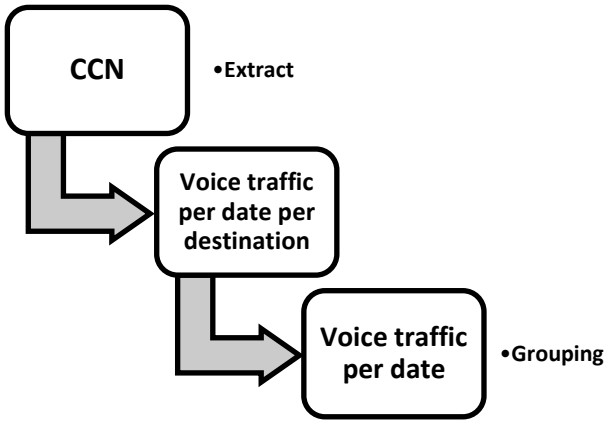
Figure 23 Calls Charge

Here we need to answer the questions of:

1. Why the duration increase has occurred?
2. When the revenue leakage issue has been introduced?

3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for prepaid usage traffic monitoring and trending performs a SQL query with a filter to extract only voice mobile originating traffic for prepaid subscribers towards all destinations then it groups them per destination type and per date afterwards as in below process.



The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figures 24 from the Neo4j after running the usage monitoring process for the date of interest.

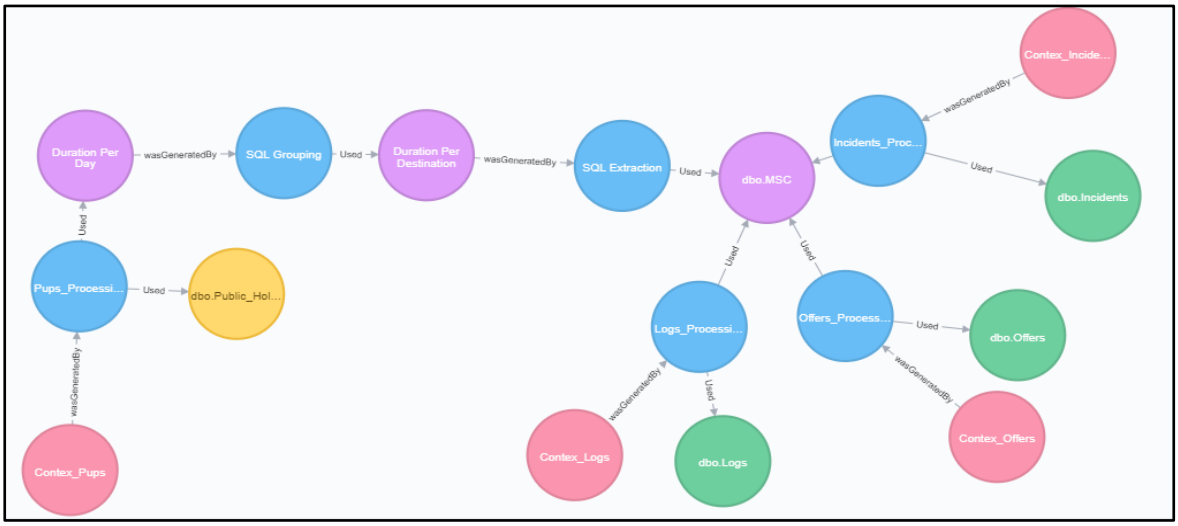


Figure 24 Provenance graph describing the execution of the data extraction query in the abnormal duration increase use case. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 24 describes the trace of the first scenario of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple node as a data source entity, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Duration_Per_Day Query Result entity at the left of the graph was produced by the Result_Per_Destination Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

To demonstrate the features of our approach, the dataset and processes done over it will be represented here. The input dataset for the first scenario will be extracted by a SQL query with a certain filter over MSC table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the

query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes as in Figures 25-26, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

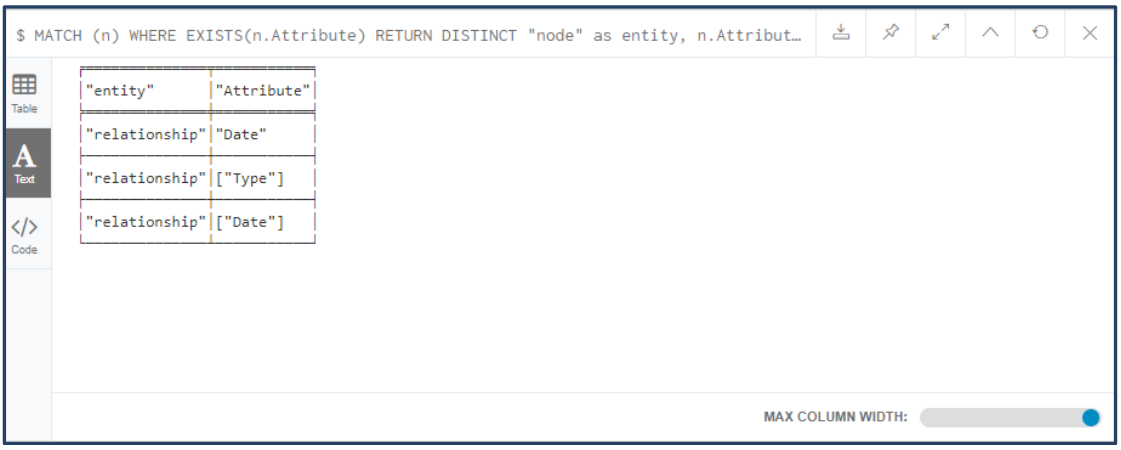


Figure 25 Mapped Attributes stored in SQL processing nodes

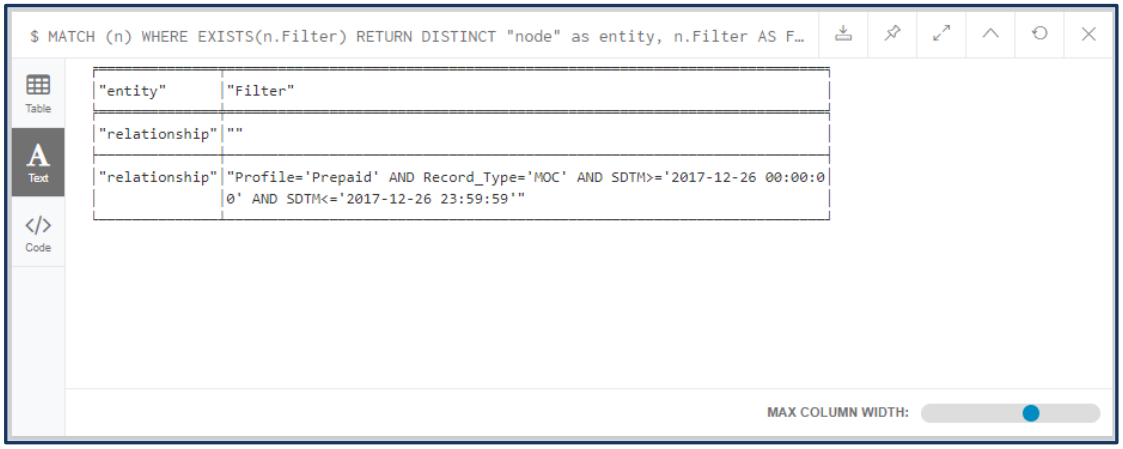


Figure 26 Used Filter stored in SQL processing

In the given scenario the abnormal value occurs starting from 26th-Dec. To back trace this value, the mapped attributes between the results table and the duration per destination are used to select the affected data in the drill down table, the filter and mapped attributes between duration per destination table and the MSC table are used to

select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the duration increase occurred after an offer for local destination calls was launched on 26th-Dec to get three free minutes for each charged call. And after the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue as in Figure 27.

```
Select COUNT(*) from dbo.First_MSC_Drill where Profile='Prepaid' AND Record_Type
='MOC' AND Traffic_Type='Voice' AND Type='Local'
Insert into dbo.Offers_Drill values ('dbo.First_MSC', 'Local', '2017-12-26 00:00
:00', '2017-12-10 16:00:00', 'Voice', 'MOC', 'Prepaid', 'None')
An offer was launched on 26-12-2017
```

Figure 27 Debugging result

The results answers the previously presented questions as:

1. Why the duration increase has occurred? **An offers was launched to grant free minutes.**
2. When the revenue leakage issue has been introduced? **26th-Dec.**
3. Where the revenue leakage issue has occurred? At what node? **MSC**.
4. What has caused this revenue leakage issue to occur? **New Offer for local prepaid calls.**

First Use Case process description:

MSC Table:

ID	Calling_Party	Called_Party	IMSI	Type	SDTM	EDTM	Traffic_Type	Record_Type	Profile	Call_Duration_Seconds	Call_Duration_Minutes	Switch
4629187	2477111111	277743500303	207409465029788	National	12/5/17 20:35	12/5/17 20:37	Voice	MOC	Prepaid	120	2	MSC3
4629220	1465555555	277787751601	207813422270201	Intl	12/10/17 18:35	12/10/17 18:41	Voice	MOC	Prepaid	360	6	MSC3
4629255	5933333333	277834684796	207224841871905	Local	12/6/17 21:10	12/6/17 21:18	Voice	MOC	Prepaid	480	8	MSC3

```
Select_stmt="SELECT Type, SUM(Call_Duration_Minutes) FROM dbo.First_MSC
where Profile='Prepaid' AND Record_Type='MOC'  AND SDTM >=
'"+ str(SDatel)+"' AND SDTM <='"+ str(SDate2)+"' GROUP BY Type"
```

Duration per destination per date Table:

Date	Distination	Total Duration
12/26/2017	Local	2047312
12/26/2017	Intl	187505
12/26/2017	Free	10362
12/26/2017	National	187292
12/26/2017	Mobile	212771

```
Select_stmt2='''select Date, SUM(Total_Duration)
from dbo.Result_Per_Destination Group By Date'''
```

Duration per date Table:

Date	Total
11/13/2017	2952445
11/14/2017	2227325
11/15/2017	2221449
11/16/2017	2224103
11/17/2017	578053

5.3.1.2 Second use case scenario (Major traffic increase)

In this scenario, a major increase in prepaid traffic in terms of records count, calls duration, and calls charge is recorded on 31st-Dec. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since the root-cause behind the traffic increase must

be identified to assure that subscribers are correctly charged and to better understand the business cases.

Figures 28-30 show the usage abnormal behavior, since the prepaid voice calls per day trends follow a specific pattern in terms of count, duration, and charge. But in this scenario, we notice that there are much count, duration, and charge increase on 31st -Dec, which is a Sunday compared to other Sundays and this behavior needs to be investigated to identify its root-cause and possible effects.

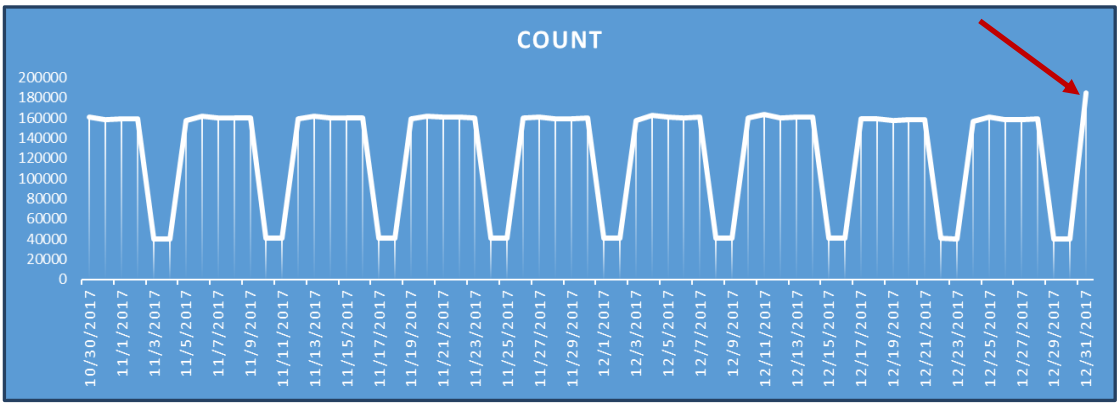


Figure 28 Prepaid voice calls count

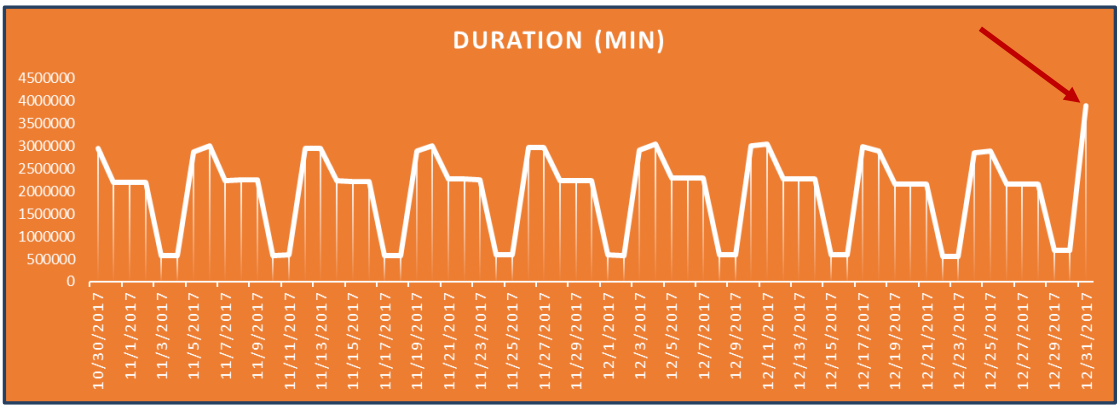


Figure 29 Prepaid voice calls duration

Figure 30 Prepaid voice calls cost

Here we need to answer the questions of:

1. Why the traffic increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for prepaid usage traffic monitoring and trending performs a SQL query with a filter to extract only voice mobile originating traffic for prepaid subscribers towards all destinations then it groups them per destination type and per date afterwards in terms of calls count, total calls duration, and total calls charge.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, as illustrated in Figure 31 from the Neo4j after running the usage monitoring process for the date 31st-Dec.

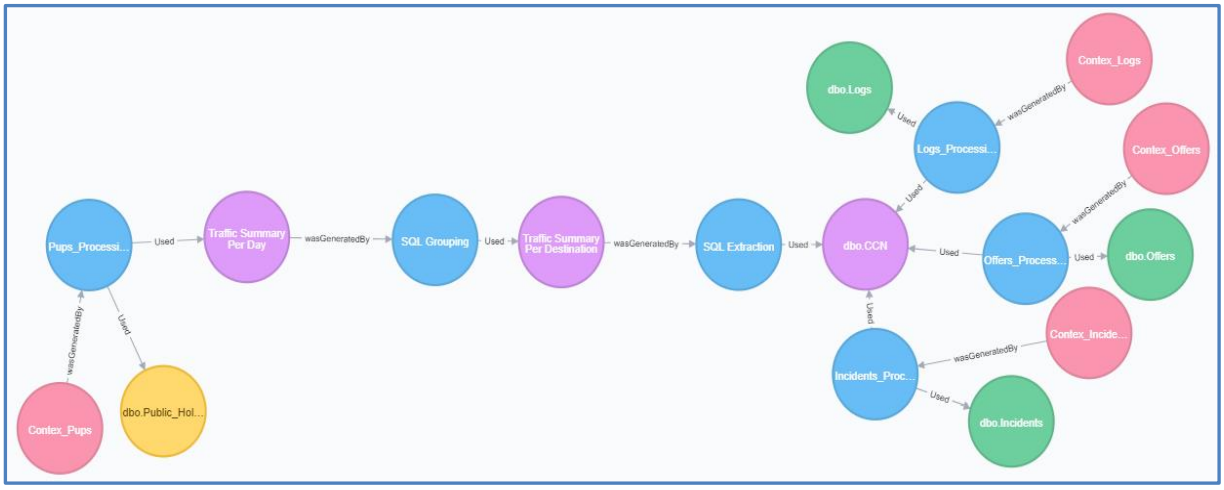


Figure 31 Provenance graph describing the execution of the data extraction query in the abnormal traffic increase use case. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 31 describes the trace of the sixth scenario of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple node as a data source entity, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Summary_Per_Day Query Result entity at the left of the graph connected to public holidays node and was produced by the Traffic_Summary_Per_Destination Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario will be extracted by a SQL query with a certain filter over CCN table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance

information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs at 31st -Dec. To back trace this value, the public holidays matched on the same day with the results table are mapped and the mapped attributes between the results table and the Traffic summary per destination are used to select the affected data in the drill down table, the filter and mapped attributes between duration per destination table and the CCN table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the duration increase occurred due to a public event (Christmas). And after the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue by identifying at the first step of back tracing that a public event has occurred on the same day of the abnormal behavior and may have caused the great traffic increase.

The results answers the previously presented questions as:

1. Why the duration increase has occurred? **A public holiday (Christmas).**
2. When the revenue leakage issue has been introduced? **31st -Dec.**
3. Where the revenue leakage issue has occurred? At what node? **CCN**.
4. What has caused this revenue leakage issue to occur? **A public holiday.**

5.3.2 Real alerts scenarios

5.3.2.1 Network and system errors leakage issues

5.2.2.1.1 Third use case scenario MSC disconnection (Missing from CCN)

In this scenario, a great revenue leakage has occurred due to a network disconnection between the switch and the charging system after an upgrade to MSC3. The disconnection caused the calls to be originated and passed through the MSC without triggering the charging system. In this scenario the MSC records are being generated and the calls do actually happen, but since the charging system is not triggered by the MSC, the calls will be free with no CCN records generated causing great revenue less unless the disconnection problem is triggered and solved. This scenario has been used as an example of real revenue leakage problem that needs to be detected, analysed to find the root-cause of leakage and solved. Figure 32 shows the gap of count of transactions for MSC compared to CCN on the date of the network disconnection 26th-Dec.

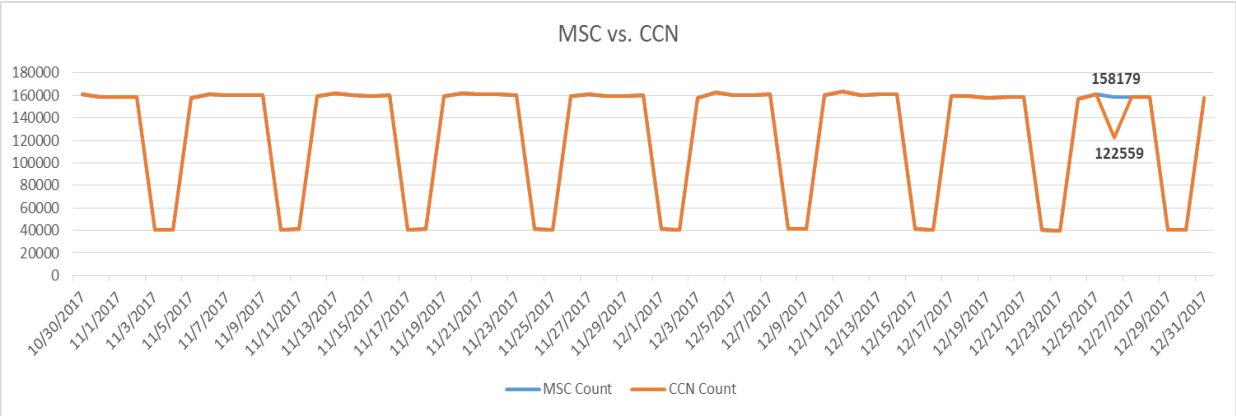


Figure 32 MSC vs. IN CCN records count comparison

The main audit responsible for the detection of such revenue leakage issue, extracts all prepaid voice calls from MSC side and extracts all voice calls from CCN side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted reports in terms of calling party, called party, call time, and duration to identify missing records from CCN side (Uncharged calls), missing calls from MSC side (Charged calls that did not happen), Mismatched calls between MSC and CCN (Overcharged or Undercharged calls).

Here we need to answer the questions of:

1. Why the revenue leakage issue has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The developed system prototype automatically builds a data workflow on the Neo4j graph database for the whole data match process, representing all data nodes included in the process, associated nodes and SQL processing nodes. Each SQL processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 33 from the Neo4j after running the second scenario process for the date of interest.

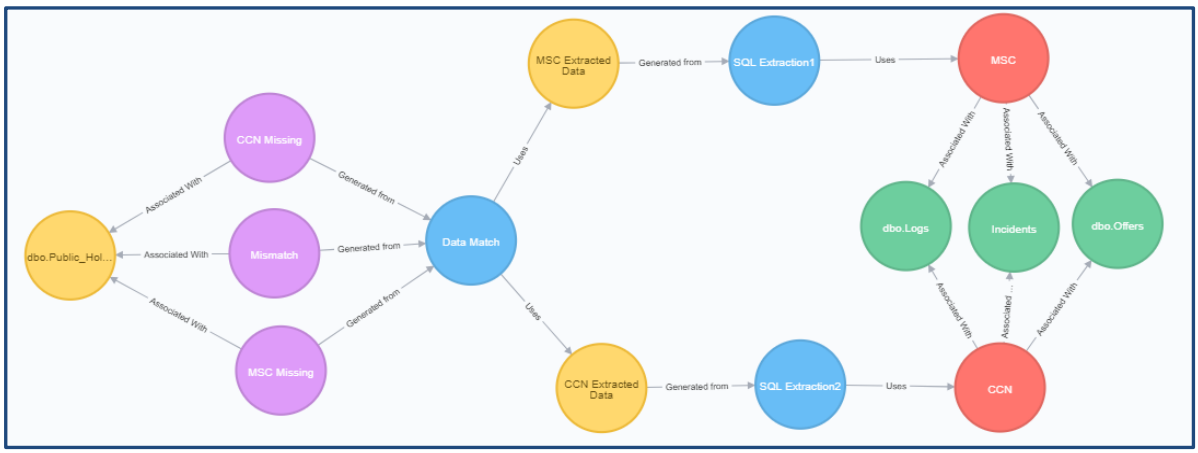


Figure 33 Provenance graph describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from CCN side caused by a network disconnection. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

The provenance graph shown in Figure 33 describes the trace of the second scenario of our model built using the Neo4j database, with blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node. The three missing and mismatch resulted data entities at the left of the graph were produced by the data match Query Execution, that was executed over the data generated by the previous Data Extraction processes containing the original SQL queries, which are connected to the data source entities that are associated to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario are extracted by two SQL queries with certain filters MSC and CCN tables for the date of leakage issue. At each step of execution the generated data is stored in a separate table to support drill down capability. After the queries are executed provenance information are generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provision entities. Nodes are automatically being added to the workflow and provenance semantic information are inserted into the SQL processing nodes, then the data extracted are used as an input for the data match query that compare the two extracted results based on the calling party, called party, and the time of call to generate the missing and mismatch reports. Backward tracing is executed if high missing or mismatch is detected on a certain date by retrieving all SQL processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities using the processing nodes to detected all affected rows and find possible root-causes for the leakage issue.

In the given scenario the network disconnection occurs on 26th-Dec starting at 9 PM. To back trace this leakage issue, the mapped attributes between the missing from CCN results table and the data match query are used to select the affected data in the drill down table, the filter and mapped attributes between data match table and the data extraction query tables are used to select the affected data from the drill down tables of MSC and CCN, then use the filter stored between incidents, logs, and offers tables to enrich the data obtained about leakage issue by finding that an MSC upgrade incident in the Incidents table occurred on 26th-Dec, that caused the revenue loss.

The results answers the previously presented questions as:

1. Why the revenue leakage issue has occurred? **An MSC upgrade**.
2. When the revenue leakage issue has been introduced? **26th-Dec**.

3. Where the revenue leakage issue has occurred? At what node? **MSC**.
4. What has caused this revenue leakage issue to occur? **An MSC upgrade incident**.

5.2.2.1.2 Fourth use case scenario CCN disconnection (Missing from CCN and mismatch with CCN)

In this scenario, a great revenue leakage has occurred due to a network disconnection on the charging system due to a power failure on SDP site on 5th-Dec. The disconnection caused calls, SMSs, and GPRS sessions to be originated and passed through the MSC, SMSC, and SASN without being charged by the charging system. Therefore, MSC, SMSC, and SASN records are being generated and do actually happen, but the transactions will be free of charge with no CCN records generated causing great revenue loss unless the disconnection problem is triggered and solved. This scenario has been used as an example of real revenue leakage problem that needs to be detected and analysed to find the root-cause of leakage and solve it as soon as possible. Figures 34-36 show the gap of count of transactions per traffic type for all nodes compared to CCN on the date of the power failure incident 5th-Dec.

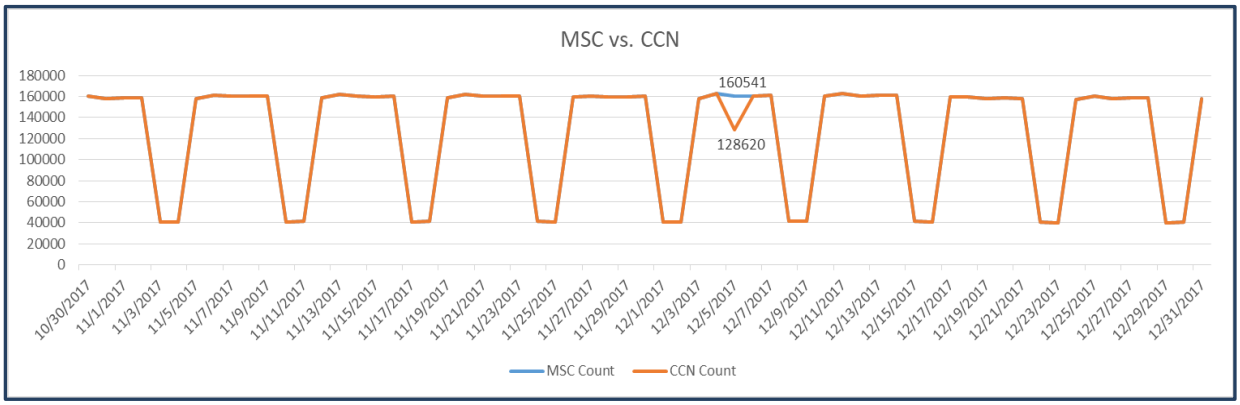


Figure 34 MSC vs. CCN Count of records

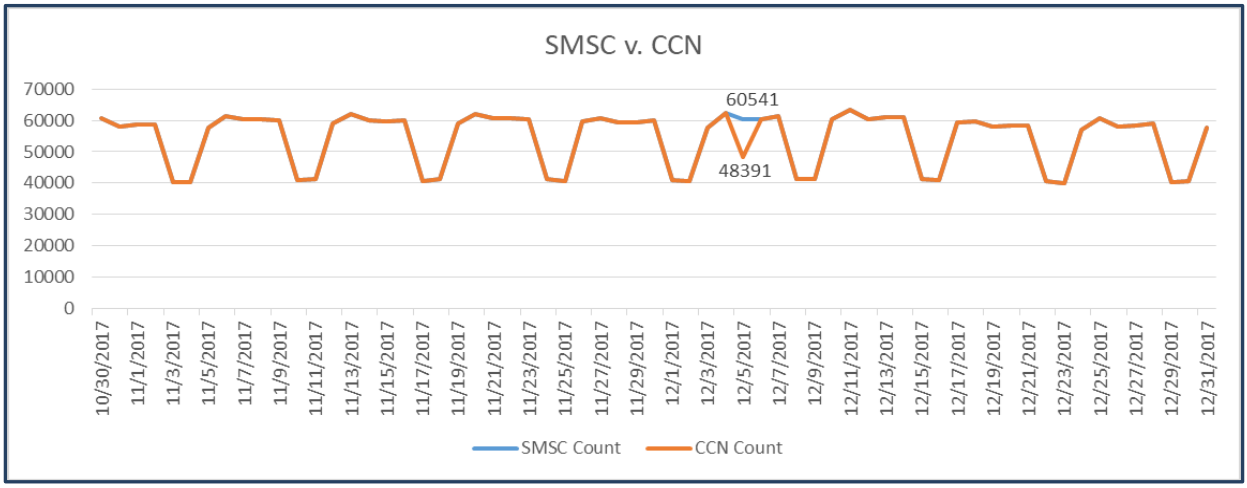


Figure 35 SMSC vs. CCN Count of records

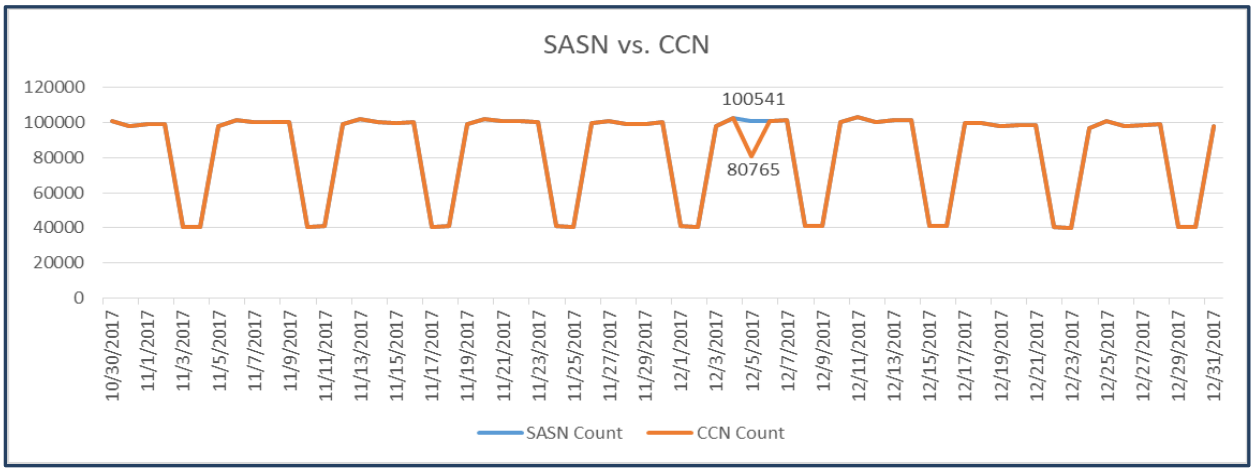


Figure 36 SASN vs. CCN Count of records

The main audits responsible for the detection of such revenue leakage issue, extract all prepaid voice calls from MSC side and extracts all voice calls from CCN side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted reports in terms of calling party, called party, call time, and duration to identify missing records from CCN side (Uncharged calls), missing calls from MSC side (Charged calls that did not happen), mismatched calls between MSC and CCN (Overcharged or Undercharged calls). The second audit extracts all prepaid SMS records from SMSC side and extracts all SMS records from CCN side for a certain date, does a count comparison between the extracted records to identify missing records from CCN side (Uncharged SMSs), missing records from

SMSC side (Charged SMSs that did not happen). And the final audit extracts all prepaid GPRS records from SASN side and extracts all GPRS records from CCN side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted to identify missing records from CCN side (Uncharged GPRS sessions), missing records from SASN side (Charged sessions that did not happen), Mismatched GPRS sessions between SASN and CCN (Overcharged or Undercharged sessions). Therefore, we have three sub-scenarios in this context and the main audits needed for revenue leakage detection are:

- MSC vs. CCN.
- SMSC vs, CCN.
- SASN vs. CCN.

Here we need to answer the questions of:

5. Why the revenue leakage issue has occurred?
6. When the revenue leakage issue has been introduced?
7. Where the revenue leakage issue has occurred? At what node?
8. What has caused this revenue leakage issue to occur?

The developed system prototype automatically builds a data workflow on the Neo4j graph database for the whole data match process for the three audits, representing all data nodes included in the process, associated nodes and SQL processing nodes. Each SQL processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters.

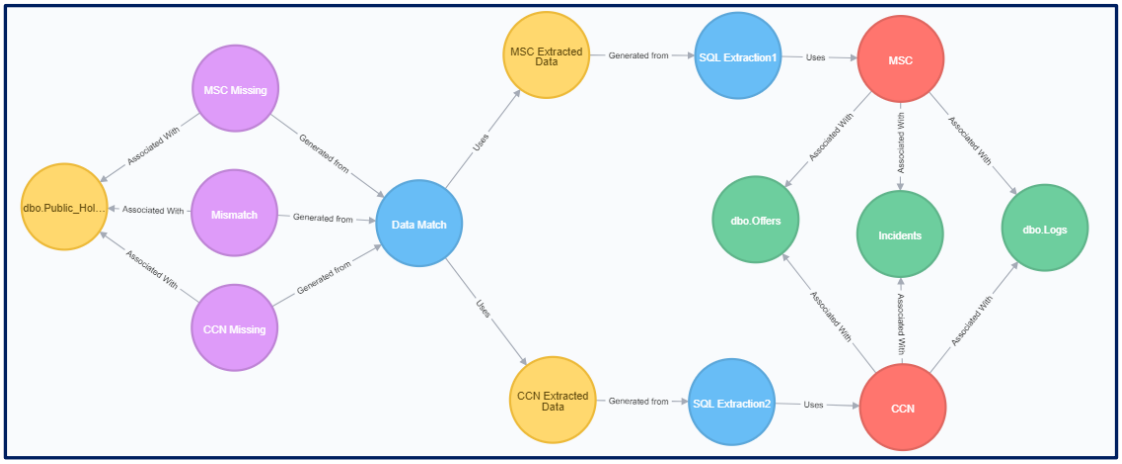


Figure 37 Provenance graph for MSC vs. CCN audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from CCN side caused by a network disconnection. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

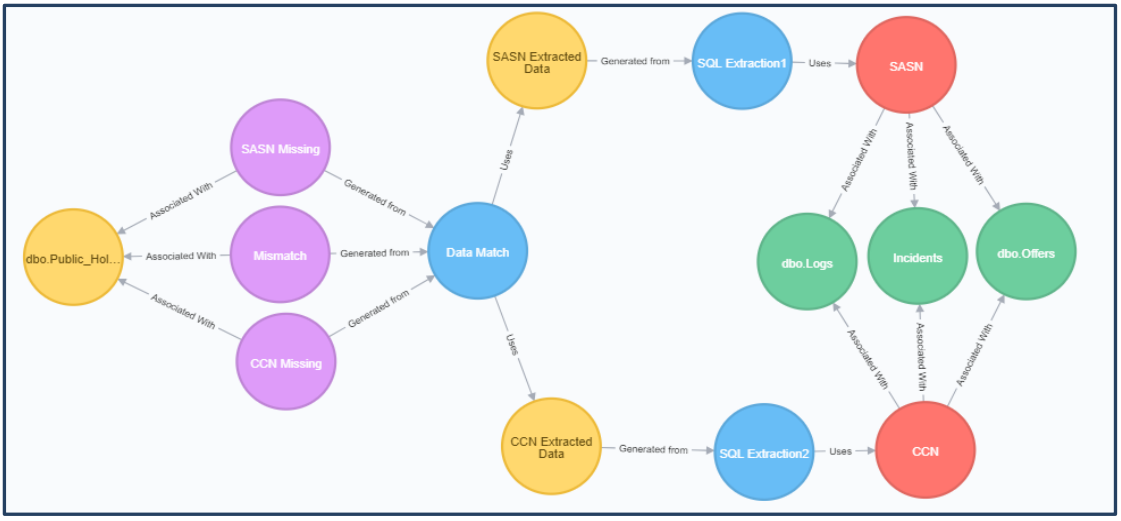


Figure 38 Provenance graph for SASN vs. CCN audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from CCN side caused by a network disconnection. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

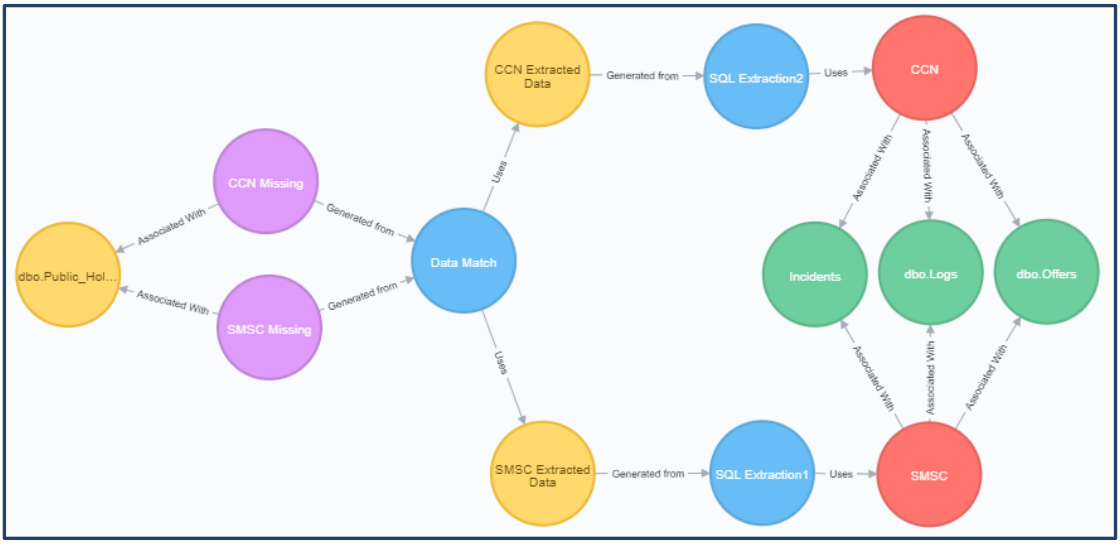


Figure 39 Provenance graph for SMSC vs. CCN audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from CCN side caused by a network disconnection. Blue nodes denote SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes, since they are 12 nodes)

The provenance graphs shown in Figures 37-39 describe the trace of the third scenario for the three audits of our model built using the Neo4j database, with blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node. The three missing and mismatch resulted data entities at the left of the graph are connected to public holidays node and were produced by the data match Query Execution, that was executed over the data generated by the previous Data Extraction processes containing the original SQL queries, which are connected to the data source entities that are associated to the

contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario are extracted by two SQL queries with certain filters MSC and CCN tables for the date of leakage issue. At each step of execution the generated data is stored in a separate table to support drill down capability. After the queries are executed provenance information are generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provision entities. Nodes are automatically being added to the workflow and provenance semantic information are inserted into the SQL processing nodes, then the data extracted are used as an input for the data match query that compare the two extracted results based on the calling party, called party, and the time of call to generate the missing and mismatch reports. Backward tracing is executed if high missing or mismatch is detected on a certain date by retrieving all SQL processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities using the processing nodes to detected all affected rows and find possible root-causes for the leakage issue. The same process is done for SMSC vs. CCN audit and SASN vs. CCN audit.

In the given scenario the CCN disconnection occurs on 5th-Dec starting at 9 PM. To back trace this leakage issue, the mapped attributes between the missing from CCN results table and the data match query are used to select the affected data in the drill down table, the filter and mapped attributes between data match table and the data extraction query tables are used to select the affected data from the drill down tables of MSC and CCN, SMSC vs. CCN, and SASN vs. CCN, then use the filter stored between incidents, logs, and offers tables to enrich the data obtained about leakage issue by

finding that a power failure incident in the Incidents table occurred on 5th-Dec, that caused the revenue loss.

The results answers the previously presented questions as:

1. Why the revenue leakage issue has occurred? **A power failure incident.**
2. When the revenue leakage issue has been introduced? **5th-Dec**.
3. Where the revenue leakage issue has occurred? At what node? **CCN**.
4. What has caused this revenue leakage issue to occur? **A power failure incident caused CCN disconnection**.

Moreover, in this scenario, the disconnection caused calls and GPRS sessions to be partially charged. Therefore, CDRs are generated from both nodes but the transactions are under charged as their duration and volume are smaller on the charging system rather than MSC and SASN. This scenario causes a great revenue loss unless the disconnection problem is triggered and solved. This scenario has been used as an example of real revenue leakage problem that needs to be detected, analysed to find the root-cause of leakage and get solved. Figures 40-41 show the gap between the total calls duration on MSC and the total calls duration on CCN and total GPRS sessions volume on SASN and total GPRS sessions volume on CCN on the date of the power failure incident 5th-Dec.

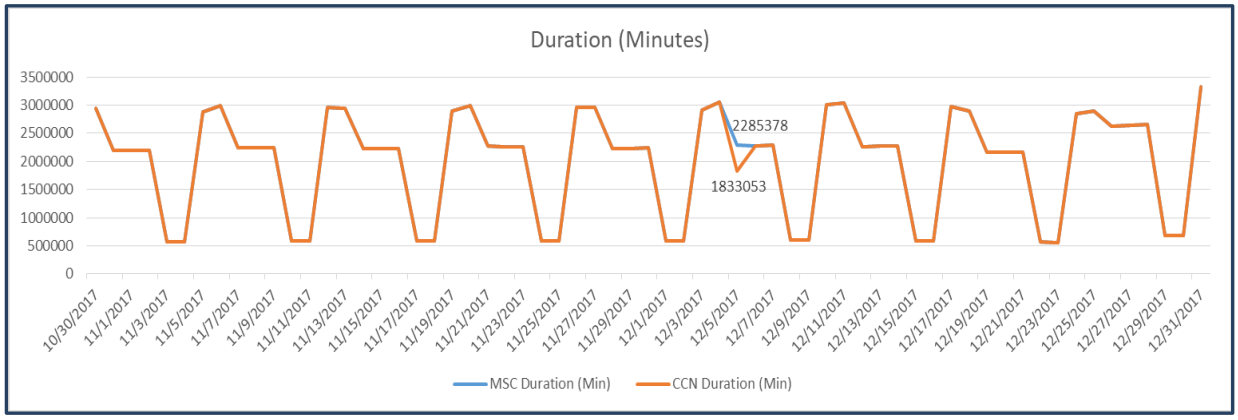


Figure 40 Total prepaid calls duration

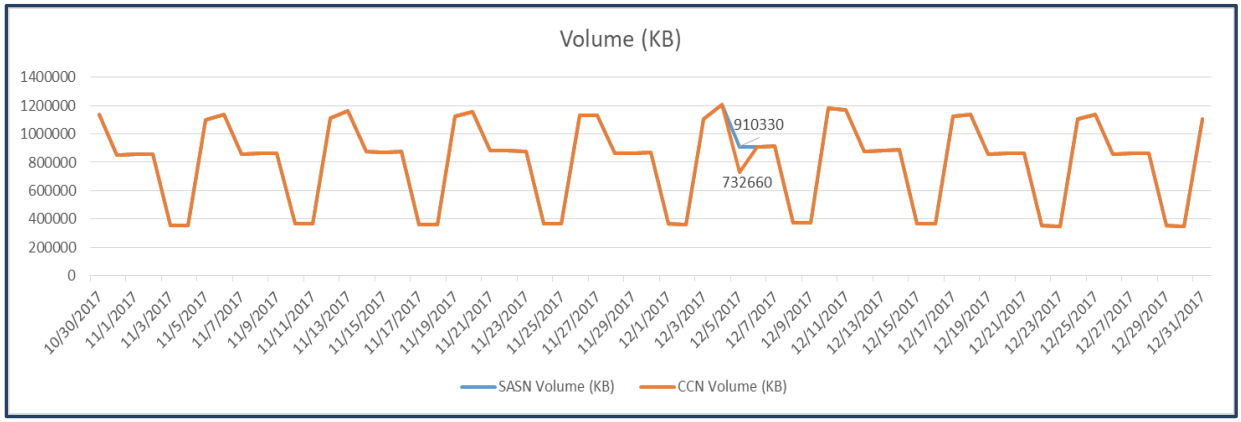


Figure 41 Total prepaid GPRS sessions volume

The main audits responsible for the detection of such revenue leakage issue, extracts all prepaid voice calls from MSC side and extracts all voice calls from CCN side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted reports in terms of calling party, called party, call time, and duration to identify missing records from CCN side (Uncharged calls), missing calls from MSC side (Charged calls that did not happen), Mismatched calls between MSC and CCN (Overcharged or Undercharged calls) by doing a compare equality check for matched records between the two sources in terms of duration for each call. The second audit extracts all prepaid GPRS records from SASN side and extracts all GPRS records from CCN side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted to identify missing records from CCN side (Uncharged GPRS sessions), missing records from SASN side (Charged sessions that did not happen), Mismatched GPRS sessions between SASN and CCN (Overcharged or Undercharged sessions) by doing a compare equality check for matched records between the two sources in terms of volume for each GPRS session.

Here we need to answer the questions of:

1. Why the revenue leakage issue has occurred?

2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The developed system prototype automatically builds a data workflow on the Neo4j graph database for the whole data match process, representing all data nodes included in the process, associated nodes and SQL processing nodes. Each SQL processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters.

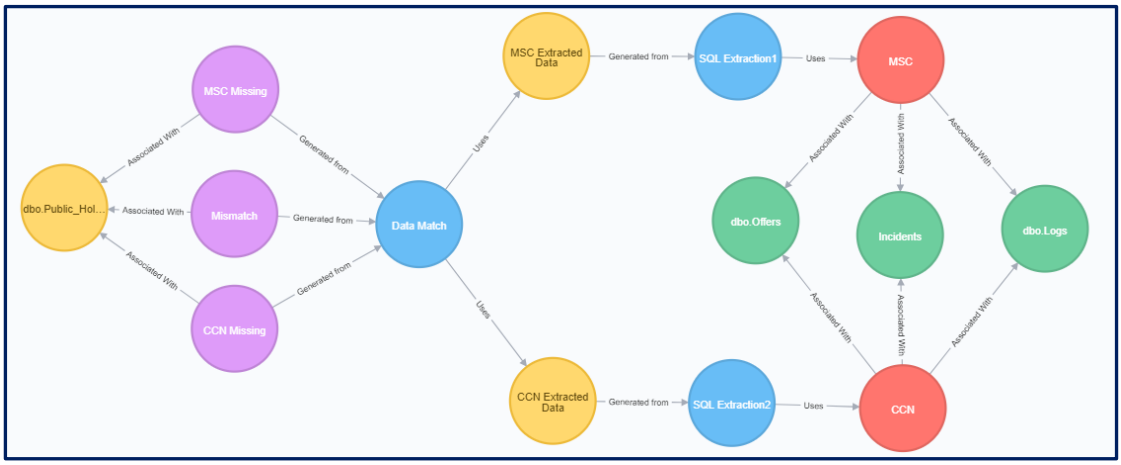


Figure 42 Provenance graph for MSC vs. CCN audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from CCN side caused by a network disconnection. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

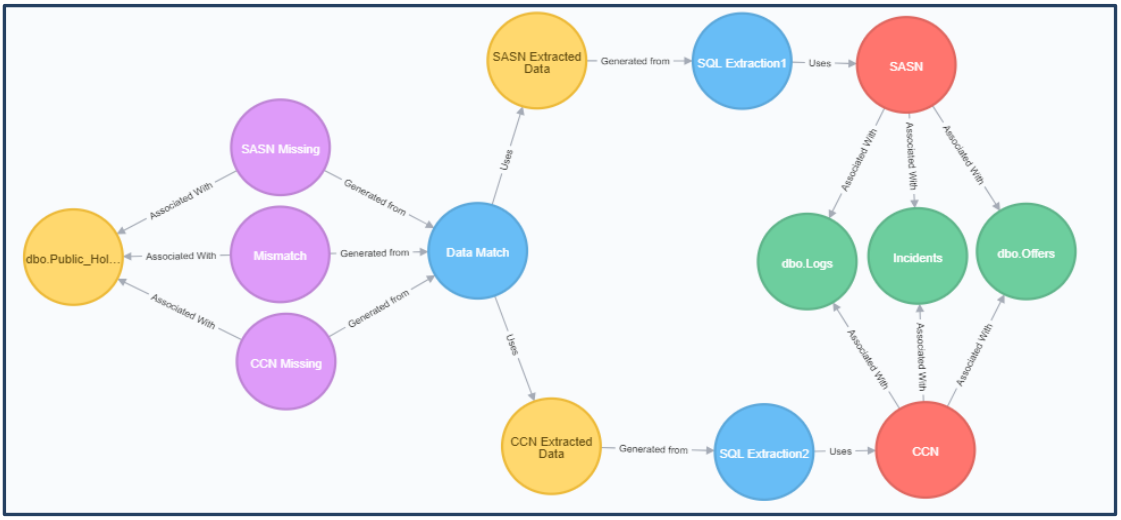


Figure 43 Provenance graph for SASN vs. CCN audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from CCN side caused by a network disconnection. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

The provenance graphs shown in Figures 42-43 describe the trace of the fourth scenario for the two audits of our model built using the Neo4j database, with blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node. The three missing and mismatch resulted data entities at the left of the graphs are connected to the public holidays node and were produced by the data match Query Execution, that was executed over the data generated by the previous Data Extraction processes containing the original SQL queries, which are connected to the data source entities that are associated to the

contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario for MSC vs. CCN audit are extracted by two SQL queries with certain filters MSC and CCN tables for the date of leakage issue. At each step of execution the generated data is stored in a separate table to support drill down capability. After the queries are executed provenance information are generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provision entities. Nodes are automatically being added to the workflow and provenance semantic information are inserted into the SQL processing nodes, then the data extracted are used as an input for the data match query that compare the two extracted results based on the calling party, called party, and the time of call to generate the missing and mismatch reports. Backward tracing is executed if high missing or mismatch is detected on a certain date by retrieving all SQL processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities using the processing nodes to detect all affected rows and find possible root-causes for the leakage issue. The same process is done for SASN vs. CCN audit.

In the given scenario, the CCN disconnection occurs on 5th-Dec starting at 9 PM. To back trace this leakage issue, the mapped attributes between the mismatch results table and the data match query are used to select the affected data in the drill down table, the filter and mapped attributes between data match table and the data extraction query tables are used to select the affected data from the drill down tables of MSC vs. CCN, and SASN vs. CCN, then use the filter stored between incidents, logs, and offers tables to enrich the data obtained about leakage issue by finding that a power failure incident in the Incidents table occurred on 5th-Dec, that caused the revenue loss.

The results answers the previously presented questions as:

1. Why the revenue leakage issue has occurred? **A power failure incident.**
2. When the revenue leakage issue has been introduced? **5th-Dec**.
3. Where the revenue leakage issue has occurred? At what node? **CCN**.
4. What has caused this revenue leakage issue to occur? **A power failure incident caused CCN disconnection**.

5.2.2.1.3 Fifth use case scenario Problem in the mediation system (Missing from Billing)

In this scenario, a great revenue leakage has occurred due to a problem in the mediation system on 10th-Dec. This problem caused the calls, SMSs, and GPRS sessions records not to be sent to the billing table. In this scenario the MSC, SMSC, and SASN records are being generated and the transactions do actually happen. But since they were not sent to the billing table, these transactions will be free with no billing records generated, causing a great revenue loss unless the problem is triggered and solved. This scenario has been used as an example of real revenue leakage problem that needs to be detected, analysed to find the root-cause of leakage and get solved. Figures 44-46 show the gap of count of transactions per traffic type for all nodes compared to Billing on the date of the mediation system problem on 10th-Dec.

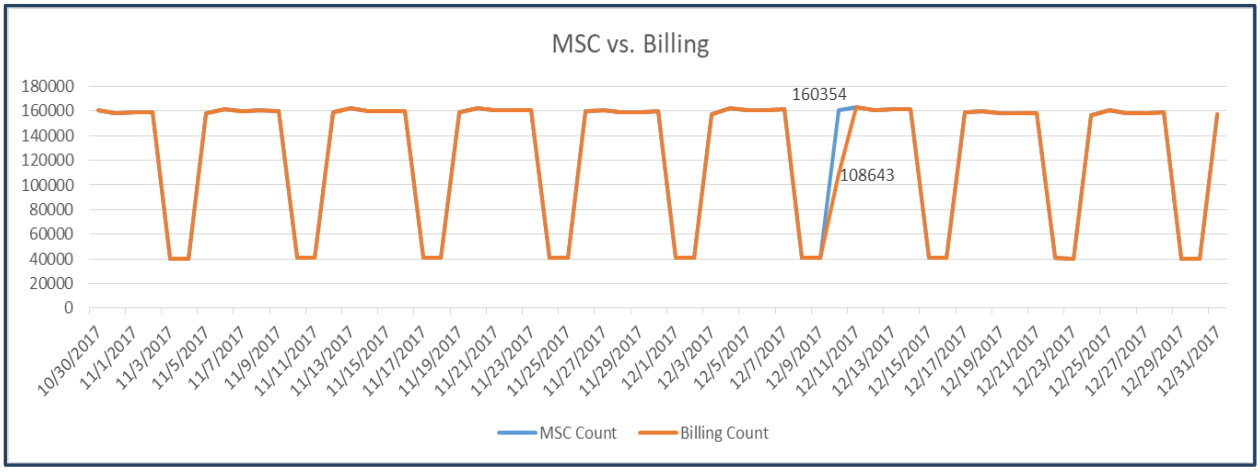


Figure 44 MSC vs. Billing Count of records

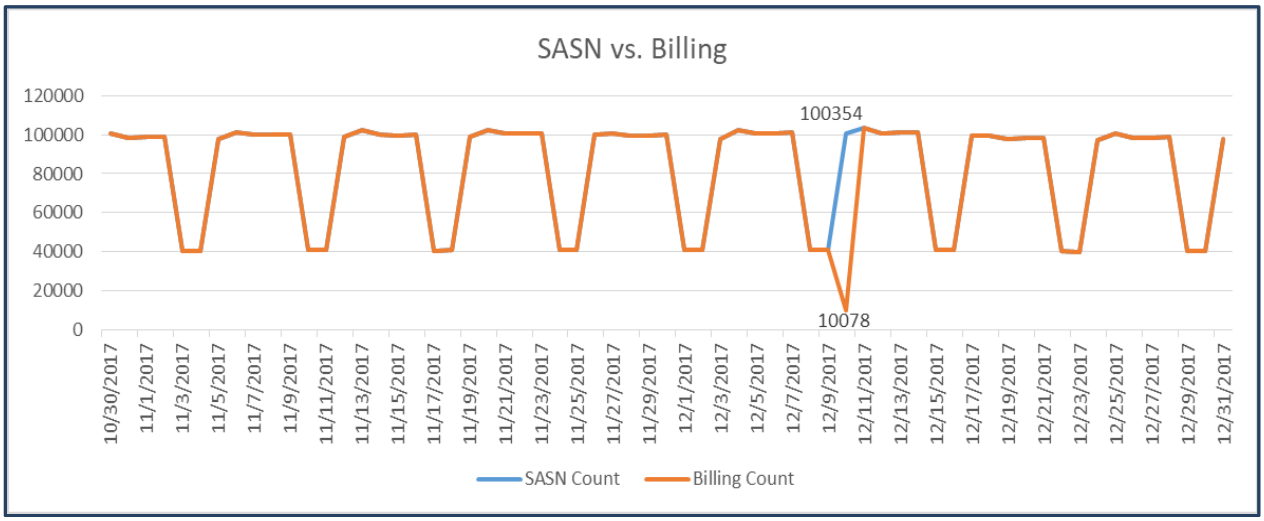


Figure 45 SASN vs. Billing Count of records

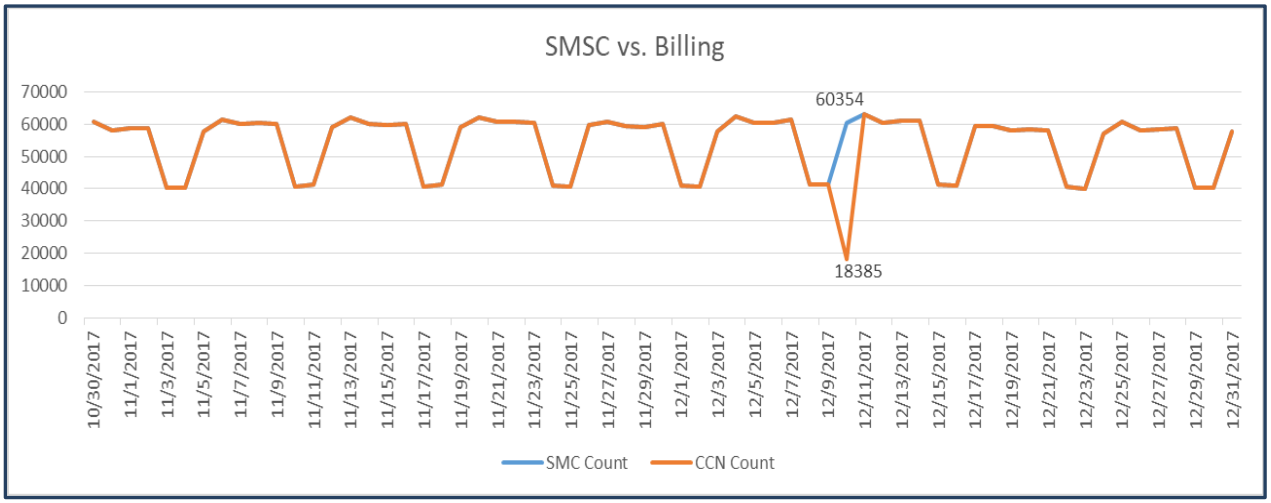


Figure 46 SMSC vs. Billing Count of records

The main audits responsible for the detection of such revenue leakage issue, extracts all postpaid voice calls from MSC side and extracts all voice calls from Billing

side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted reports in terms of calling party, called party, call time, and duration to identify missing records from Billing side (Uncharged calls), missing calls from MSC side (Charged calls that did not happen), Mismatched calls between MSC and Billing (Overcharged or Undercharged calls). The second audit extracts all postpaid SMS records from SMSC side and extracts all SMS records from Billing side for a certain date, does a count comparison between the extracted records to identify missing records from Billing side (Uncharged SMSs), missing records from SMSC side (Charged SMSs that did not happen). And the final audit extracts all postpaid GPRS records from SASN side and extracts all GPRS records from Billing side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted to identify missing records from Billing side (Uncharged GPRS sessions), missing records from SASN side (Charged sessions that did not happen), Mismatched GPRS sessions between SASN and Billing (Overcharged or Undercharged sessions).

Here we need to answer the questions of:

1. Why the revenue leakage issue has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The developed system prototype automatically builds a data workflow on the Neo4j graph database for the whole data match process, representing all data nodes included in the process, associated nodes and SQL processing nodes. Each SQL processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters.

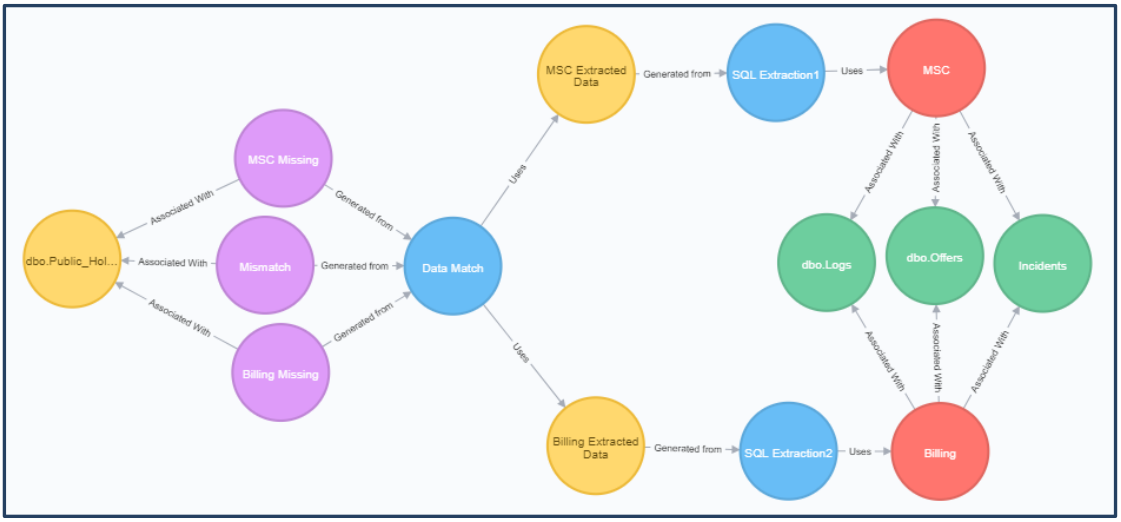


Figure 47 Provenance graph for MSC vs. Billing audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from Billing side caused by a problem in the mediation system. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

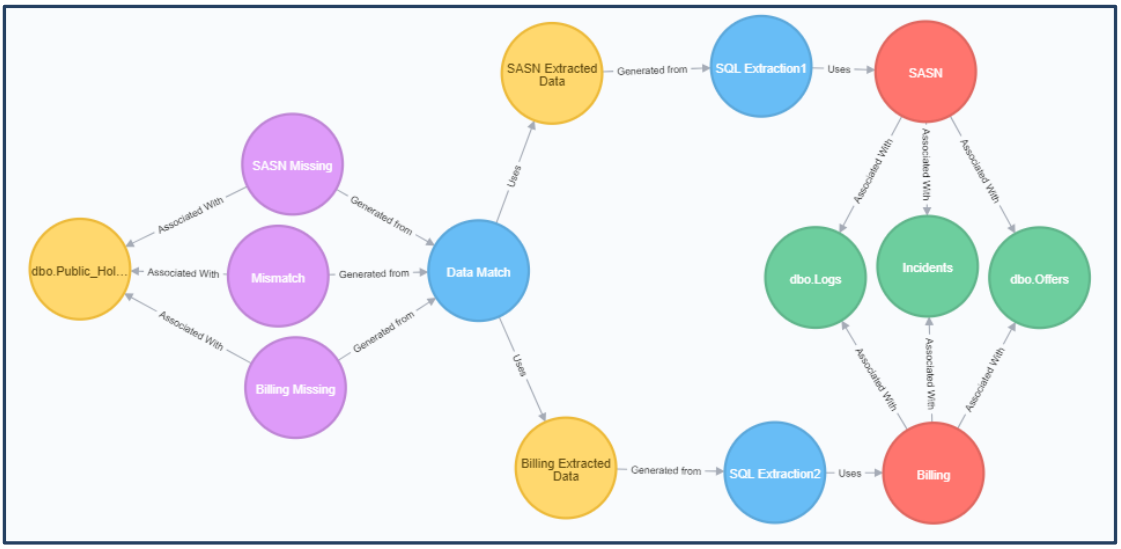


Figure 48 Provenance graph for SASN vs. Billing audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from Billing side caused by a problem in the mediation system. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision

using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

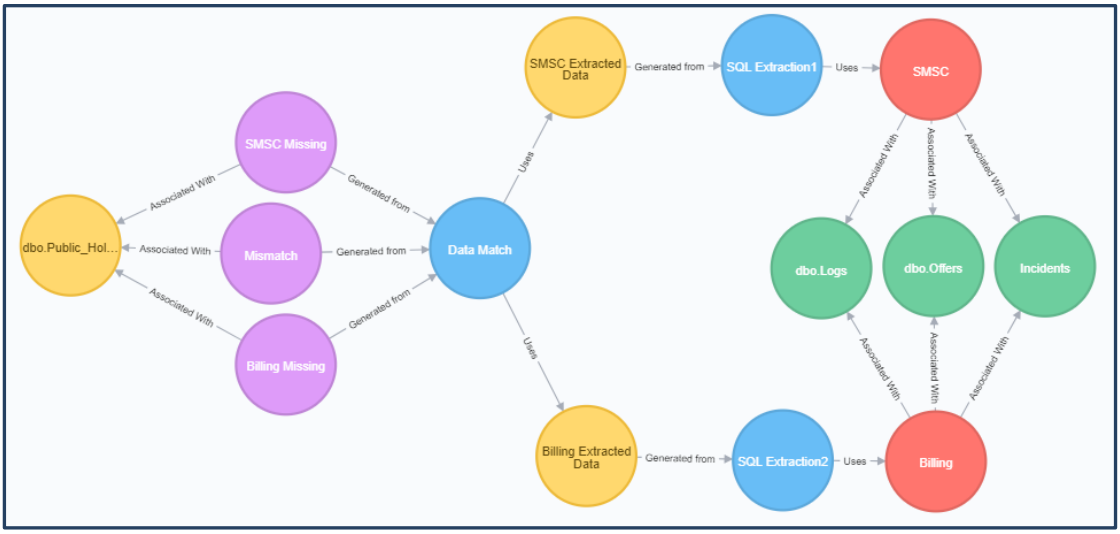


Figure 49 Provenance graph for SMSC vs. Billing audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from Billing side caused by a problem in the mediation system. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

The provenance graphs shown in Figures 47-49 describe the trace of the fifth scenario for the three audits of our model built using the Neo4j database, with blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node. The three missing and mismatch resulted data entities at the left of the graphs are connected to the public holiday node and were produced by the data match Query Execution, that was executed

over the data generated by the previous Data Extraction processes containing the original SQL queries, which are connected to the data source entities that are associated to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario are extracted by two SQL queries with certain filters MSC and Billing tables for the date of leakage issue. At each step of execution the generated data is stored in a separate table to support drill down capability. After the queries are executed provenance information are generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provision entities. Nodes are automatically being added to the workflow and provenance semantic information are inserted into the SQL processing nodes, then the data extracted are used as an input for the data match query that compare the two extracted results based on the calling party, called party, and the time of call to generate the missing and mismatch reports. Backward tracing is executed if high missing or mismatch is detected on a certain date by retrieving all SQL processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities using the processing nodes to detected all affected rows and find possible root-causes for the leakage issue. And the same is done for SMSC vs. Billing audit and SASN vs. Billing audit.

In the given scenario a problem in the mediation system occurs on 10^{th}-Dec starting at 7 PM. To back trace this leakage issue, the mapped attributes between the missing from Billing results table and the data match query are used to select the affected data in the drill down table, the filter and mapped attributes between data match table and the data extraction query tables are used to select the affected data from the drill down tables of MSC vs. Billing, SMSC vs. Billing, and SASN vs. Billing, then use

the filter stored between incidents, logs, and offers tables to enrich the data obtained about leakage issue by finding that a problem in the mediation system in the logs table occurred on 10th-Dec, that caused the revenue loss, which will be recovered in this case.

The results answers the previously presented questions as:

1. Why the revenue leakage issue has occurred? **A problem in the mediation system.**
2. When the revenue leakage issue has been introduced? **10th-Dec**.
3. Where the revenue leakage issue has occurred? At what node? **Billing**.
4. What has caused this revenue leakage issue to occur? **A problem in the mediation system.**

5.2.2.1.4 Sixth use case scenario Error in CDRs generation (Drop in Billing)

This scenario results in six sub-scenarios, as the error in CDRs generation in one of the telecom nodes responsible for the traffic generation (MSC, SASN, and SMSC), resulting in great missing in MSC compared to CCN, great missing in SMSC compared to CCN, and great missing in SASN compared to CCN for prepaid traffic. The transactions actually occur and are charged but there is a problem in CDRs generation. It also results in great revenue losses in postpaid traffic as the Billing systems generates bills according to received CDRs from the telecom nodes (MSC, SASN, and SMSC). Therefore, we'll notice great drop in MSC trends for postpaid subscribers, SMSC trends for postpaid subscribers, SASN trends for postpaid subscribers, Billing trends for voice calls, Billing trends for short messages, and Billing trends for GPRS sessions, without missing records in main postpaid audits (MSC vs. Billing, SMSC vs. Billing, and SASN vs. Billing). So, this problem causes quality and financial issues.

There are three prepaid audits responsible for the detection of quality issues related to the missing records from network main nodes compared to CCN, three measures responsible for the detection of the great drop in postpaid traffic in main nodes and three measures responsible for the detection of revenue leakage due to the great drop in telecom traffic in billing system. Below are the main audits and measures:

- Missing from MSC (MSC vs. CCN audit).
- Missing from SASN (SASN vs. CCN audit)
- Missing from SMSC (SMSC vs. CCN audit)
- Count of postpaid traffic from MSC (MSC measure).
- Count of postpaid traffic from SMSC (SMSC measure).
- Count of postpaid traffic from SASN (SASN measure).
- Count of voice calls from Billing (Billing measure for voice calls).
- Count of short messages from Billing (Billing measure for short messages).
- Count of GPRS sessions from Billing (Billing measure for GPRS sessions).

5.2.2.1.4.1 Missing from MSC (MSC vs. IN CCN audit)

The main audit responsible for the detection of such a problem that may cause revenue leakage issue, extracts all prepaid voice calls from MSC side and extracts all voice calls from CCN side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted reports in terms of calling party, called party, call time, and duration to identify missing records from CCN side (Uncharged calls), missing calls from MSC side (Charged calls that may did not happen), Mismatched calls between MSC and IN CCN (Overcharged or Undercharged calls). Figures 50-51 show the gap of count and total calls duration of transactions for MSC compared to CCN respectively on the date of the configurations change 1st-Nov.

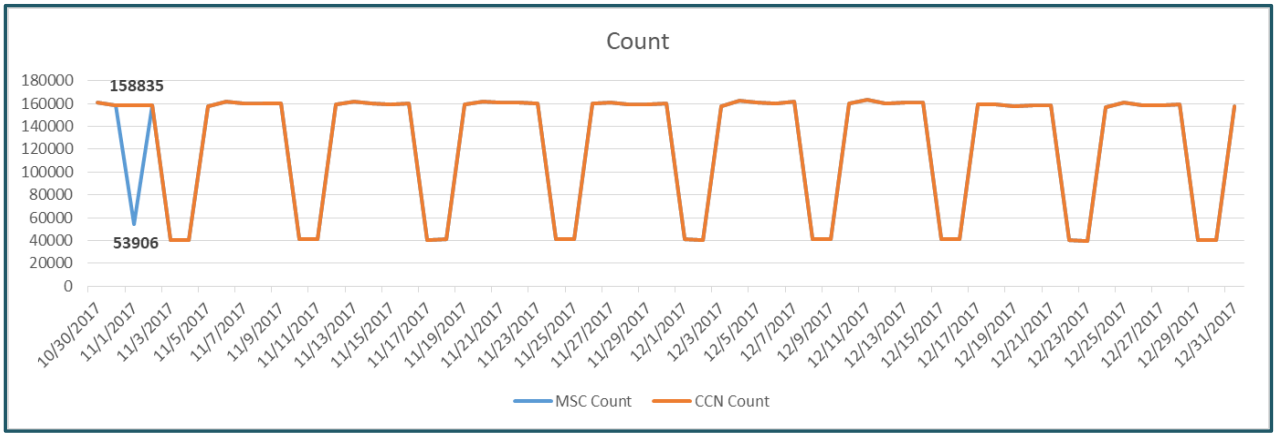


Figure 50 MSC vs. IN CCN count of records comparison

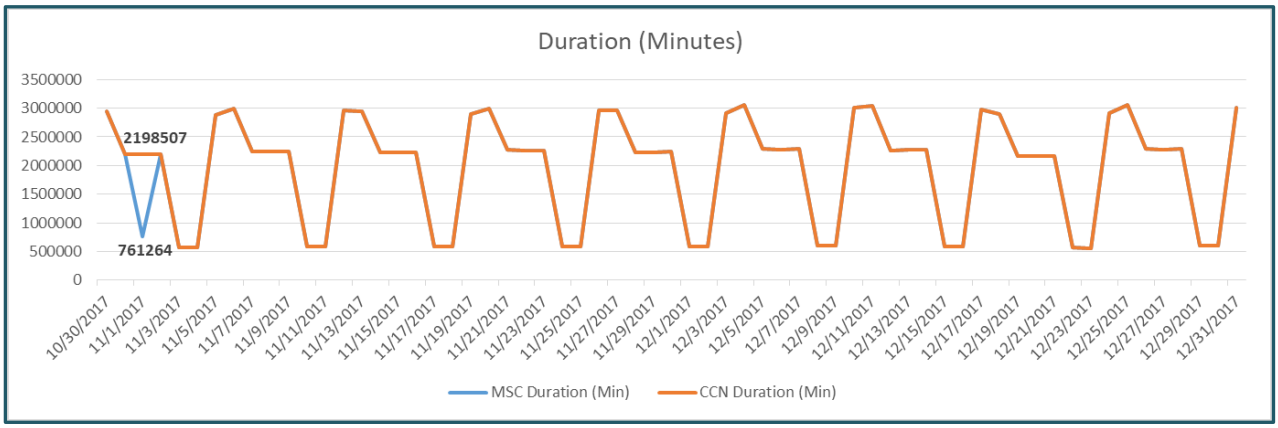


Figure 51 MSC vs. IN CCN total calls duration comparison

Here we need to answer the questions of:

1. Why the revenue leakage issue has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The developed system prototype automatically builds a data workflow on the Neo4j graph database for the whole data match process, representing all data nodes included in the process, associated nodes and SQL processing nodes. Each SQL processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in figure 52 from the Neo4j after running the sixth scenario (first sub-scenario) process for the date of interest.

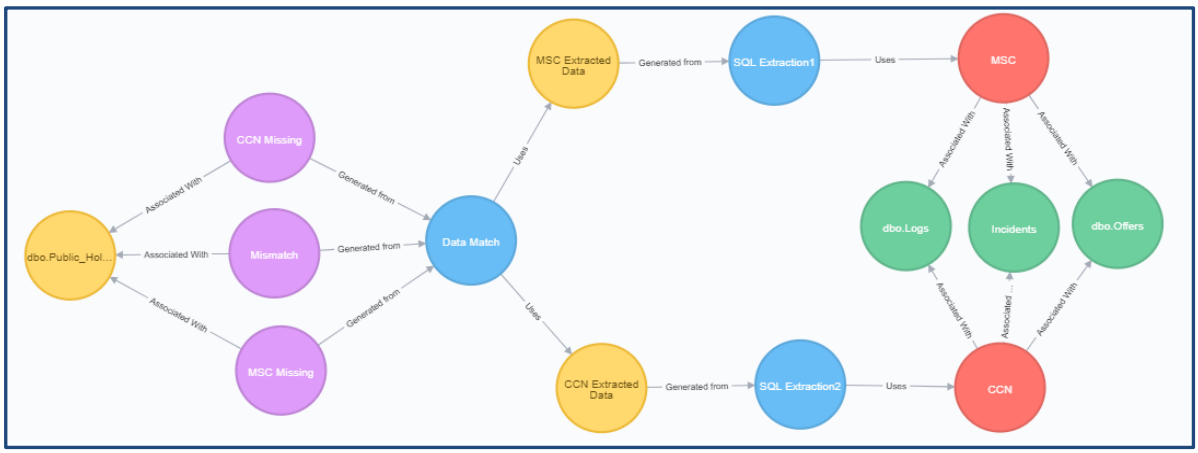


Figure 52 Provenance graph for MSC vs. CCN audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from CCN side caused by a network disconnection. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

The provenance graph shown in Figure 52 describes the trace of the sixth scenario (first sub-scenario) of our model built using the Neo4j database, with blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node. The three missing and mismatch resulted data entities at the left of the graph were produced by the data match Query Execution, that was executed over the data generated by the previous Data Extraction processes containing the original SQL queries, which are connected to the data source entities that are associated to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario are extracted by two SQL queries with certain filters MSC and CCN tables for the date of leakage issue. At each step of execution the generated data is stored in a separate table to support drill down capability. After the queries are executed provenance information are generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provision entities. Nodes are automatically being added to the workflow and provenance semantic information are inserted into the SQL processing nodes, then the data extracted are used as an input for the data match query that compare the two extracted results based on the calling party, called party, and the time of call to generate the missing and mismatch reports. Backward tracing is executed if high missing or mismatch is detected on a certain date by retrieving all SQL processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities using the processing nodes to detected all affected rows and find possible root-causes for the leakage issue.

In the given scenario the MSC configurations change occurs on 1st-Nov. To back trace this issue, the mapped attributes between the missing from MSC results table and the data match query are used to select the affected data in the drill down table, the filter and mapped attributes between data match table and the data extraction query tables are used to select the affected data from the drill down tables of MSC and CCN, then use the filter stored between incidents, logs, and offers tables to enrich the data obtained about leakage issue by finding that an MSC configurations change in the Logs table occurred on 1st-Nov, that caused the problem.

The results answers the previously presented questions as:

1. Why the revenue leakage issue has occurred? **An MSC configurations change.**
2. When the revenue leakage issue has been introduced? **1st –Nov.**

3. Where the revenue leakage issue has occurred? At what node? **MSC**.
4. What has caused this revenue leakage issue to occur? **An MSC configurations change.**

5.2.2.1.4.2 Missing from SMSC (SMSC vs. CCN audit)

The main audit responsible for the detection of such a problem that may cause revenue leakage issue, extracts all prepaid SMSs from SMSC side and extracts all SMSs from CCN side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted reports in terms of calling party, called party, and call time to identify missing records from CCN side (Uncharged SMSs), and missing SMSs from SMSC side (Charged SMSs that may did not happen). Figure 53 shows the gap of count of transactions for SMSC compared to CCN on the date of the configurations change 1st-Nov.

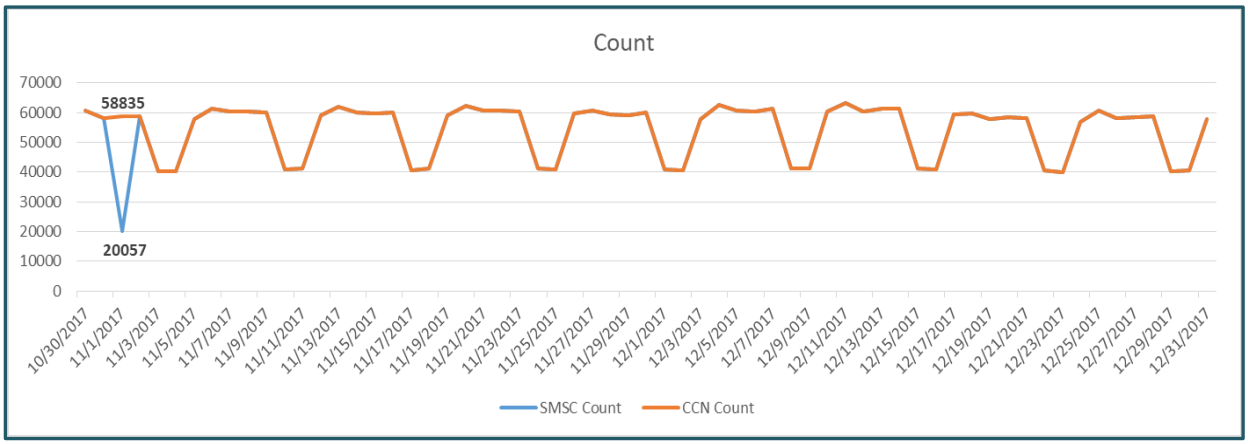


Figure 53 SMSC vs. CCN records count comparison

Here we need to answer the questions of:

1. Why the revenue leakage issue has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The developed system prototype automatically builds a data workflow on the Neo4j graph database for the whole data match process, representing all data nodes included in the process, associated nodes and SQL processing nodes. Each SQL processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 54 from the Neo4j after running the sixth scenario (second sub-scenario) process for the date of interest.

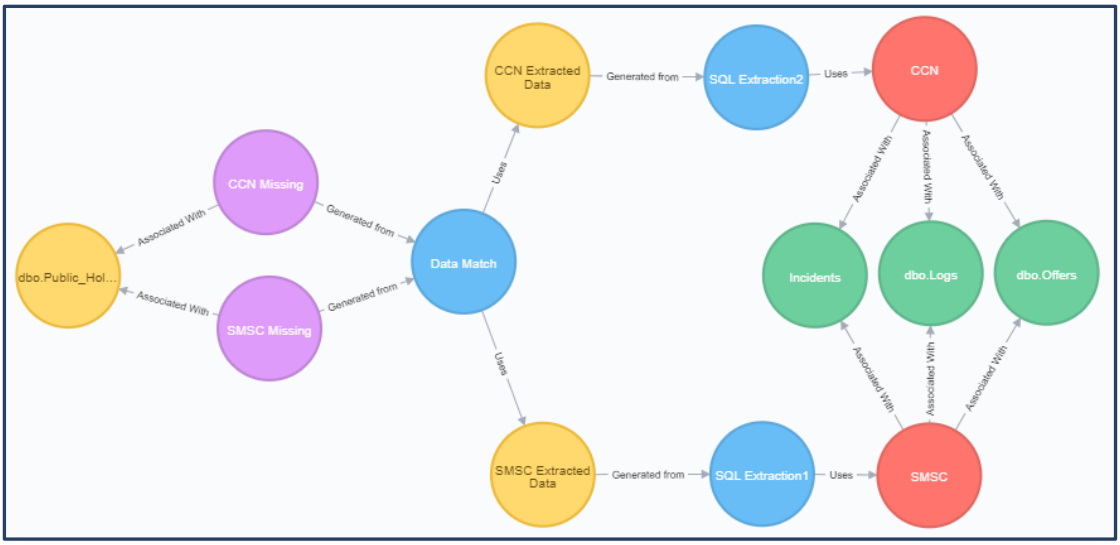


Figure 54 Provenance graph for SMSC vs. CCN audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from CCN side caused by a network disconnection. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

The provenance graph shown in Figure 54 describes the trace of the sixth scenario (second sub-scenario) of our model built using the Neo4j database, with blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source

entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node. The two missing resulted data entities at the left of the graph were produced by the data match Query Execution, that was executed over the data generated by the previous Data Extraction processes containing the original SQL queries, which are connected to the data source entities that are associated to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario are extracted by two SQL queries with certain filters SMSC and CCN tables for the date of the issue. At each step of execution the generated data is stored in a separate table to support drill down capability. After the queries are executed provenance information are generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provision entities. Nodes are automatically being added to the workflow and provenance semantic information are inserted into the SQL processing nodes, then the data extracted are used as an input for the data match query that compare the two extracted results based on the calling party, called party, and the time of call to generate the missing reports. Backward tracing is executed if high missing or mismatch is detected on a certain date by retrieving all SQL processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities using the processing nodes to detected all affected rows and find possible root-causes for the leakage issue.

In the given scenario the SMSC configurations change occurs on 1st-Nov. To back trace this issue, the mapped attributes between the missing from SMSC results table and the data match query are used to select the affected data in the drill down table, the filter and mapped attributes between data match table and the data extraction

query tables are used to select the affected data from the drill down tables of SMSC and CCN, then use the filter stored between incidents, logs, and offers tables to enrich the data obtained about leakage issue by finding that an SMSC configurations change in the Logs table occurred on 1st-Nov, that caused the problem.

The results answers the previously presented questions as:

1. Why the revenue leakage issue has occurred? **An SMSC configurations change.**
2. When the revenue leakage issue has been introduced? **1st –Nov.**
3. Where the revenue leakage issue has occurred? At what node? S**MSC**.
4. What has caused this revenue leakage issue to occur? **An SMSC configurations change.**

5.2.2.1.4.3 Missing from SASN (SASN vs. IN CCN audit)

The main audit responsible for the detection of such a problem that may cause revenue leakage issue, extracts all prepaid GPRS sessions from SASN side and extracts all GPRS sessions from CCN side for a certain date, does a count comparison between the extracted records and data match comparison between the extracted reports in terms of MSISDN, APN, transaction time, and data volume to identify missing records from CCN side (Uncharged GPRS sessions), missing records from SASN side (Charged GPRS sessions that may did not happen), Mismatched GPRS sessions between SASN and CCN (Overcharged or Undercharged GPRS sessions). Figures 55-56 show the gap of count and total data volume of GPRS transactions for SASN compared to CCN respectively on the date of the configurations change 1st-Nov.

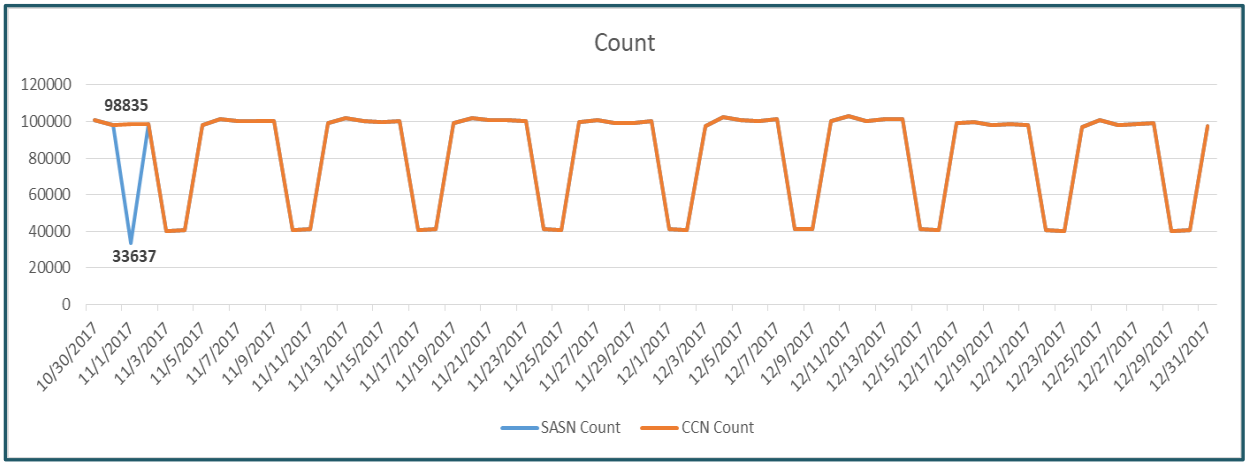


Figure 55 SASN vs. IN CCN count of records comparison

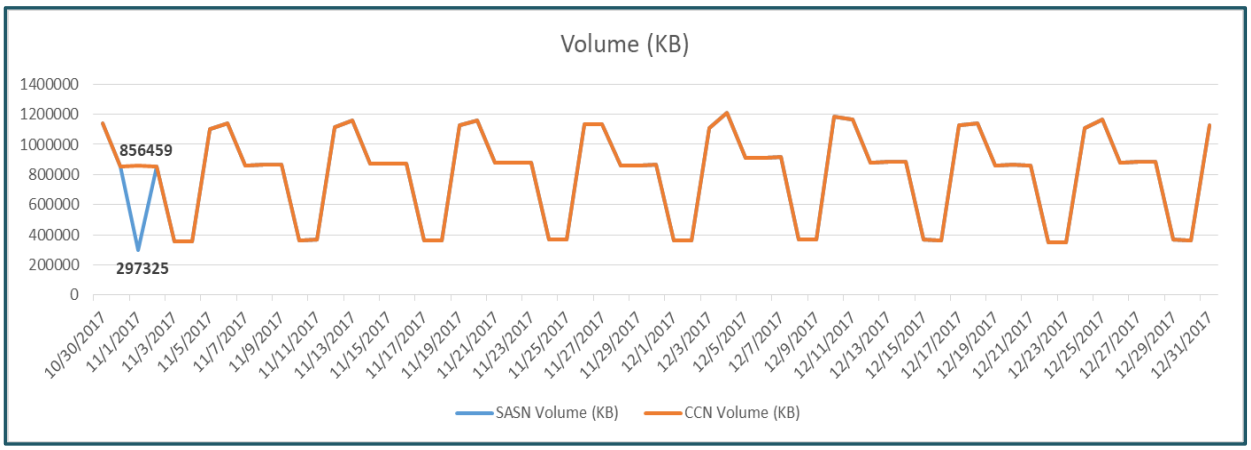


Figure 56 SASN vs. IN CCN total data volume comparison

Here we need to answer the questions of:

1. Why the revenue leakage issue has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The developed system prototype automatically builds a data workflow on the Neo4j graph database for the whole data match process, representing all data nodes included in the process, associated nodes and SQL processing nodes. Each SQL processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 57 from the Neo4j after running the sixth scenario (third sub-scenario) process for the date of interest.

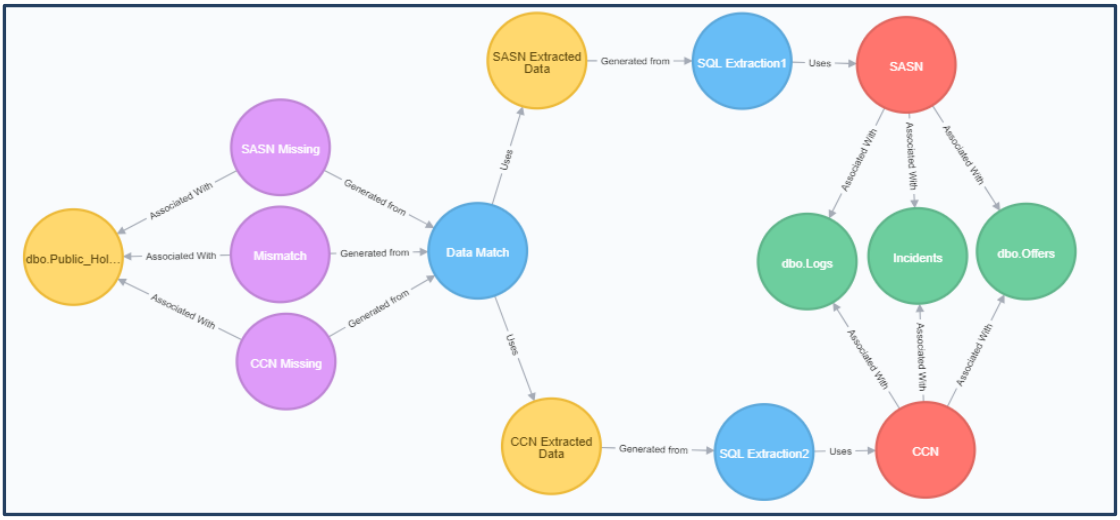


Figure 57 Provenance graph for SASN vs. CCN audit describing the execution of the data extraction, join, and data matching queries in the case of data match audit due to a great missing from CCN side caused by a network disconnection. Blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node (In this diagram, processing nodes for contextual information were represented by relations for simplicity here, but in the model they are represented as processing nodes , since they are 12 nodes)

The provenance graph shown in figure 57 describes the trace of the sixth scenario (third sub-scenario) of our model built using the Neo4j database, with blue nodes denote SQL processing nodes, Purple nodes denote resulted data items entities, green nodes denote associated entities to the data source entities for contextual information provision using SQL processing nodes, red nodes denote data source entities, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node. The three missing and mismatch resulted data entities at the left of the graph were produced by the data match Query Execution, that was executed over the data generated by the previous Data Extraction processes containing the original SQL queries, which are connected to the data source entities that are associated to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario are extracted by two SQL queries with certain filters SASN and CCN tables for the date of the issue. At each step of execution the generated data is stored in a separate table to support drill down capability. After the queries are executed provenance information are generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provision entities. Nodes are automatically being added to the workflow and provenance semantic information are inserted into the SQL processing nodes, then the data extracted are used as an input for the data match query that compare the two extracted results based on the calling party, called party, and the time of transaction to generate the missing and mismatch reports. Backward tracing is executed if high missing or mismatch is detected on a certain date by retrieving all SQL processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities using the processing nodes to detected all affected rows and find possible root-causes for the leakage issue.

In the given scenario the SASN configurations change occurs on 1st-Nov. To back trace this issue, the mapped attributes between the missing from SASN results table and the data match query are used to select the affected data in the drill down table, the filter and mapped attributes between data match table and the data extraction query tables are used to select the affected data from the drill down tables of SASN and CCN, then use the filter stored between incidents, logs, and offers tables to enrich the data obtained about leakage issue by finding that a SASN configurations change in the Logs table occurred on 1st-Nov, that caused the problem.

The results answers the previously presented questions as:

1. Why the revenue leakage issue has occurred? **A SASN configurations change.**
2. When the revenue leakage issue has been introduced? **1st –Nov.**

3. Where the revenue leakage issue has occurred? At what node? **SASN**.
4. What has caused this revenue leakage issue to occur? **A SASN configurations change.**

5.2.2.1.4.4 Count of postpaid traffic from MSC (MSC measure)

In this scenario, a major drop in postpaid voice calls traffic in terms of records count, and calls duration is recorded on 1st-Nov. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since the root-cause behind the traffic decrease must be identified. Figures 58-59 show the usage abnormal behavior, since the postpaid voice calls per day trends follow a specific pattern in terms of count, and duration. But in this scenario, we notice that there are much count, and duration decrease on 1st –Nov, and this behavior needs to be investigated to identify its root-cause and possible effects.

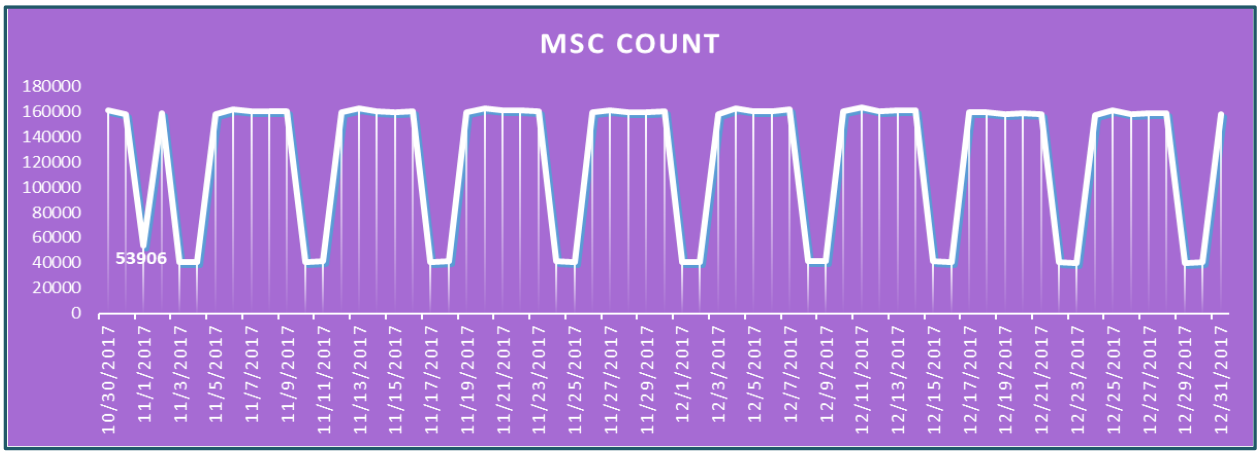


Figure 58 Count of postpaid voice calls

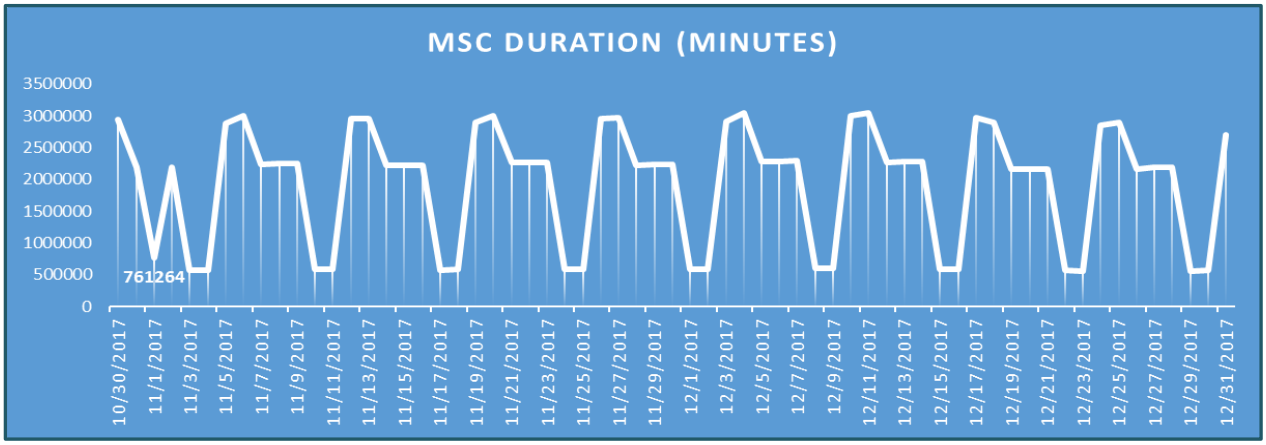


Figure 59 Total duration of postpaid voice calls

Here we need to answer the questions of:

1. Why the traffic increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for postpaid usage traffic monitoring and trending performs a SQL query with a filter to extract only voice mobile originating traffic for postpaid subscribers towards all destinations then it groups them per destination type and per date afterwards in terms of calls count, and total calls duration.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 60 from the Neo4j after running the usage monitoring process for the date 1st-Nov.

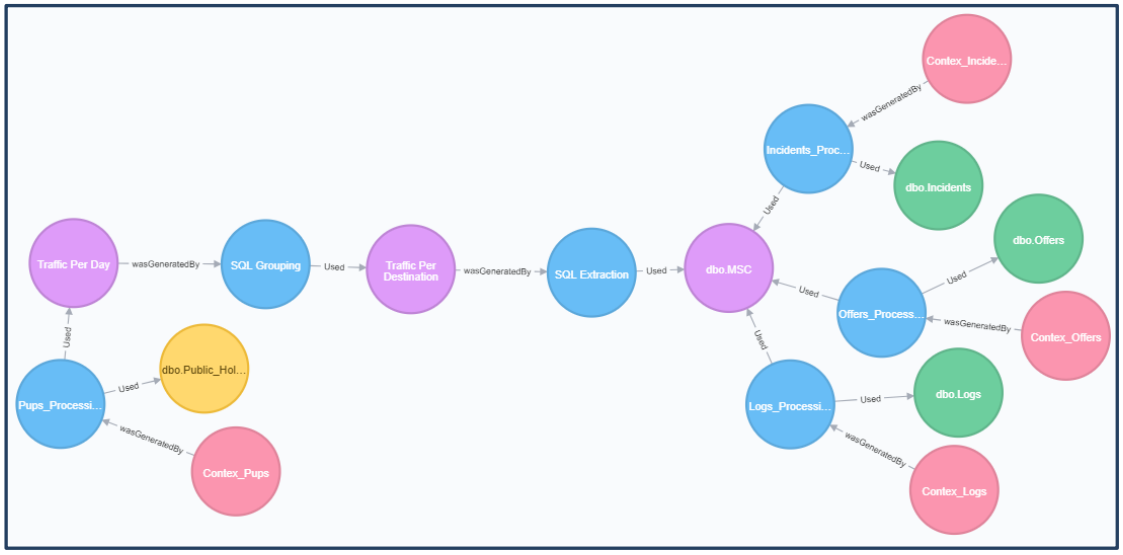


Figure 60 Provenance graph describing the execution of the data extraction query in the abnormal postpaid voice calls traffic decrease use case over MSC. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 60 describes the trace of the sixth scenario (Sixth sub-scenario) of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple node as a data source entity, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Summary_Per_Day Query Result entity at the left of the graph connected to public holidays node and was produced by the Traffic_Summary_Per_Destination Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario will be extracted by a SQL query with a certain filter over MSC table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance

information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs on 1st -Nov. To back trace this value, the public holidays matched on the same day with the results table are mapped and the mapped attributes between the results table and the Traffic summary per destination are used to select the affected data in the drill down table, the filter and mapped attributes between duration per destination table and the MSC table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the traffic drop occurred due to an MSC configurations change. And after the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue through back till the logs node, noting that an MSC configurations change has

occurred on the same day of the abnormal behavior and may have caused the great traffic decrease.

The results answers the previously presented questions as:

1. Why the duration decrease has occurred? **An MSC configurations change.**
2. When the revenue leakage issue has been introduced? **1st -Nov.**
3. Where the revenue leakage issue has occurred? At what node? **MSC**.
4. What has caused this revenue leakage issue to occur? **An MSC configurations change.**

5.2.2.1.4.5 Count of postpaid traffic from SMSC (SMSC measure)

In this scenario, a major drop in postpaid SMS traffic in terms of records count is recorded on 1st-Nov. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since the root-cause behind the traffic decrease must be identified. Figure 61 show the usage abnormal behavior, since postpaid SMSs per day trend follows a specific pattern in terms of count. But in this scenario, we notice that there are much count decrease on 1st -Nov, and this behavior needs to be investigated to identify its root-cause and possible effects.

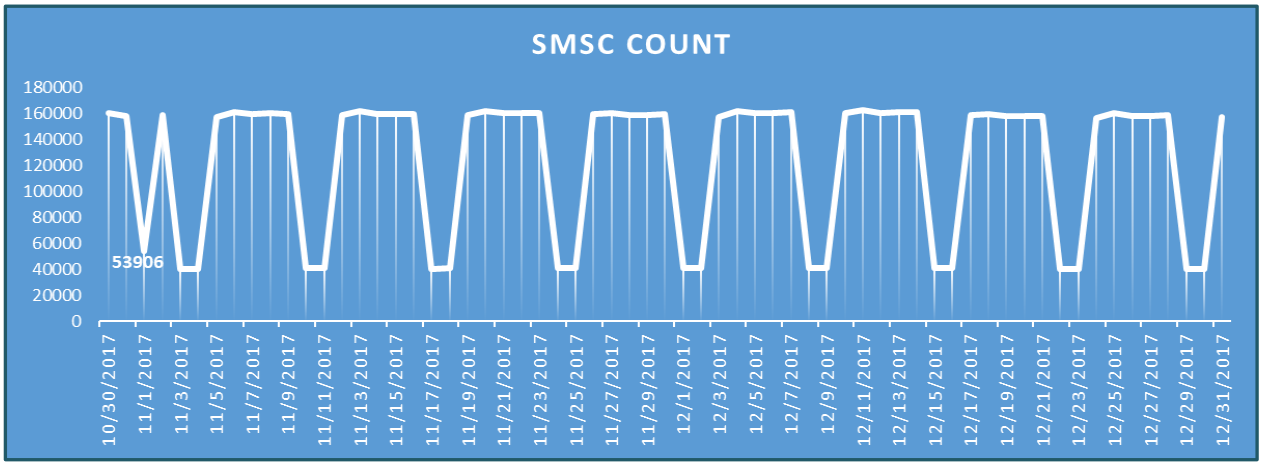


Figure 61 Count of postpaid SMS records from SMSC side

Here we need to answer the questions of:

1. Why the traffic increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for postpaid usage traffic monitoring and trending performs a SQL query with a filter to extract only short messages traffic for postpaid subscribers towards all destinations then it groups them per destination type and per date afterwards in terms of transactions count.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 62 from the Neo4j after running the usage monitoring process for the date 1st-Nov.

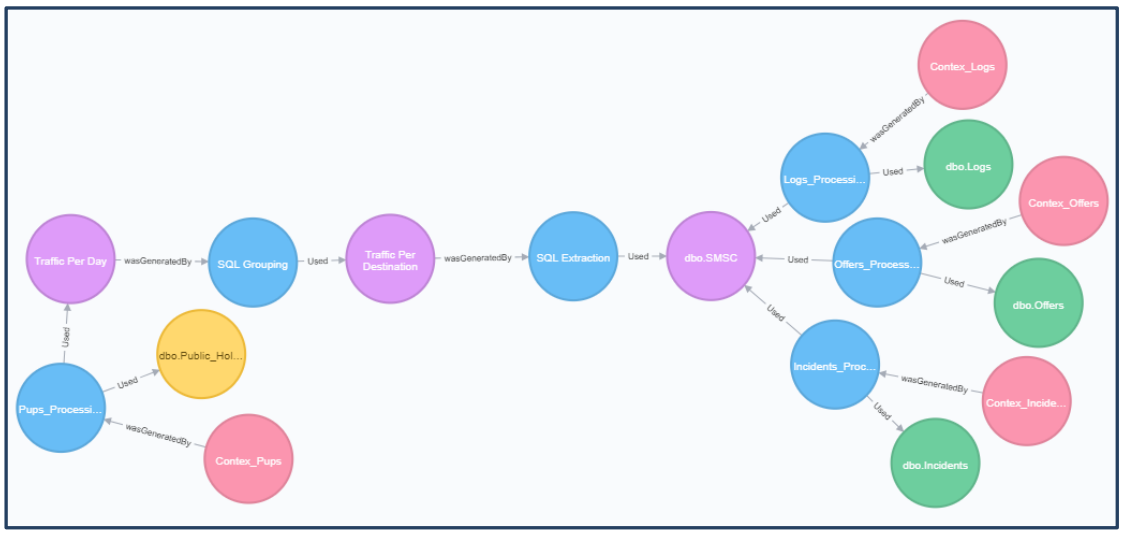


Figure 62 Provenance graph describing the execution of the data extraction query in the abnormal postpaid SMS traffic decrease use case over SMSC. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for

contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 62 describes the trace of the sixth scenario (Sixth sub-scenario) of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple node as a data source entity, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Summary_Per_Day Query Result entity at the left of the graph connected to public holidays node and was produced by the Traffic_Summary_Per_Destination Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario will be extracted by a SQL query with a certain filter over SMSC table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs on 1st -Nov. To back trace this value, the public holidays matched on the same day with the results table are mapped

and the mapped attributes between the results table and the Traffic summary per destination are used to select the affected data in the drill down table, the filter and mapped attributes between Traffic per destination table and the SMSC table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the traffic drop occurred due to an SMSC configurations change. And after the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue through back till the logs node, noting that an SMSC configurations change has occurred on the same day of the abnormal behavior and may have caused the great traffic decrease.

The results answers the previously presented questions as:

1. Why the duration decrease has occurred? **An SMSC configurations change.**
2. When the revenue leakage issue has been introduced? **1st -Nov.**
3. Where the revenue leakage issue has occurred? At what node? **SMSC**.
4. What has caused this revenue leakage issue to occur? **An SMSC configurations change.**

5.2.2.1.4.6 Count of postpaid traffic from SASN (SASN measure)

In this scenario, a major drop in postpaid GPRS sessions traffic in terms of records count, and data volume is recorded on 1st-Nov. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since the root-cause behind the traffic decrease must be identified. Figures 63-64 show the usage abnormal behavior, since the postpaid GPRS sessions per day trends follow a specific pattern in terms of count, and volume. But in this scenario, we notice that there are much count, and volume decrease on 1st – Nov, and this behavior needs to be investigated to identify its root-cause and possible effects.

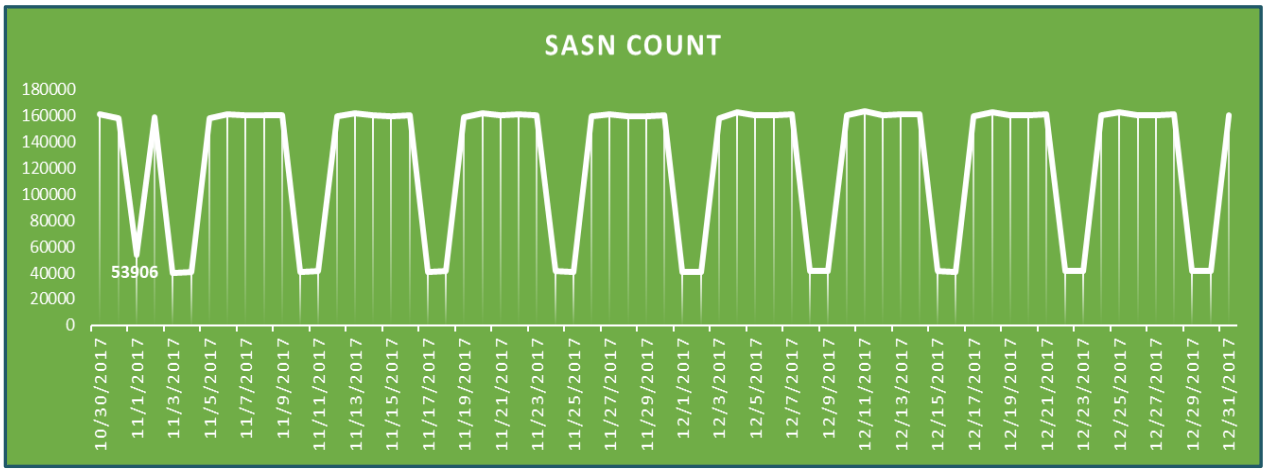


Figure 63 Count of postpaid GPRS records from SASN side

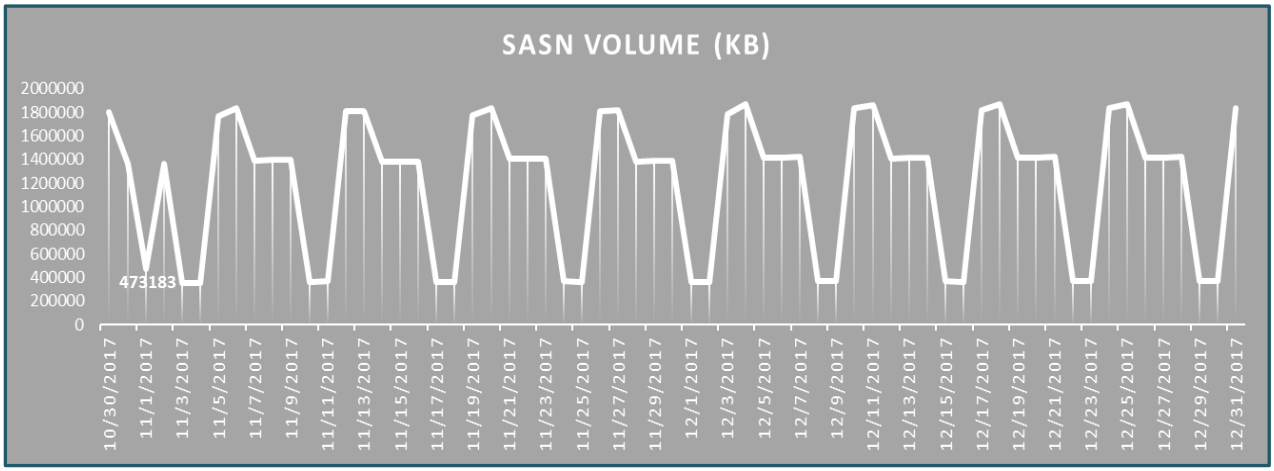


Figure 64 Total data volume for postpaid GPRS traffic from SASN side

Here we need to answer the questions of:

1. Why the traffic increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for postpaid usage traffic monitoring and trending performs a SQL query with a filter to extract only GPRS sessions traffic for postpaid subscribers towards all APNs then it groups them per APN and per date afterwards in terms of transactions count, and total data volume.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 65 from the Neo4j after running the usage monitoring process for the date 1st-Nov.

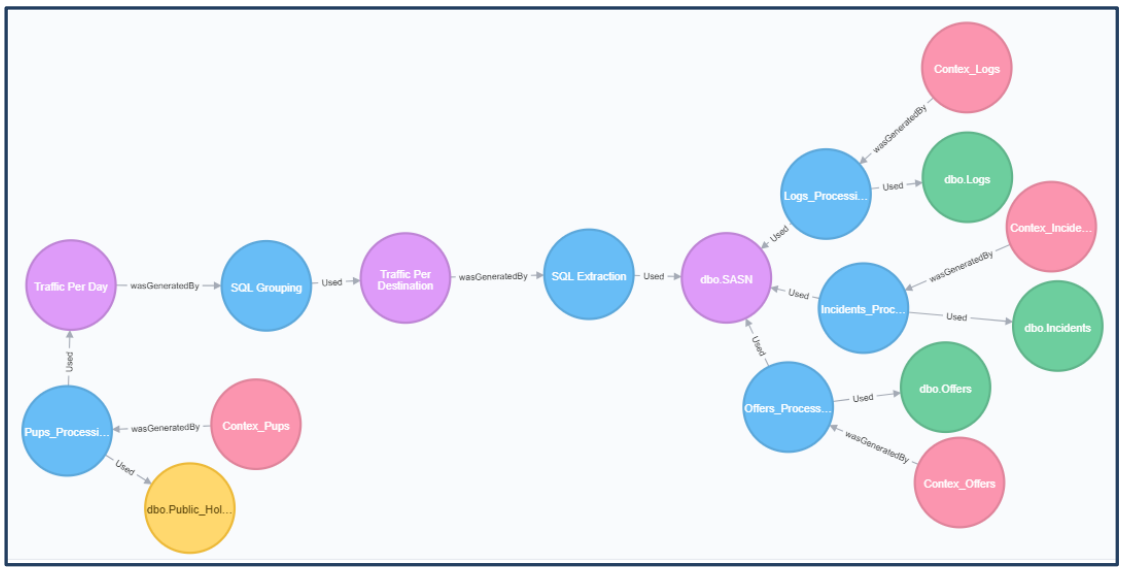


Figure 65 Provenance graph describing the execution of the data extraction query in the abnormal postpaid GPRS traffic decrease use case over SASN. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for

contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 65 describes the trace of the sixth scenario (Sixth sub-scenario) of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple node as a data source entity, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Summary_Per_Day Query Result entity at the left of the graph connected to public holidays node and was produced by the Traffic_Summary_Per_APN Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario will be extracted by a SQL query with a certain filter over SASN table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs on 1st -Nov. To back trace this value, the public holidays matched on the same day with the results table are mapped

and the mapped attributes between the results table and the Traffic summary per APN are used to select the affected data in the drill down table, the filter and mapped attributes between volume per APN table and the SASN table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the traffic drop occurred due to a SASN configurations change. And after the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue through back till the logs node, noting that a SASN configurations change has occurred on the same day of the abnormal behavior and may have caused the great traffic decrease.

The results answers the previously presented questions as:

1. Why the duration decrease has occurred? **A SASN configurations change.**
2. When the revenue leakage issue has been introduced? **1st -Nov.**
3. Where the revenue leakage issue has occurred? At what node? **SASN**.
4. What has caused this revenue leakage issue to occur? **A SASN configurations change.**

5.2.2.1.4.7 Count of voice calls from Billing (Billing measure for voice calls).

In this scenario, a major drop in postpaid voice calls traffic in terms of records count, calls duration, and calls charge is recorded on 1st-Nov. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since the root-cause behind the traffic decrease must be identified. Figures 66-68 show the usage abnormal behavior, since the postpaid voice calls per day trends follow a specific pattern in terms of count, duration, and charge. But in this scenario, we notice that there are much count, duration, and charge decrease on 1st –Nov, and this behavior needs to be investigated to identify its root-cause and possible effects.

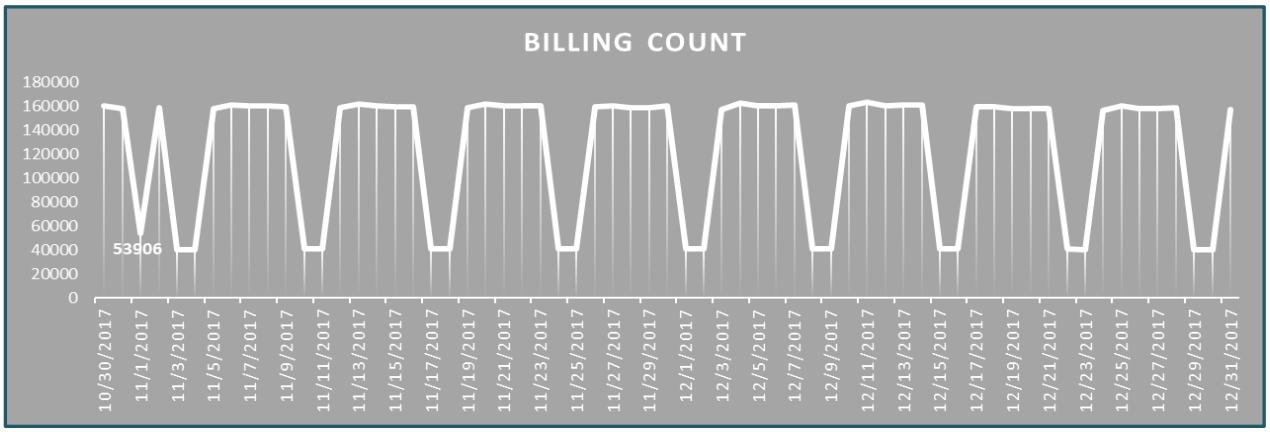


Figure 66 Count of postpaid voice calls from Billing side

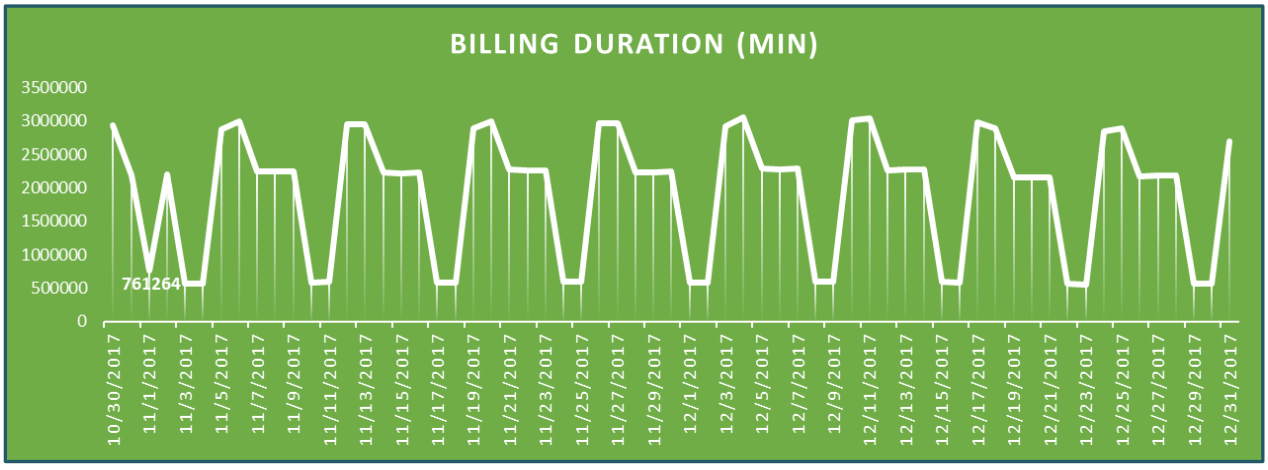


Figure 67 Total duration of postpaid voice calls from Billing side

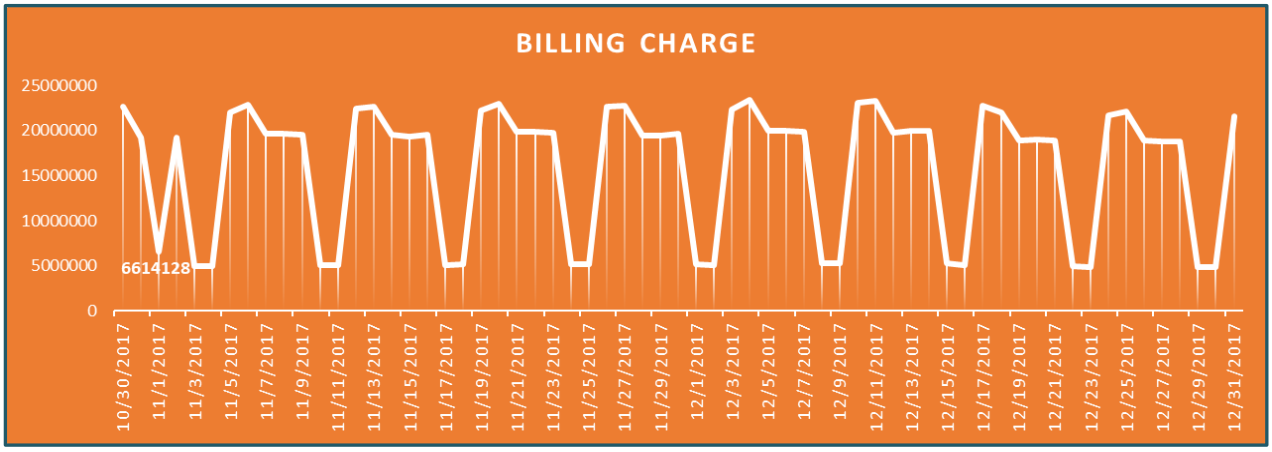


Figure 68 Total charge of postpaid voice calls from Billing side

Here we need to answer the questions of:

1. Why the traffic increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for postpaid usage traffic monitoring and trending performs a SQL query with a filter to extract only voice mobile originating traffic for postpaid subscribers towards all destinations then it groups them per destination type and per date afterwards in terms of calls count, total calls duration, and total calls charge.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 69 from the Neo4j after running the usage monitoring process for the date 1st-Nov.

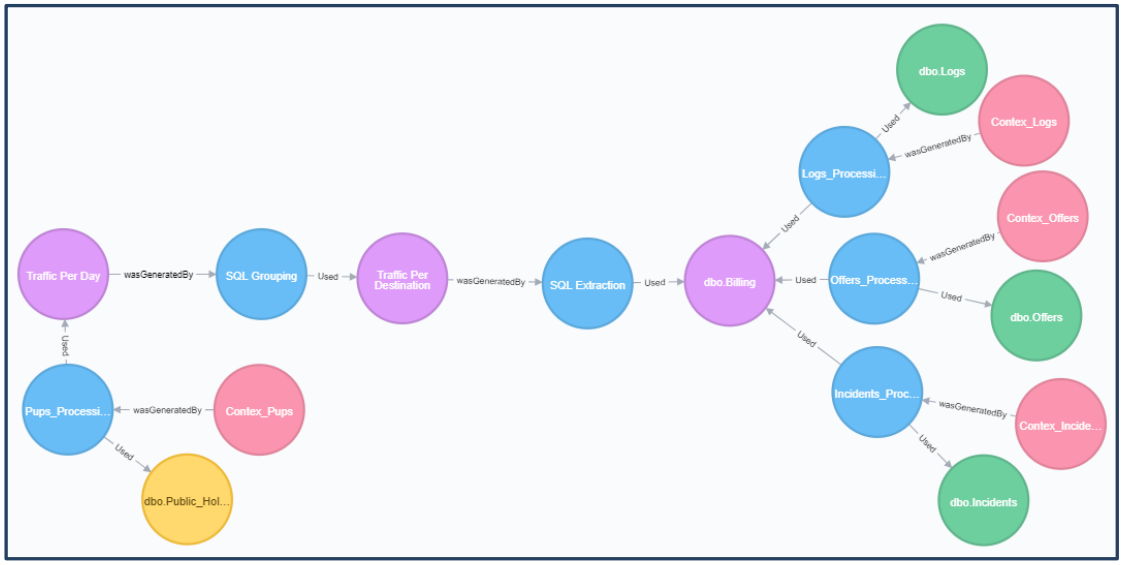


Figure 69 Provenance graph describing the execution of the data extraction query in the abnormal postpaid voice calls traffic decrease use case over Billing. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denoteassociated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 69 describes the trace of the sixth scenario (Sixth sub-scenario) of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple node as a data source entity, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Summary_Per_Day Query Result entity at the left of the graph connected to public holidays node and was produced by the Traffic_Summary_Per_Destination Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario will be extracted by a SQL query with a certain filter over Billing table. At each step of execution the generated data is stored

in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs on 1st -Nov. To back trace this value, the public holidays matched on the same day with the results table are mapped and the mapped attributes between the results table and the Traffic summary per destination are used to select the affected data in the drill down table, the filter and mapped attributes between Traffic per destination table and the Billing table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the traffic drop occurred due to an MSC configurations change. And after the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue through back till the logs node, noting that an MSC configurations change has

occurred on the same day of the abnormal behavior and may have caused the great traffic decrease.

The results answers the previously presented questions as:

1. Why the duration decrease has occurred? **An MSC configurations change.**
2. When the revenue leakage issue has been introduced? **1st -Nov.**
3. Where the revenue leakage issue has occurred? At what node? **MSC**.
4. What has caused this revenue leakage issue to occur? **An MSC configurations change.**

5.2.2.1.4.8 Count of short messages from Billing (Billing measure for short messages).

In this scenario, a major drop in postpaid SMS traffic in terms of records count and charge is recorded on 1st-Nov. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since the root-cause behind the traffic decrease must be identified. Figures 70-71 show the usage abnormal behavior, since postpaid SMSs per day trend follows a specific pattern in terms of count. But in this scenario, we notice that there are much count and charge decrease on 1st -Nov, and this behavior needs to be investigated to identify its root-cause and possible effects.

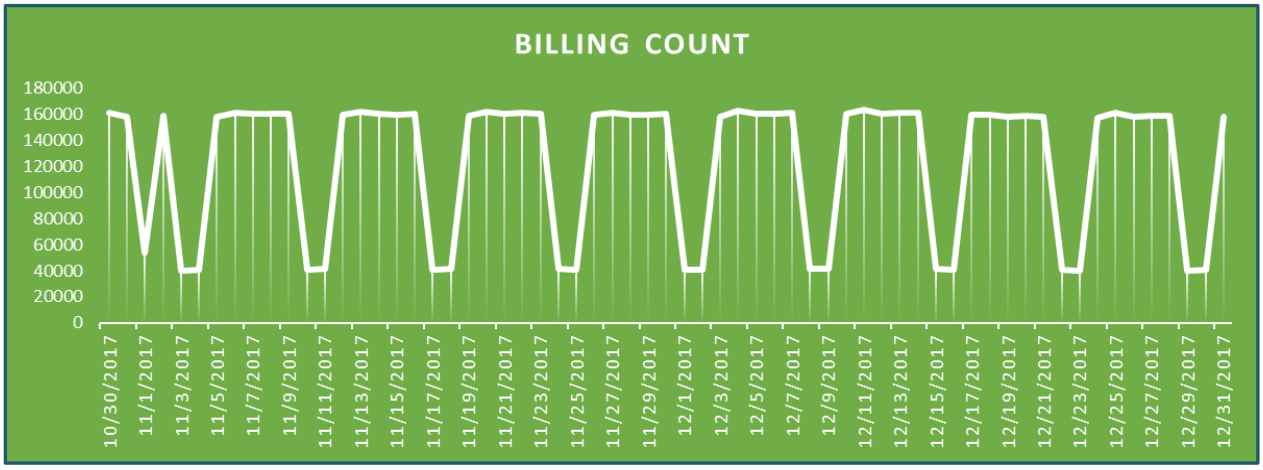


Figure 70 Count of postpaid SMS records from Billing side

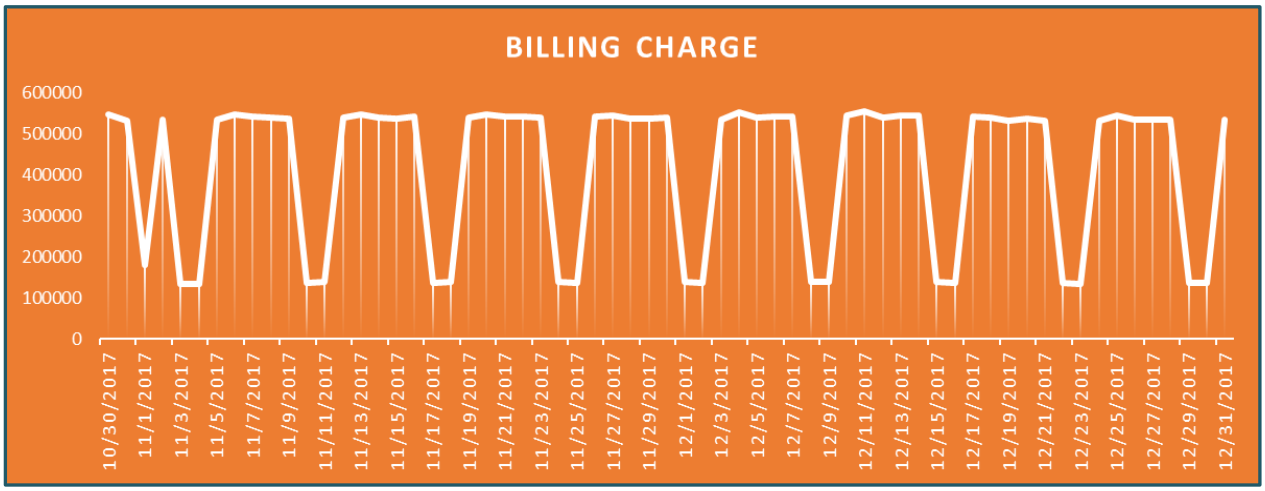


Figure 71 Total charge of postpaid SMS traffic from Billing side

Here we need to answer the questions of:

1. Why the traffic increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for postpaid usage traffic monitoring and trending performs a SQL query with a filter to extract only short messages traffic for postpaid subscribers towards all destinations then it groups them per destination type and per date afterwards in terms of transactions count.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 72 from the Neo4j after running the usage monitoring process for the date 1st-Nov.

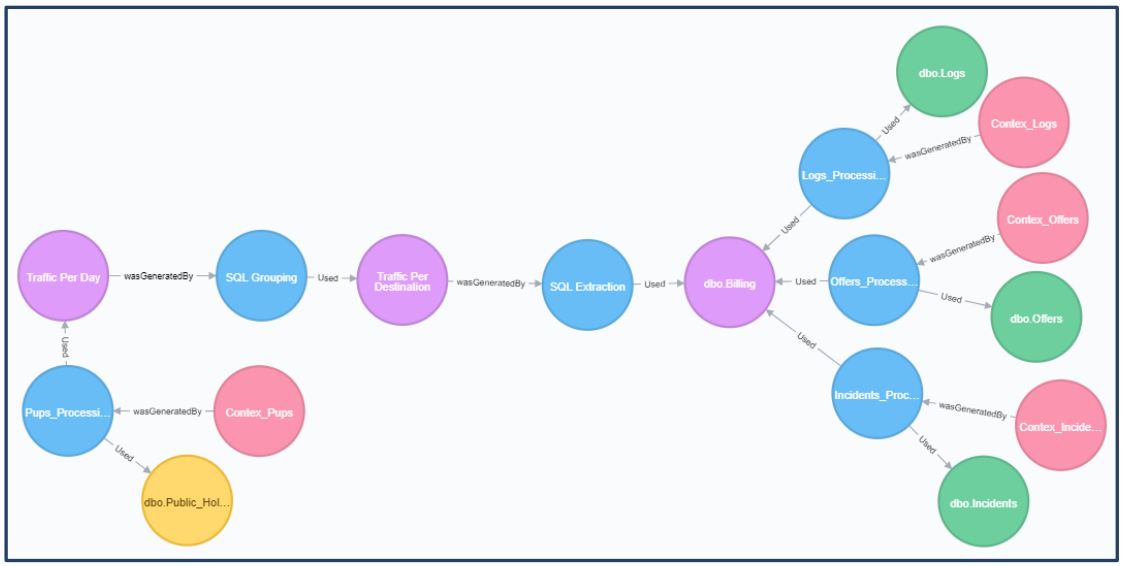


Figure 72 Provenance graph describing the execution of the data extraction query in the abnormal postpaid SMS traffic decrease use case over Billing. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 72 describes the trace of the sixth scenario (eighth sub-scenario) of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple node as a data source entity, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Summary_Per_Day Query Result entity at the left of the graph connected to

public holidays node and was produced by the Traffic_Summary_Per_Destination Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario will be extracted by a SQL query with a certain filter over Billing table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs on 1st -Nov. To back trace this value, the public holidays matched on the same day with the results table are mapped and the mapped attributes between the results table and the Traffic summary per destination are used to select the affected data in the drill down table, the filter and mapped attributes between Traffic per destination table and the Billing table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the traffic drop occurred due to an SMSC configurations change. And after the dataset was introduced to the model. The model

has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue through back till the logs node, noting that an SMSC configurations change has occurred on the same day of the abnormal behavior and may have caused the great traffic decrease.

The results answers the previously presented questions as:

1. Why the duration decrease has occurred? **An SMSC configurations change.**
2. When the revenue leakage issue has been introduced? **1st -Nov.**
3. Where the revenue leakage issue has occurred? At what node? **SMSC**.
4. What has caused this revenue leakage issue to occur? **An SMSC configurations change.**

5.2.2.1.4.9 Count of GPRS sessions from Billing (Billing measure for GPRS sessions).

In this scenario, a major drop in postpaid GPRS sessions traffic in terms of records count, data volume, and charge is recorded on 1st-Nov. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since the root-cause behind the traffic decrease must be identified. Figures 73-75 show the usage abnormal behavior, since the postpaid GPRS sessions per day trends follow a specific pattern in terms of count, volume, and charge. But in this scenario, we notice that there are much count, volume, and charge

decrease on 1st –Nov, and this behavior needs to be investigated to identify its root-cause and possible effects.

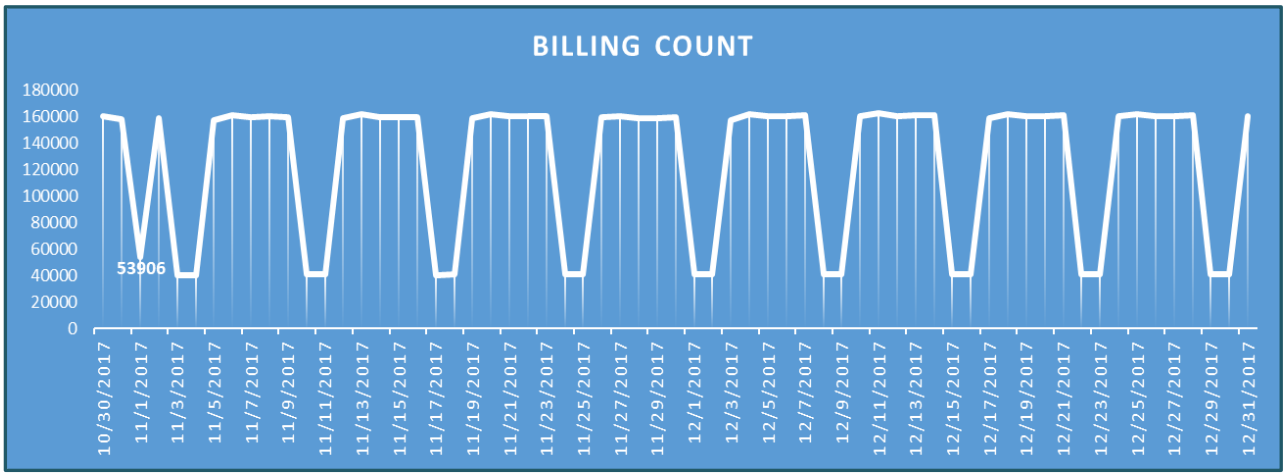


Figure 73 Count of postpaid GPRS sessions from Billing side

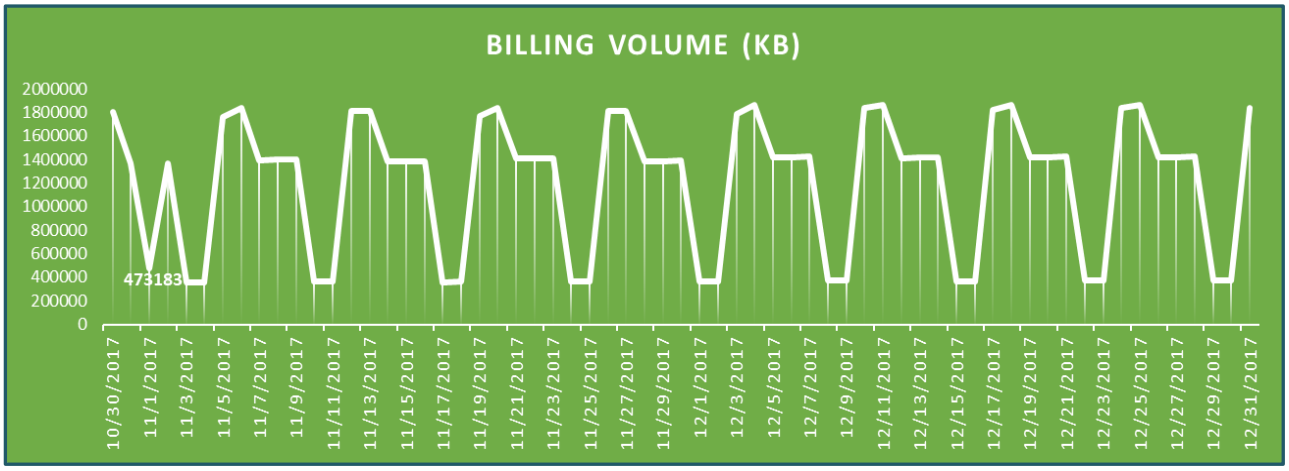


Figure 74 Total data volume of postpaid GPRS traffic from Billing side

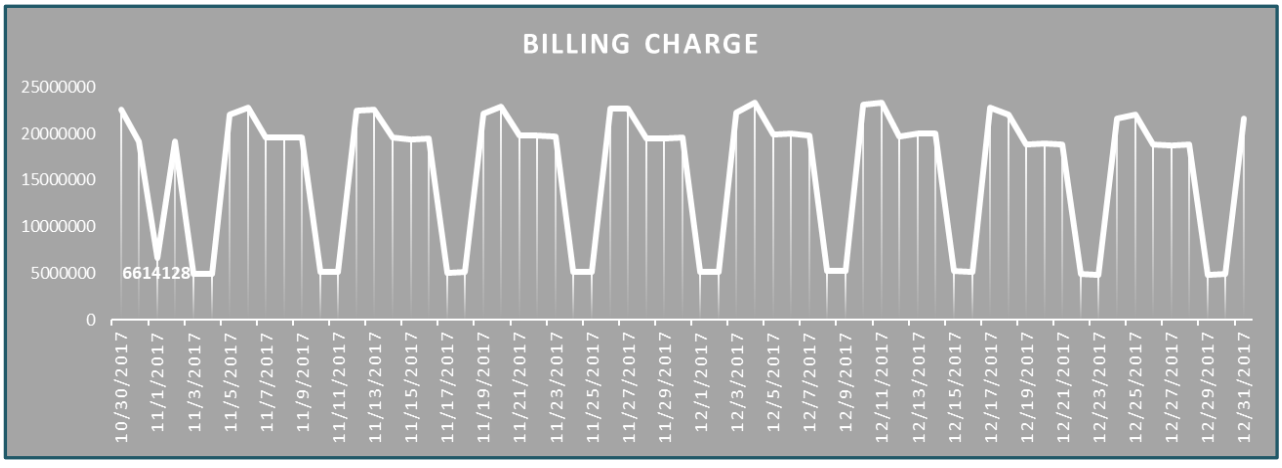


Figure 75 Total charge for postpaid GPRS traffic from Billing side

Here we need to answer the questions of:

1. Why the traffic increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for postpaid usage traffic monitoring and trending performs a SQL query with a filter to extract only GPRS sessions traffic for postpaid subscribers towards all APNs then it groups them per APN and per date afterwards in terms of transactions count, total data volume, and total charge.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 76 from the Neo4j after running the usage monitoring process for the date 1st-Nov.

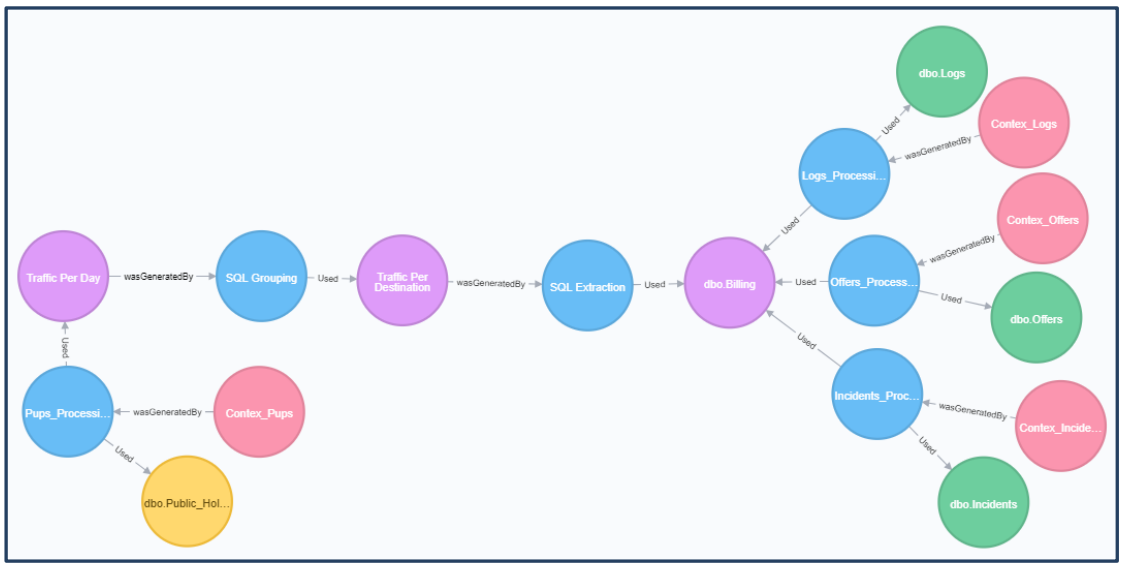


Figure 76 Provenance graph describing the execution of the data extraction query in the abnormal postpaid GPRS traffic decrease use case over Billing. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for

contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 76 describes the trace of the sixth scenario (Ninth sub-scenario) of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple node as a data source entity, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Summary_Per_Day Query Result entity at the left of the graph connected to public holidays node and was produced by the Traffic_Summary_Per_APN Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input datasets for this scenario will be extracted by a SQL query with a certain filter over Billing table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs on 1st -Nov. To back trace this value, the public holidays matched on the same day with the results table are mapped

and the mapped attributes between the results table and the Traffic summary per APN are used to select the affected data in the drill down table, the filter and mapped attributes between volume per APN table and the Billing table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the traffic drop occurred due to a SASN configurations change. And after the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue through back till the logs node, noting that a SASN configurations change has occurred on the same day of the abnormal behavior and may have caused the great traffic decrease.

The results answers the previously presented questions as:

5. Why the duration decrease has occurred? **A SASN configurations change.**
6. When the revenue leakage issue has been introduced? **1st -Nov.**
7. Where the revenue leakage issue has occurred? At what node? **SASN**.
8. What has caused this revenue leakage issue to occur? **A SASN configurations change.**

5.3.2.2 Human errors (Miss Configuration) leakage issues

5.2.2.2.1 Seventh use case scenario Prices Change

In this scenario, a major increase in total calls charge for prepaid subscribers is recorded with no additional major increase in calls count or calls duration. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since logically any additional voice charge must introduce additional usage in the normal case which is not happening here, indicating that a revenue leakage may have occurred and revenue losses are still ongoing because of customer complaints and churn (Loosing subscribers) due to charge increase per minute.

Figure 79 shows the charge abnormal behavior, since the prepaid voice calls total charge per day follows a specific pattern, higher calls charge on Sunday and Monday, much lower calls duration during Tuesday to Thursday, and very low calls charge on weekends. But on this trend, we notice that there are much charge increase starting from 24th-Dec till 31st-Dec. By taking a look at the calls count and calls duration trends, we do not notice any proportional increase to this charge increase as in Figures 77-78, which is unrealistic in normal situations.

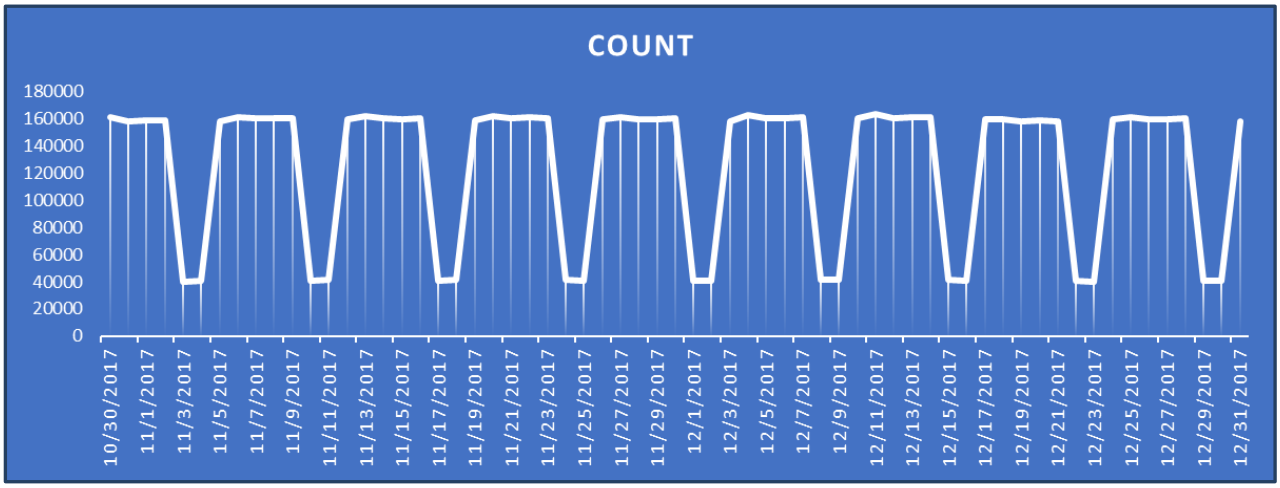


Figure 77 Prepaid voice calls count

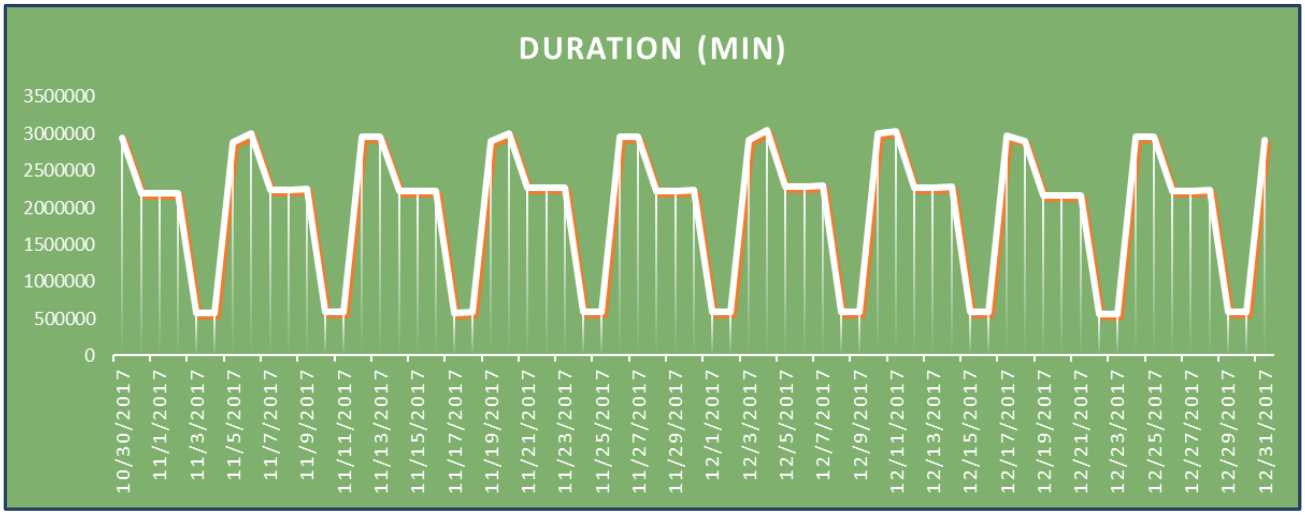


Figure 78 Prepaid voice calls duration

Figure 79 Prepaid voice cost

Here we need to answer the questions of:

1. Why the charge increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for prepaid usage traffic monitoring and trending performs a SQL query with a filter to extract only voice mobile originating traffic for prepaid subscribers for all destinations, then it groups them per destination and per date.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 80 from the Neo4j after running the usage monitoring process for the date of interest.

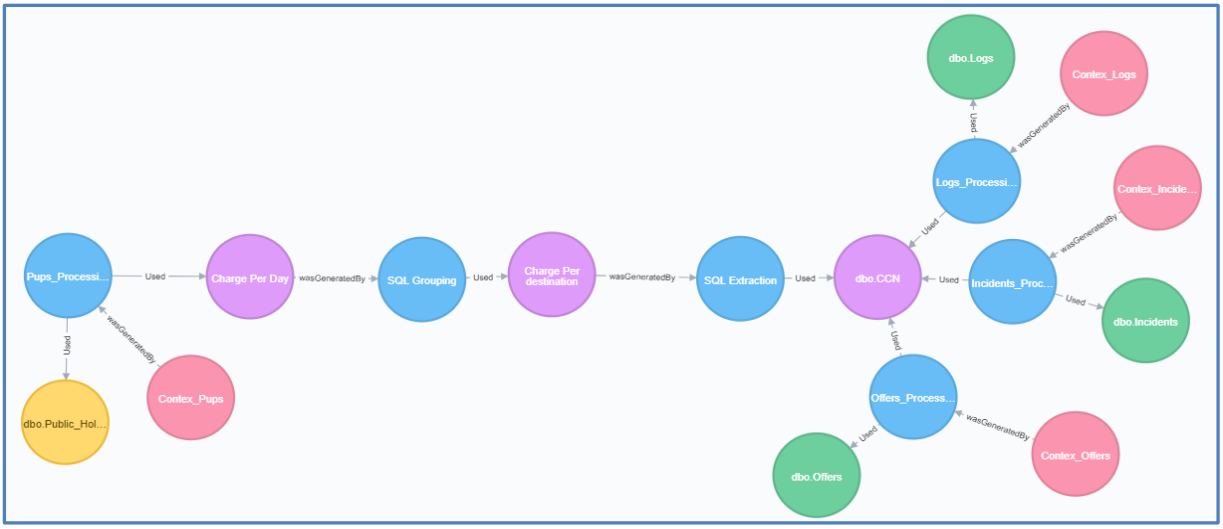


Figure 80 Provenance graph describing the execution of the data extraction query in the abnormal charge increase use case. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 80 describes the trace of the Seventh scenario of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple nodes as data source entities, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Charge_Per_Day Query Result entity at the left of the graph connected to the public holidays node and was produced by the Charge_Per_Destination Query Execution and Extraction process

that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input dataset for the Seventh scenario will be extracted by a SQL query with a certain filter over CCN table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs starting from 24th-Dec. To back trace this value, the mapped attributes between the results table and the charge per destination are used to select the affected data in the drill down table, the filter and mapped attributes between charge per destination table and the CCN table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the charge increase occurred due to prices change on 24th-Dec. After the dataset was introduced to the model, the model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information

was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue by identifying a prices change in the logs table on the same date.

The results answers the previously presented questions as:

1. Why the duration increase has occurred? **Prices change was occurred.**
2. When the revenue leakage issue has been introduced? **24th-Dec.**
3. Where the revenue leakage issue has occurred? At what node? **CCN**.
4. What has caused this revenue leakage issue to occur? **Prices change.**

5.3.2.3 Poor product or service design leakage issues

5.2.2.3.1 Eighth use case scenario Rounding Criteria (Fake Alert)

In this scenario, a major decrease in calls duration units and totals calls charge is recorded starting from 24th-Dec noting that total calls count and total calls duration in seconds remain the same as expected following the normal daily pattern due to a new product that changes the units rounding criteria. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side.

Figures 82-83 show the prepaid voice calls duration in units and total calls charge abnormal behavior. Here we notice a great decrease in duration in units and charge starting from 24th-Dec. But Figure 81 show a normal pattern without any recorded abnormal decrease.

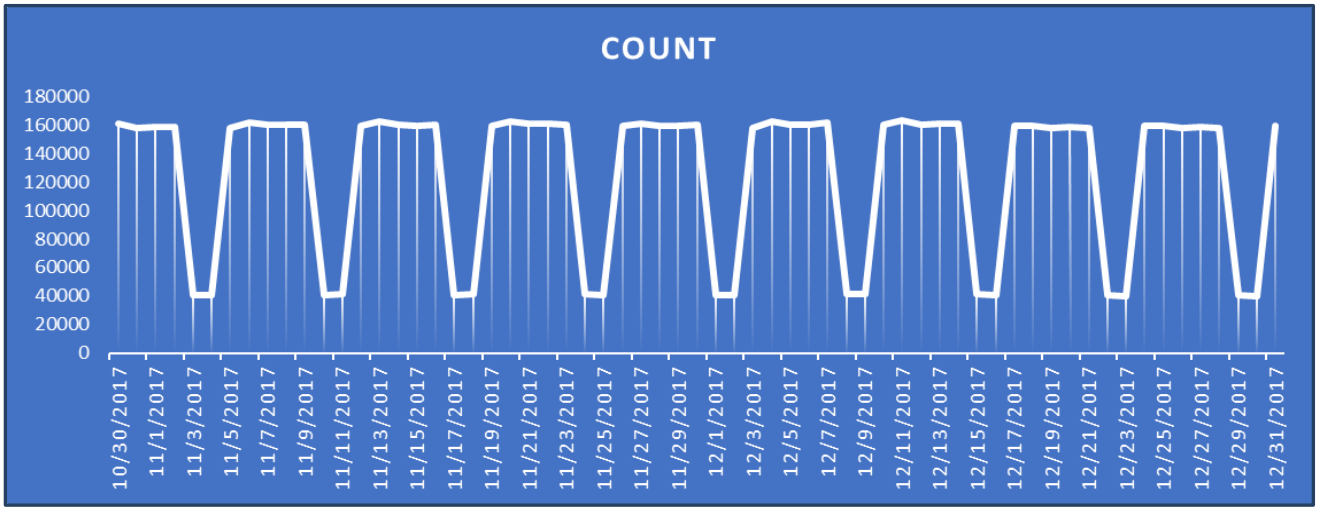


Figure 81 Prepaid voice calls count

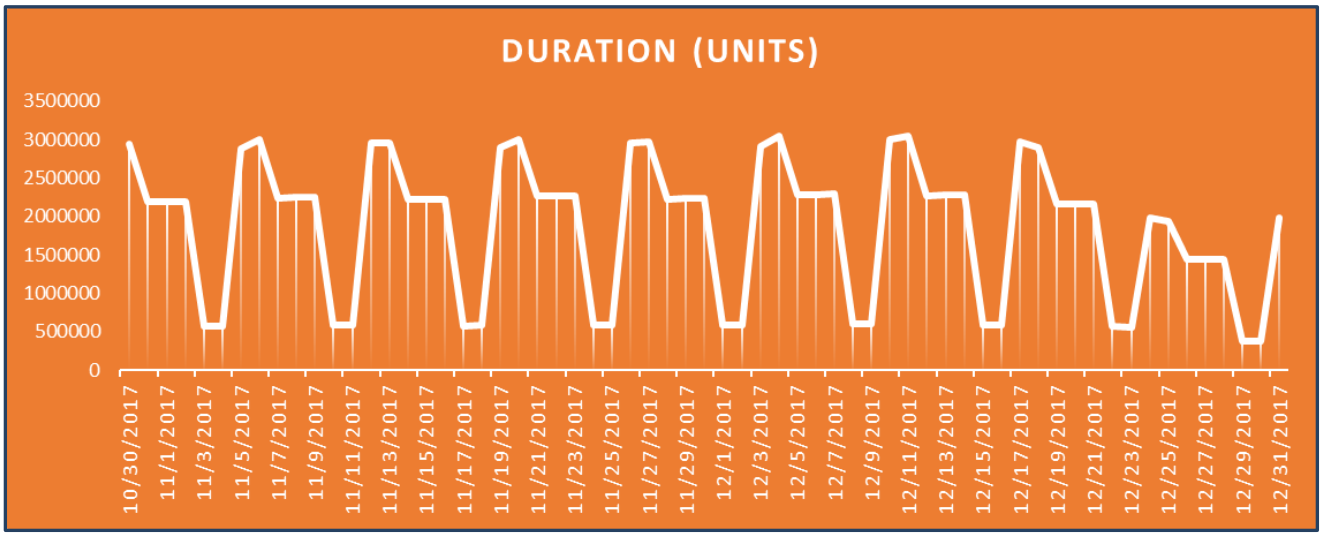


Figure 82 Prepaid voice calls duration in units

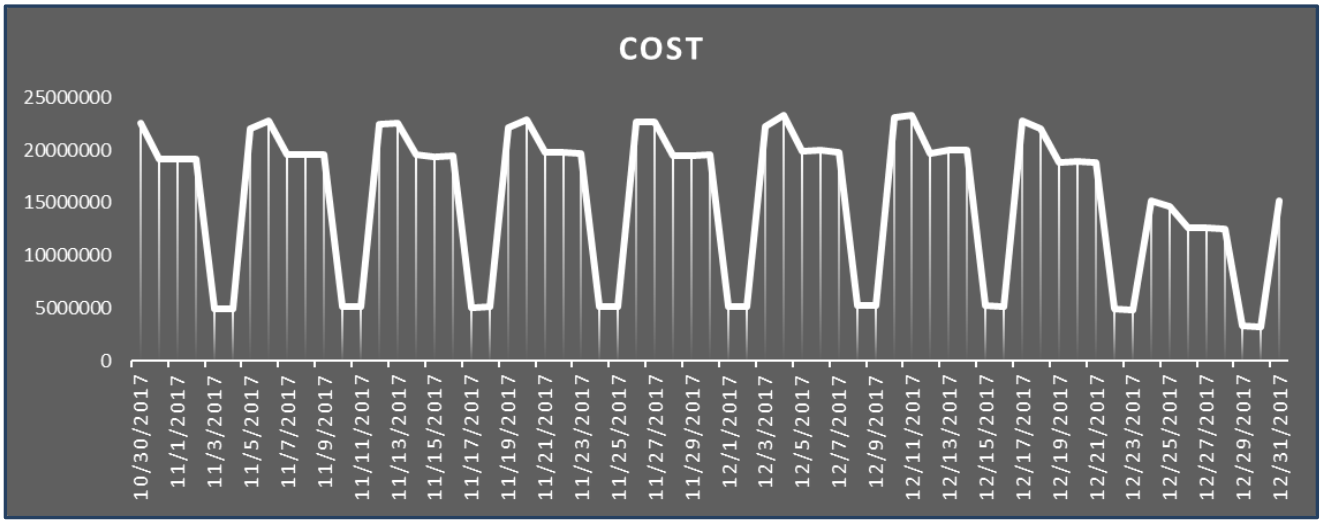


Figure 83 prepaid voice calls total charge

Here we need to answer the questions of:

1. Why the charge increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The RA function responsible for prepaid usage traffic monitoring and trending performs a SQL query with a filter to extract only charged voice mobile originating traffic for prepaid subscribers towards all destinations then it groups them per destination type and per date afterwards. The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 84 from the Neo4j after running the usage monitoring process for 24th-Dec.

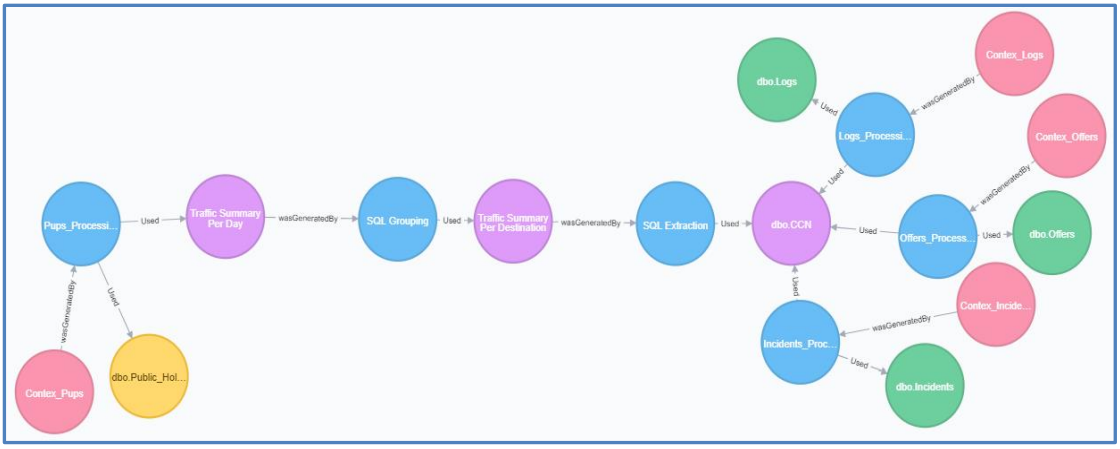


Figure 84 Provenance graph describing the execution of the data extraction query in the abnormal duration in units and charge decrease use case. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 84 describes the trace of the eighth scenario of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple nodes as data source entities, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Per_Day Query Result entity at the left of the graph connected to the public holiday node and was produced by the Traffic_Per_Destination Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input dataset for this scenario will be extracted by a SQL query with a certain filter over CCN table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal values occurs starting from 24th -Dec. To back trace this value, the mapped attributes between the results table and the traffic per destination are used to select the affected data in the drill down table, the filter and mapped attributes between duration per destination table and the CCN table are used to

select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that this behavior on 24th -Dec was due to a new product launch for rounding units of voice calls to 90 seconds instead of 60 seconds for the same price recorded on offers table. And after the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue.

The results answers the previously presented questions as:

1. Why the duration increase has occurred? **New product for rounding units of voice calls to 90 seconds instead of 60 seconds for the same price.**
2. When the revenue leakage issue has been introduced? **24th -Dec.**
3. Where the revenue leakage issue has occurred? At what node? **CCN**.
4. What has caused this revenue leakage issue to occur? **New product for rounding units of voice calls to 90 seconds instead of 60 seconds for the same price.**

5.2.2.3.2 Ninth use case scenario Conflicting Campaigns (Real Alert)

In this scenario, a major decrease in prepaid international calls traffic using a certain product (campaign) is recorded due to another conflicting campaigns launch for

prepaid subscribers and the subscribers could benefit from both campaigns, affecting each product estimated revenue. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since this is an indicator that a revenue leakage may have occurred. Figures 85-87 show the great decrease in calls count, duration and total charge respectively starting from 24th-Dec.

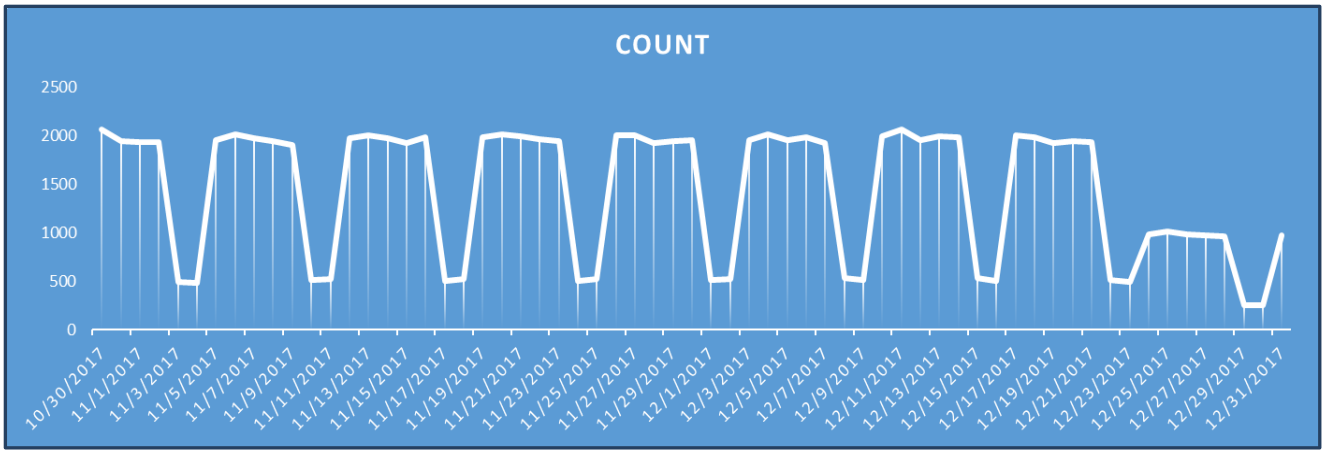


Figure 85 Prepaid international voice calls count for Product ID 1120

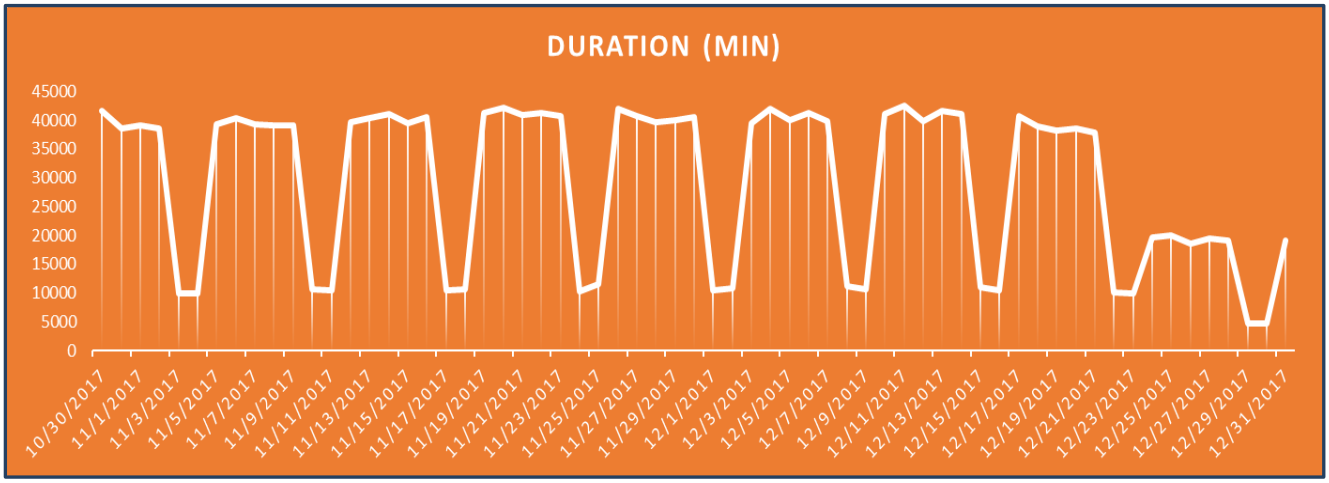


Figure 86 Prepaid international voice calls duration for Product ID 1120

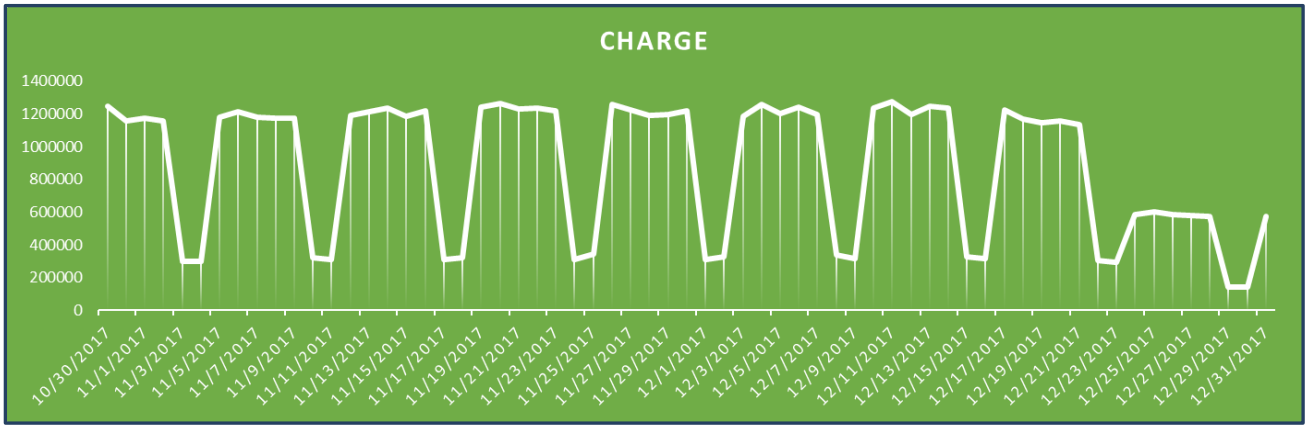


Figure 87 Prepaid international voice calls charge for Product ID 1120

Here we need to answer the questions of:

1. Why the charge increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for prepaid usage traffic monitoring and trending performs a SQL query with a filter to extract all voice mobile originating calls for prepaid subscribers towards international destinations then it groups them per product ID and per date afterwards as in below process.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 88 from the Neo4j after running the usage monitoring process for the date of interest.

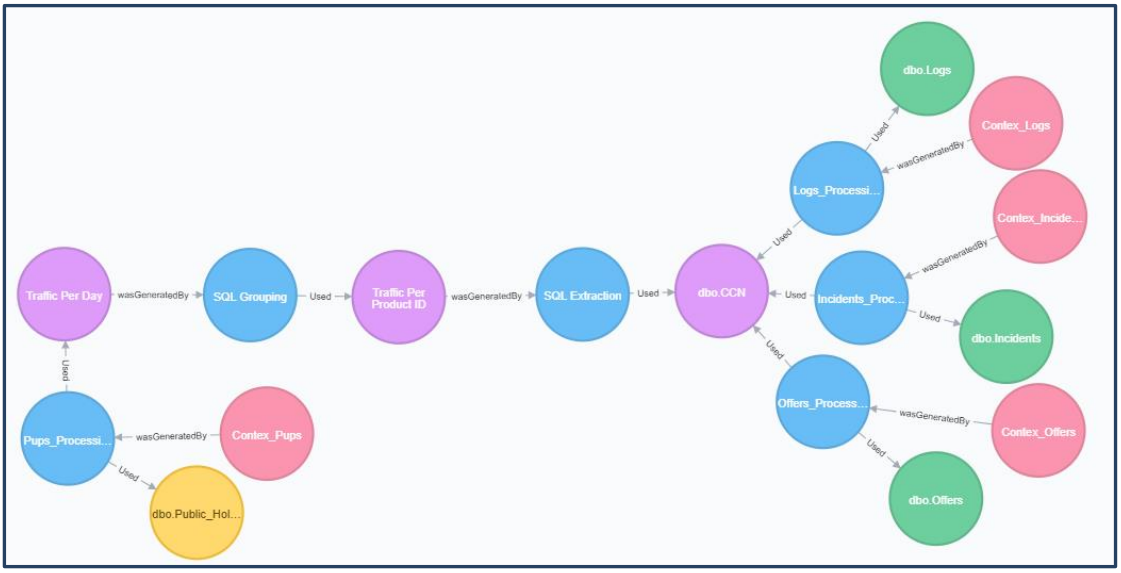


Figure 88 Provenance graph describing the execution of the data extraction query in the abnormal campaign international total traffic decrease use case. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 88 describes the trace of the ninth scenario of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple nodes as data source entities, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Per_Day Query Result entity at the left of the graph connected to the public holidays node and was produced by the Traffic_Per_Product_ID Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input dataset for the ninth scenario will be extracted by a SQL query with a certain filter over CCN table. At each step of execution the generated data is stored in

a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value starts occurring from 24th-Dec. To back trace this value, the mapped attributes between the results table and the traffic per Product_ID are used to select the affected data in the drill down table, the filter and mapped attributes between traffic per Product_ID table and the CCN table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that drop in total international calls for product_ID 1120 occurred due to a launch of a conflicting campaign on 24th –Dec. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue by identifying another conflicting campaign for prepaid international voice traffic in the offers table on the same date.

The results answers the previously presented questions as:

1. Why the duration increase has occurred? **Conflicting campaigns.**
2. When the revenue leakage issue has been introduced? **24th-Dec.**
3. Where the revenue leakage issue has occurred? At what node? **CCN**.
4. What has caused this revenue leakage issue to occur? **Conflicting campaigns.**

5.3.2.4 Internal and external fraud leakage issues

5.3.2.4.1 Internal fraud

5.3.2.4.1.1 Tenth use case scenario Voice Price

In this scenario, a major decrease in calls charge is recorded due to an increase in total calls with zero charge for prepaid subscribers. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since this is an indicator that a revenue leakage may have occurred. Figure 91 shows the great decrease in calls total charge on 23rd -Nov, which is a Thursday compared to other Thursdays while total calls count and total calls duration remains the same as in Figures 89-90.

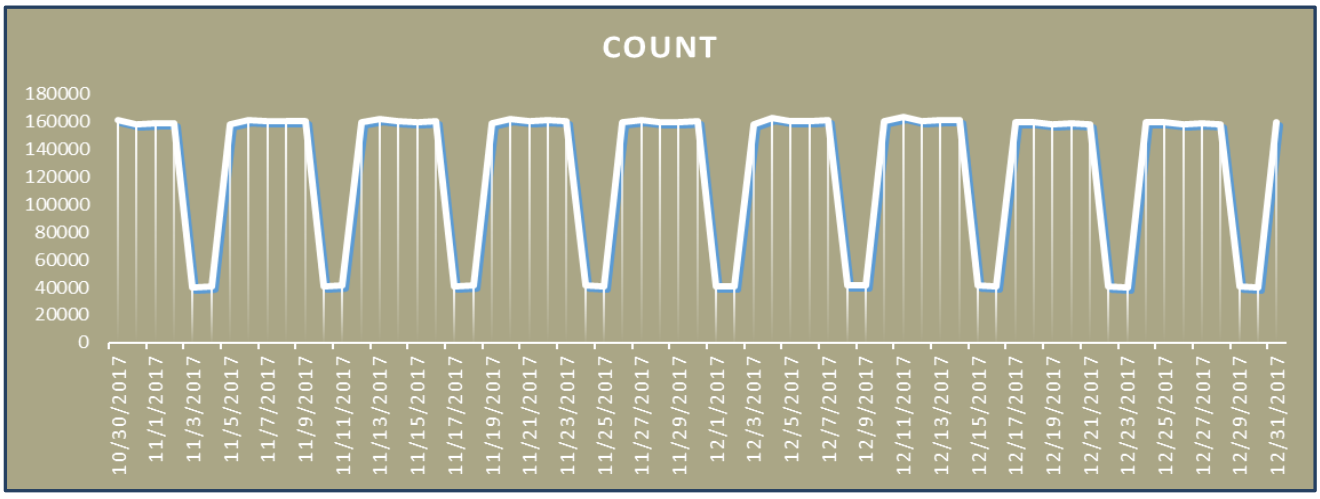


Figure 89 Prepaid voice calls count

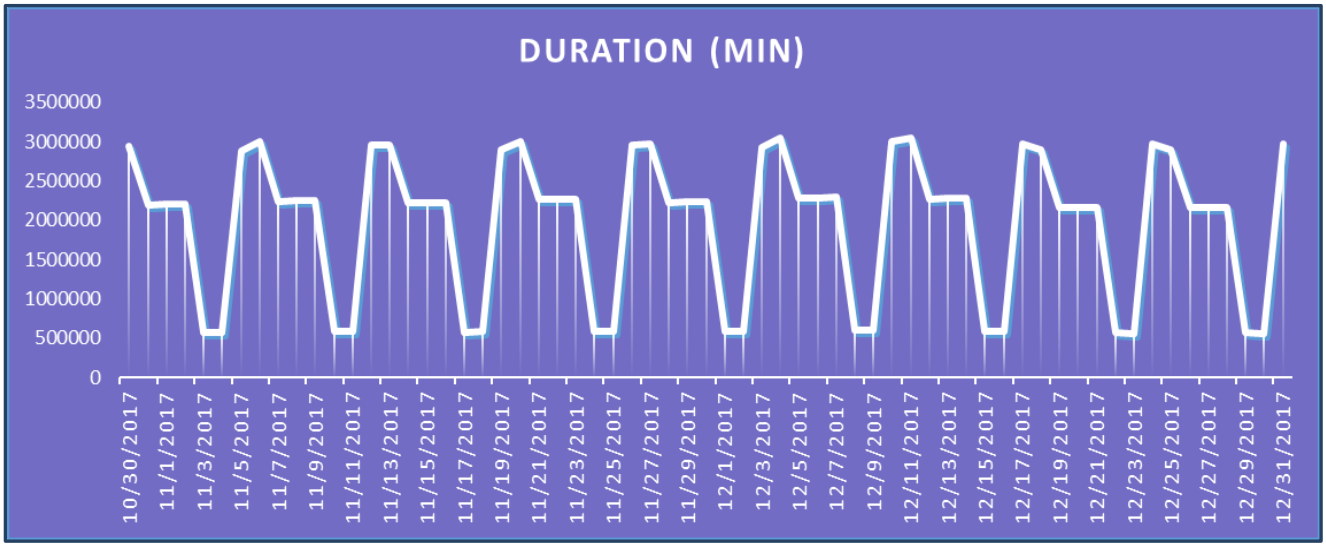


Figure 90 Prepaid voice calls duration

Figure 91 Prepaid voice calls cost

Here we need to answer the questions of:

5. Why the charge increase has occurred?
6. When the revenue leakage issue has been introduced?
7. Where the revenue leakage issue has occurred? At what node?
8. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for prepaid usage traffic monitoring and trending performs a SQL query with a filter to extract all voice mobile originating

calls for prepaid subscribers towards all destinations then it groups them per destination type and per date afterwards as in below process.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 92 from the Neo4j after running the usage monitoring process for the date of interest.

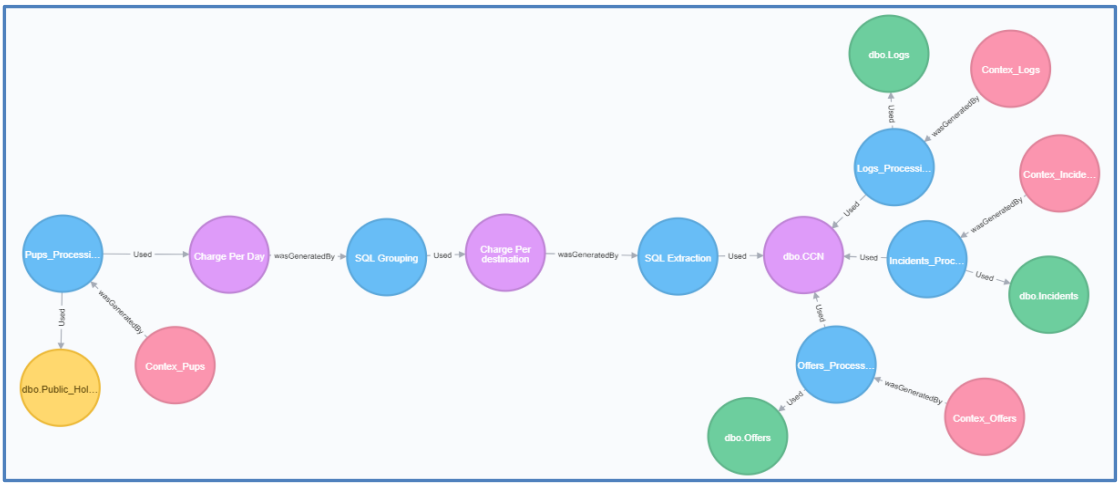


Figure 92 Provenance graph describing the execution of the data extraction query in the abnormal charge decrease use case. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 92 describes the trace of the tenth scenario of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple nodes as data source entities, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Charge_Per_Day

Query Result entity at the left of the graph connected to the public holidays node and was produced by the Charge Charge _Per_Destination Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input dataset for the tenth scenario will be extracted by a SQL query with a certain filter over CCN table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs on 23rd -Nov. To back trace this value, the mapped attributes between the results table and the charge per destination are used to select the affected data in the drill down table, the filter and mapped attributes between charge per destination table and the CCN table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that drop in total calls charge occurred due to a wrong tariff configuration on 23rd –Nov by a charging employee through the definition

minute price to zero recorded in the logs table. After the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue by identifying a prices change in the logs table on the same date.

The results answers the previously presented questions as:

5. Why the duration increase has occurred? **Wrong tariff configuration.**
6. When the revenue leakage issue has been introduced? **23rd -Nov.**
7. Where the revenue leakage issue has occurred? At what node? **CCN**.
8. What has caused this revenue leakage issue to occur? **Wrong tariff configuration.**

5.2.2.4.1.2 Eleventh use case scenario SMS Price

In this scenario, a major decrease in total SMSs charge is recorded due to a major increase in total SMSs with zero charge for prepaid subscribers on 7th-Nov. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since this is an indicator that a revenue leakage may have occurred. Figure (94) shows the great drop in SMS charges, while total SMS count remains the same as expected to be as in Figure 93, which is unrealistic in the normal case.

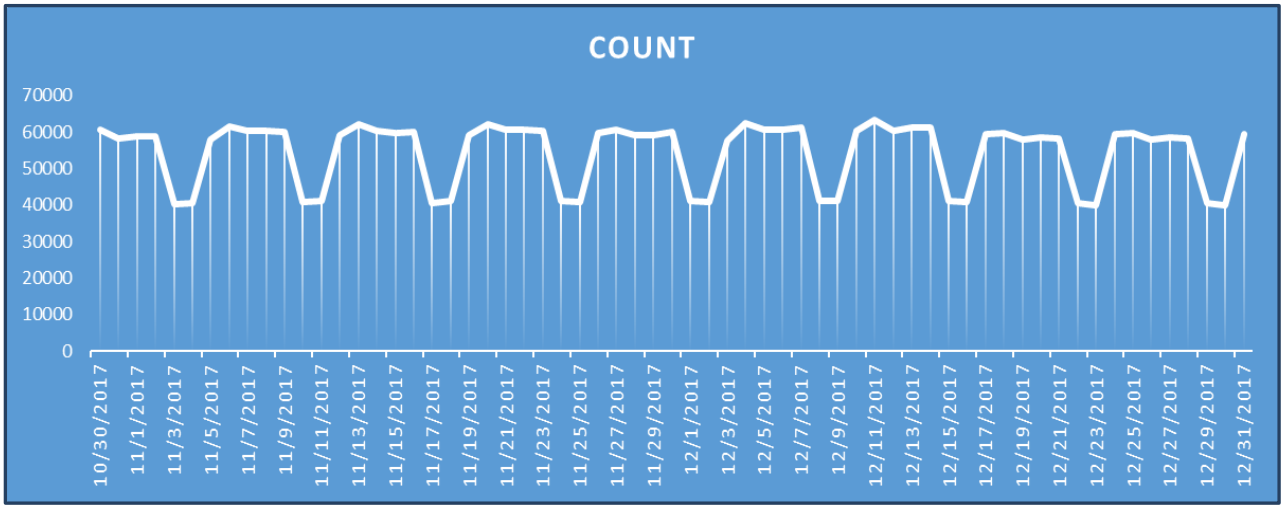


Figure 93 Prepaid SMS total count

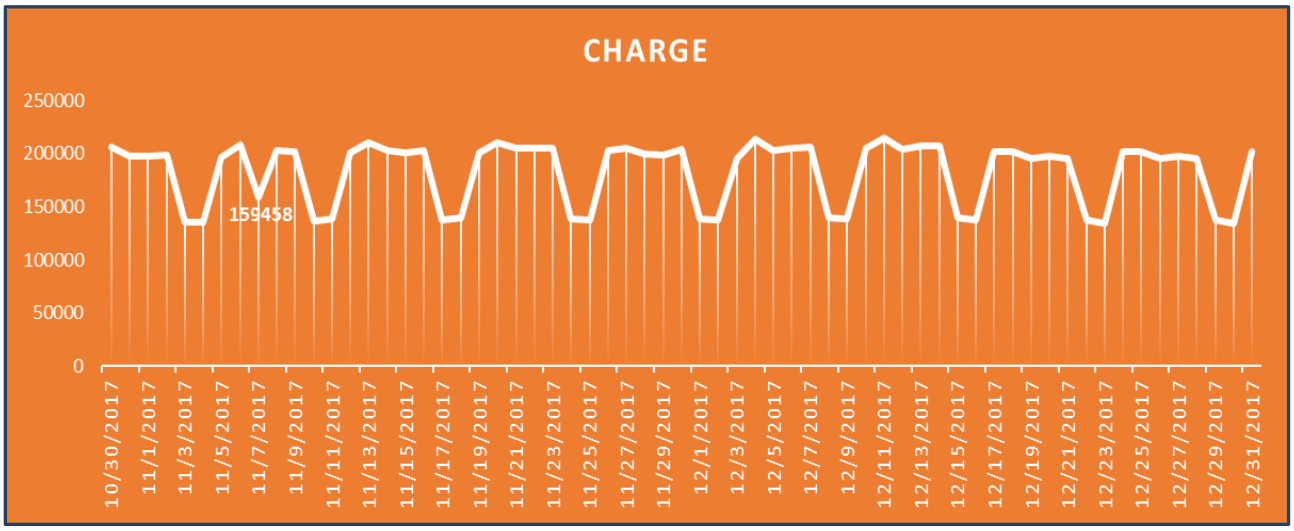


Figure 94 Prepaid SMS total charge

Here we need to answer the questions of:

1. Why the charge increase has occurred?
2. When the revenue leakage issue has been introduced?
3. Where the revenue leakage issue has occurred? At what node?
4. What has caused this revenue leakage issue to occur?

The responsible RA function for prepaid usage traffic monitoring and trending performs a SQL query with a filter to extract all originating SMS traffic for prepaid subscribers towards all destinations then it groups them per destination type and per

date afterwards as in below process. The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 95 from the Neo4j after running the usage monitoring process for 7th-Nov.

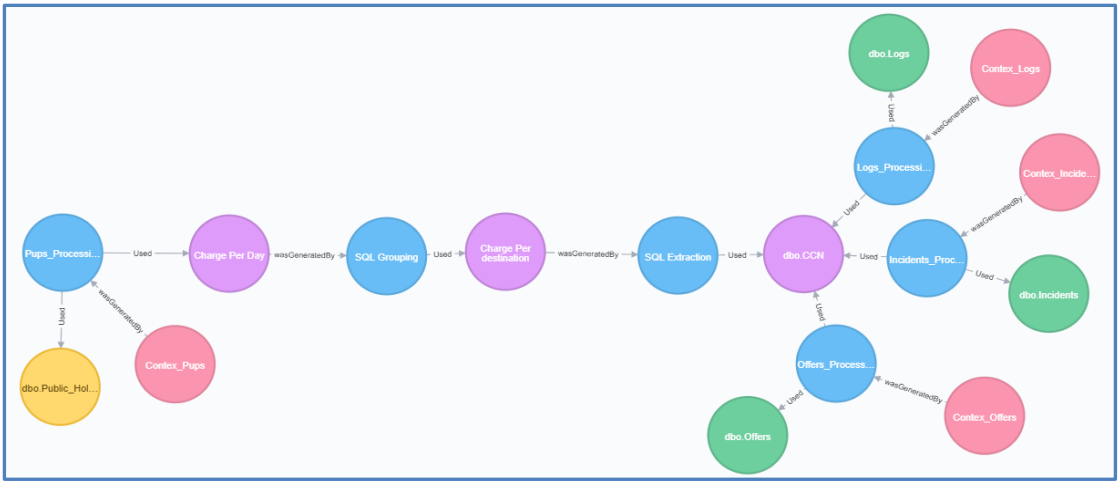


Figure 95 Provenance graph describing the execution of the data extraction query in the abnormal charge decrease use case. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 95 describes the trace of the eleventh scenario of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple nodes as data source entities, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Charge_Per_Day Query Result entity at the left of the graph connected to the public holidays node and was produced by the Charge_Per_Destination Query Execution and Extraction process

that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input dataset for the eleventh scenario will be extracted by a SQL query with a certain filter over CCN table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs on 7th-Nov. To back trace this value, the mapped attributes between the results table and the charge per destination are used to select the affected data in the drill down table, the filter and mapped attributes between charge per destination table and the CCN table are used to select the affected data from the drill down table and use the filter stored between incidents, logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the zero charged SMS count increase occurred due to prices change because of a wrong tariff configuration on 7th–Nov. And after the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down

capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue by identifying a prices change in the logs table on the same date.

The results answers the previously presented questions as:

9. Why the duration increase has occurred? **Wrong tariff configuration.**
10. When the revenue leakage issue has been introduced? **7th-Nov.**
11. Where the revenue leakage issue has occurred? At what node? **CCN**.
12. What has caused this revenue leakage issue to occur? **Wrong tariff configuration.**

5.3.2.4.2 External fraud

5.2.2.4.2.1 Twelfth use case scenario GPRS Initiation

In this scenario, a major increase in total GPRS sessions count and data volume for prepaid subscribers is recorded with no additional increase in GPRS traffic charges. This scenario has been used as an example of an abnormal behavior in the usage monitoring trends that needs to be investigated from RA team side, since logically any additional GPRS usage must introduce additional fees or charges in the normal case which is not happening here, indicating that a revenue leakage may have occurred and revenue losses are still ongoing.

Figures 96-98 show the usage abnormal behavior, since the prepaid GPRS traffic per day follows a specific trend. But on these trends, we notice that there are much GPRS sessions count and data volume increase starting from 3rd-Nov till 5th-Nov,

and we do not notice any proportional increase to this traffic increase, which is unrealistic in the normal case.

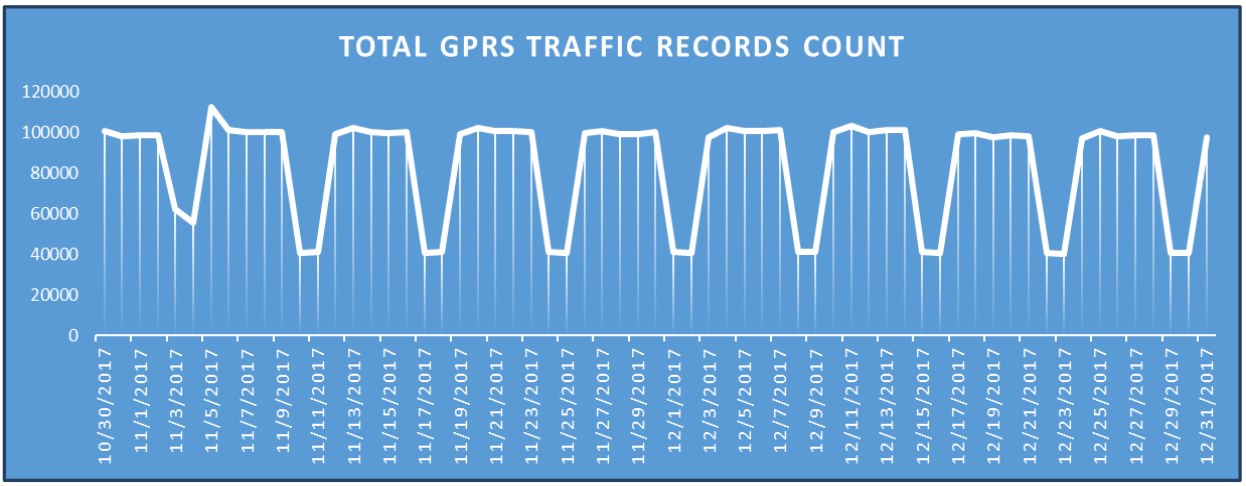


Figure 96 Prepaid GPRS sessions count

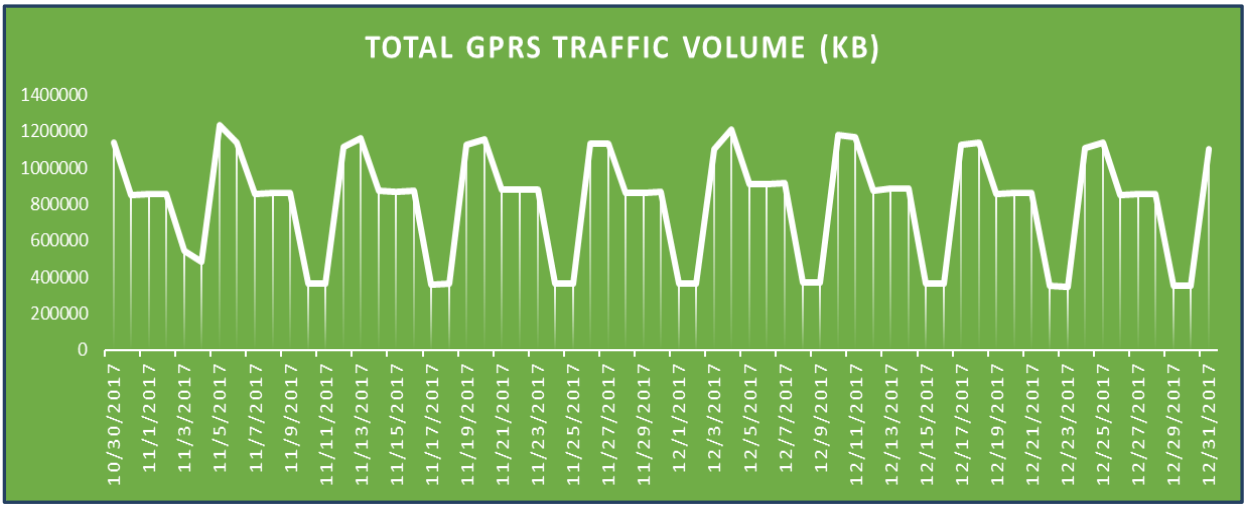


Figure 97 Prepaid GPRS sessions data volume

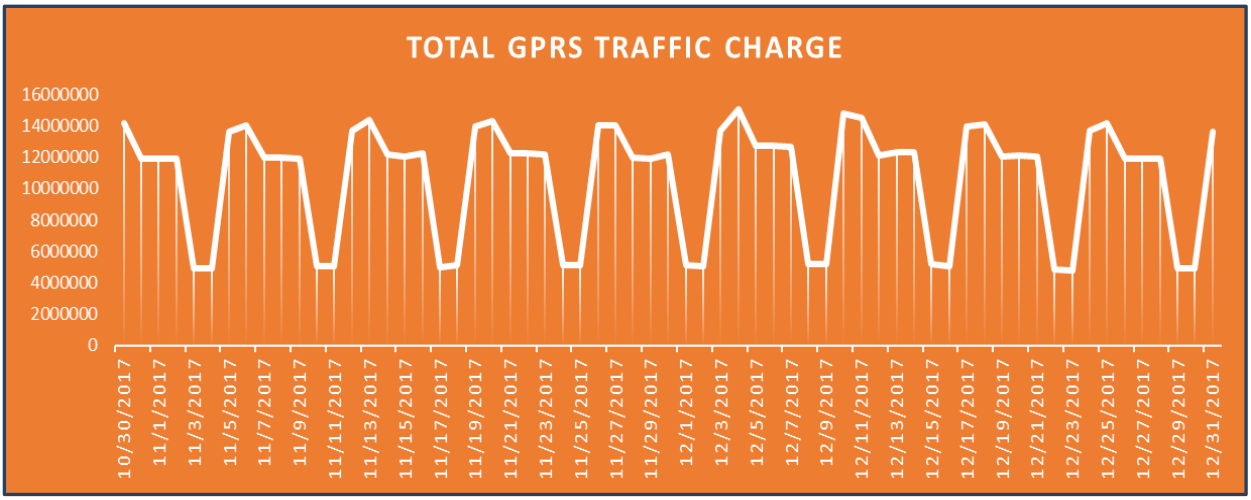


Figure 98 Prepaid GPRS sessions charge

Here we need to answer the questions of:

5. Why the duration increase has occurred?
6. When the revenue leakage issue has been introduced?
7. Where the revenue leakage issue has occurred? At what node?
8. What has caused this revenue leakage issue to occur?

The responsible RA function responsible for prepaid usage traffic monitoring and trending performs a SQL query with a filter to extract only GPRS traffic for prepaid subscribers towards all APNs then it groups them per APN and per date afterwards.

The developed system prototype automatically builds a data workflow on the Neo4j graph database representing all data nodes included in the process, associated nodes and executed SQL processing nodes. Each processing node in the provenance workflow stores the semantic information, which consist of mapped attributes and used filters, such as in Figure 99 from the Neo4j after running the usage monitoring process for the date of interest.

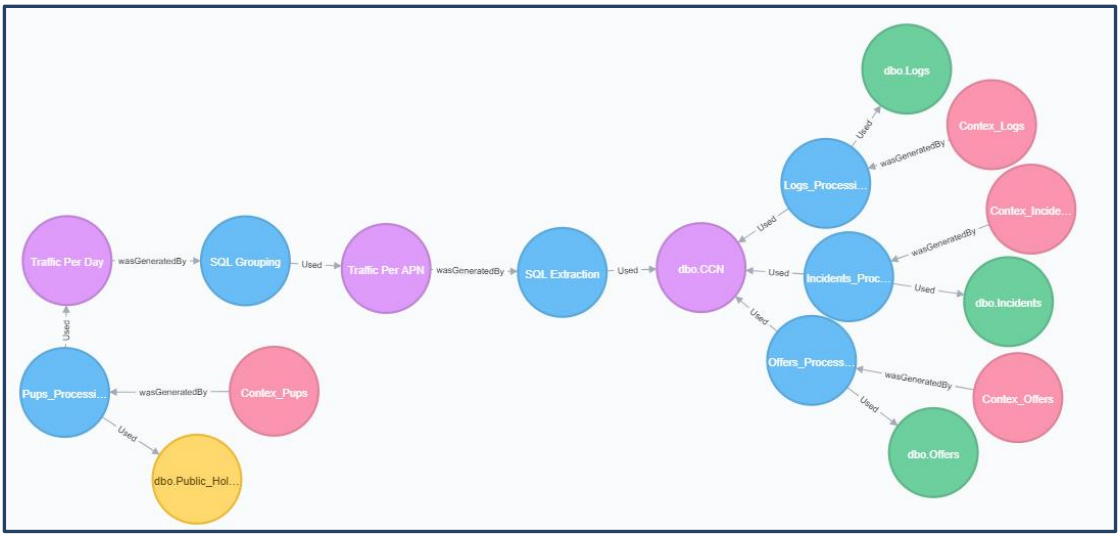


Figure 99 Provenance graph describing the execution of the data extraction query in the abnormal prepaid GPRS traffic increase use case. Blue nodes denote SQL processing nodes, Purple node denote data source entity, green nodes denote associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node denote associated entity to the final destination entity for contextual information provision using SQL processing node.

The provenance graph shown in Figure 99 describes the trace of the twelfth scenario of our model built using the Neo4j database, with blue nodes as SQL processing nodes, Purple node as a data source entity, green nodes as associated entities to the data source entity for contextual information provision using SQL processing nodes, and yellow node as associated entity to the final destination entity for contextual information provision using SQL processing node. The Traffic_Per_Day Query Result entity at the left of the graph was produced by the Traffic_Per_Day Query Execution and Extraction process that contains the original SQL query, which in turn connected to the data source entity that is connected to the contextual information provision entities (Incidents, Logs, and Offers) using SQL processing nodes.

The input dataset for the twelfth scenario is extracted by a SQL query with a certain filter over CCN table. At each step of execution the generated data is stored in a separate table to support drill down capability. After the query is executed provenance information is generated consisting of attributes mapped between the input and output tables, used filters, and contextual information provisioned. Nodes are automatically being added to the workflow and provenance semantic information are inserted into SQL processing nodes, then the row with the abnormal value in the results table is investigated through backward tracing by retrieving all processing nodes from the graph database and reverse their order of execution till the source entity is reached with its associated entities.

In the given scenario the abnormal value occurs starting from 3rd-Nov till 5th-Nov. To back trace this value, the mapped attributes between the results table and the traffic per APN are used to select the affected data in the drill down table, the filter and mapped attributes between traffic per APN table and the CCN table are used to select the affected data from the drill down table and use the filter stored between incidents,

logs, and offers tables to enrich the data obtained about the abnormal behavior, if there was an incident, changes, or offers on that date that may cause this leakage or fake alert of leakage.

The used scenario assumes that the GPRS traffic abnormal increase occurred after an external fraud to change GPRS traffic to look as an initiation traffic to the charging system, which is a zero charged traffic starting from 3rd-Nov. After the dataset was introduced to the model. The model has effectively built the data workflow graph, drill down tables constructed at each processing step to enhance the drill down capability, provenance semantic information was stored at the SQL processing nodes and the contextual information were invoked from associated entities using the processing nodes, retrieved and used for debugging the root-cause for the presented issue, and has successfully identified the reason of the issue as external fraud from public holidays and events node.

The results answers the previously presented questions as:

1. Why the GPRS traffic increase has occurred? **The use of a fraudulent application to use GPRS for free.**
2. When the revenue leakage issue has been introduced? **3rd-Nov.**
3. Where the revenue leakage issue has occurred? At what node? **CCN**.
4. What has caused this revenue leakage issue to occur? **The use of a fraudulent application to use GPRS for free.**

Chapter 6 Evaluation

This chapter describes the evaluation methodology used to evaluate our approach, the evaluation metrics, the datasets used for testing and evaluation, and finally discusses the evaluation results.

6.1 *Evaluation methodology*

This section describes the evaluation methodology used to evaluate the proposed approach using completeness and accuracy, meaning that the approach needs to be evaluated in terms of how many revenue leakage issues root-causes were identified out of the total detected revenue leakage issues, and how many root-causes were accurate and correct out of the total predicted root causes. In order to verify our proposed approach, we have first to verify that all of its elements work according to the designed conceptual model and according to its specified theoretical design. In this study we have tested the total of 12 revenue leakage root-cause instances with 26 symptoms, each root-cause must have at least one symptom as an effect of its occurrence. These 12 issues include true positives and false positives, datasets were introduced to the proposed model to evaluate the root-cause analysis and drill down features for each. We have four main different types of cases, of issues, as described below:

- True Positive (TP): Real alert, raises an alert for a real revenue leakage problem.
- True Negative (TN): No alert is raised as there is no real problem indicator.
- False Positive (FP): Fake alerts, raises an alert for a fake revenue leakage problem indicator.

- False Negative (FN): No alert is raised while there is a real revenue leakage problem.

The evaluation was conducted in two stages: we have conducted a dry run experiments based on the evaluation methodology by running the proposed model on two revenue leakage root-cause instances to check if the experiment design works well; then a wet run experiments were conducted on 12 revenue leakage root-cause instances with 26 possible revenue leakage symptoms to evaluate completeness and correctness of our approach. Our approach aims to identify all possible reasons for detected revenue leakage issues and identify them correctly and accurately. The two types of evaluation features used in this study will be discussed in the following section.

In order to correctly evaluate our approach and find out more about its limitations and strengths, we followed the below steps:

1. We have prepared 10 datasets per revenue leakage root-cause instance symptom, which means 260 datasets, to evaluate the ability of our approach to identify the root-cause for all possible symptoms of introduced revenue leakage issues and its accuracy.
2. Another 26 datasets for TN per issue to represent the dataset without any issue or revenue leakage symptom, in order to identify the approach ability of not identifying root causes for datasets that do not have leakage issues and do not raise alerts.
3. The datasets were introduced to the developed model, and results were recorded and evaluated.
4. For each introduced dataset, completeness was evaluated in terms of how many leakage issues root-cause were identified out of the total number of revenue leakage alerts.

5. For each introduced dataset, accuracy was evaluated in terms of how many identified leakage issues root-causes were correct out of the total number of identified revenue leakage issues root-causes.

On the other hand, it is worth mentioning that conducting a comparative evaluation with the existing provenance-based approach Panda, which was our reference for improving and extending the current rule-based revenue leakage detection approach is not scientifically valid. The Panda system performs syntactic analysis of SQL queries to capture provenance information (used filters and mapped attributes). It, then, uses generated syntactic analysis data to trace operations to investigate the reason of erroneous values of the SQL queries to locate where the error has occurred. While in our proposed approach it uses the same provenance data and tracing method, but additionally enhanced with contextual information to perform root-cause of the revenue leakage issue, which Panda is not designed to do.

6.2 *Evaluation features*

This section describes the evaluation features used to evaluate the proposed approach as below:

1. Completeness test was done to insure that our proposed approach could perform root-cause analysis and debugging for all different use cases that generate alerts. In other words, completeness refers to the proportion of predicted root-causes of the total number of alerts per root-cause instance [14, 30]. But in the case of the TN it refers to the proportion of total number of unpredicted root-causes over the total number of datasets as in equation 1. In order to evaluate the completeness of our approach, we use a gold standard test with 12 revenue leakage issues (with their root causes) and 26 possible revenue leakage symptoms as a gold standard discussed in the implementation chapter.

$$Completeness = \frac{PRC}{T} \qquad (1)$$

Where T is the total number of alerts

And PRC is the number of predicated root-cause

2. Accuracy (Correctness) test was done to evaluate how many of the predicted root-causes for detected cases were correct, whether these cases were real leakage issues or just fake alerts indicating an abnormal behavior. In other words, accuracy refers to the proportion of correctly predicted root-causes of the total number of predicted root-causes per root-cause instance [30]. But in the case of the TN it refers to the proportion of total number of unpredicted root-causes over the total number of datasets as in equation 2. In order to evaluate the accuracy and correctness of our approach, we use a gold standard test with 12 revenue leakage issues (with their root causes) and 26 possible revenue leakage symptoms as a gold standard discussed in the implementation chapter.

$$Accuracy = \frac{CPRC}{PRC} \qquad (2)$$

Where PRC is the number of predicated root-cause

And CPRC is the number of correctly predicted root-cause

6.3 *Data Collection*

The datasets consist of call detail records originated from five network nodes: MSC, SASN, SMSC, Billing and CCN, which are responsible for telecom usage traffic. For example, once a call is originated from the MSC, the MSC triggers the charging system CCN to start the call online charging and both nodes generated call records. If a record is missing from one of the sources, this means:

- If the record is missing from the CCN but exist on the MSC side, it means that the call occurred but not charged which means revenue loss, or that the call record is not loaded into RA system which means inaccurate monitoring and reporting of the data.
- If the record is missing from the MSC but exist on the CCN side, it means that the call did not occur but the subscriber was charged which means customer complaints and revenue loss in long term, or that the call record is not loaded into RA system which means inaccurate monitoring and reporting of the data.

The datasets were generated using call detail generation tool developed by a Google employee called Paul Kinlan [25]. The tool provides the researcher with the ability to change many parameters to generate a near real dataset, such as:

1. Calls or GPRS sessions made per subscriber.
2. Number of accounts.
3. Start date
4. End Date.
5. Call Types.
6. Day distribution.
7. Outgoing call parameters, such as cost.
8. Time distribution.

To ensure valid representation and consistency of the generated dataset for the considered scenarios, the data generation tool has been validated by the author manually. For each of the considered scenarios, the values of above parameters were repeatedly changed to generate a small dataset, which were then manually inspected and checked. In all cases, the generated datasets were found to be valid

representation of the input scenarios and describe the real behaviour of the telecom traffic in terms of peak and off peak periods, destinations, different usage types, different subscriber accounts, defined cost amounts per destination type, and traffic distribution among days. The generated datasets provided the repetitive behaviour for the same parameter values. Further explanation regarding the dataset generation process can be found in the appendix.

The datasets were generated over two months to cover ten use cases (including eight sub scenarios):

- CCN Data with Major duration Increase with no cost increase on local traffic from 26th -Dec till 31st-Dec. This dataset was generated to evaluate the ability of the approach to accurately identify root-causes for fake alerts that do not result in any revenue loss but they represent an extremely strange behavior. This strange behavior was introduced by adding 3 additional minutes to the calls duration starting from 26th-Dec without increasing the call charge.
- Missing records from CCN due to an MSC Disconnection on 26th-Dec. This dataset was generated to evaluate the ability of the approach to accurately identify root-causes for real alerts that result in revenue losses due to MSC disconnection caused by switch upgrade. This issue was introduced by not introducing MSC calls to the charging system table on the date of the problem.
- Missing records originated from MSC, SMSC and SASN from CCN side due to CCN Disconnection on 5th-Dec. These datasets were generated to evaluate the ability of the approach to accurately identify root-causes for real alerts that result in revenue losses due to CCN disconnection caused by power failure at SDP site. This issue was introduced by not introducing MSC, SMSC, and SASN transactions to the charging system table on the date of the problem.

- Mismatched records originated from MSC and SASN with records originated from CCN side due to a CCN Disconnection on 5th-Dec. These datasets were generated to evaluate the ability of the approach to accurately identify root-causes for real alerts that result in revenue losses due to CCN disconnection caused by power failure at SDP site and causing partial transactions charging. This issue was introduced by not introducing MSC and SASN transactions with less data volume and call duration to the charging system table on the date of the problem.
- Missing records originated from MSC, SMSC and SASN from Billing side due to a problem in the mediation system caused records not to be uploaded to the billing table on 10th-Dec. These datasets were generated to evaluate the ability of the approach to accurately identify root-causes for real alerts that result in revenue losses due to Mediation system problem caused by an upgrade on the mediation system. This issue was introduced by not introducing MSC, SMSC, and SASN transactions to the billing system table on the date of the problem.
- Major missing from different nodes and postpaid traffic drop on 1st- Nov due to error in CDRs generation at different nodes because of nodes configurations change. These datasets were generated to evaluate the ability of the approach to accurately identify root-causes for real alerts that result in revenue losses due to CDRs generation problem caused by a node configurations change. This issue was introduced by not generating MSC, SMSC, and SASN transactions for postpaid and prepaid subscribers and not introducing equivalent records to the billing system table on the date of the problem.
- Major increase in total prepaid voice calls count, duration, and charge on 31st-Dec due to a public event or holiday. This dataset was generated to evaluate the

ability of the approach to accurately identify root-causes for fake alerts that do not result in any revenue loss but they represent an extremely strange behavior due to a public event. This strange behavior was introduced by increasing the traffic percentage on the date of alert.

- Major increase in total prepaid calls charge with no additional major increase in calls count or calls duration due to a prices change starting from 24th-Dec. This dataset was generated to evaluate the ability of the approach to accurately identify root-causes for human errors that result in revenue losses and represent an extremely strange behavior due to a wrong prices change. This strange behavior was introduced by increasing the call minute price on the date of alert.
- Major decrease in calls charge due to an increase in total calls with zero charge for prepaid subscribers on 23rd-Nov because of wrong tariff configuration due to internal fraud. This dataset was generated to evaluate the ability of the approach to accurately identify root-causes for internal fraud cases that result in great and noticeable revenue leakages for voice calls traffic, and represent an extremely strange behavior due to a wrong prices change. This strange behavior was introduced by changing call minute price to zero on the date of alert.
- Major decrease in total SMSs charge due to an increase in total SMSs with zero charge for prepaid subscribers on 7th-Nov because of wrong tariff configuration due to internal fraud. This dataset was generated to evaluate the ability of the approach to accurately identify root-causes for internal fraud cases that result in great and noticeable revenue leakages for SMS traffic, and represent an extremely strange behavior due to a wrong prices change. This strange behavior was introduced by changing SMS price to zero on the date of alert.

- Major decrease in total prepaid voice calls charge and duration in units starting from 24th -Dec, while the total prepaid voice calls duration in seconds and count follows their normal pattern without any major decrease in traffic. This behavior was due to a new product launch for rounding units of voice calls to 90 seconds instead of 60 seconds for the same price. This dataset was generated to evaluate the ability of the approach to accurately identify root-causes for bad product design cases that result in quality issues indicating unreal revenue leakage issues, and represent an extremely strange behavior due to a rounding criteria change. This strange behavior was introduced by changing the calls rounding criteria on the date of alert.
- The launch of conflicting campaigns for international traffic that affect the estimated revenues for each other. This dataset was generated to evaluate the ability of the approach to accurately identify root-causes for bad design for conflicting products cases result in financial issues indicating revenue leakage issues, and represent a strange behavior. This strange behavior was introduced by reducing the products traffic percentage on the date of alert.
- Great GPRS traffic increase and total charges decrease due to external fraud using fraudulent application. This dataset was generated to evaluate the ability of the approach to accurately identify root-causes for external fraud cases that result in great and noticeable revenue leakages for GPRS traffic, and represent an extremely strange behavior due to the use of fraudulent application. This strange behavior was introduced by changing GPRS traffic to the initiation APN that is zero charged.

In this context, we are interested in studying the cases that generate alerts to detect possible leakage issues and identify all of their possible root-causes and to make

sure that the datasets that do not generate alerts as they do not have any revenue leakage issues within, will not be analysed to identify any root-causes.

Therefore, Table 2 illustrates revenue leakage issues used in this study, their type, description, root-cause instances and symptoms. While table 3 represents different datasets used per revenue leakage issue and different dates of the issue per datasets. For example, fourth scenario represents a CCN disconnection problem due SDP site power failure root-cause, causing five possible symptoms to occur, each of these symptoms was studied separately from the others on 10 different datasets to assure the proposed model sufficiency, completeness, accuracy, and correctness to show the potential of the proposed approach and its plausibility.

Scenario	Type	Description	Root-cause	Symptom
First	FP	Major duration increase	New Offer	Great increase in MSC trending
Second	FP	Major traffic increase	Public holiday	Great increase in CCN trending
Third	TP	MSC disconnection	Switch upgrade	Great voice calls missing from CCN
Fourth	TP	CCN disconnection	SDP site power failure	Great voice calls missing from CCN
				Great SMSs missing from CCN
				Great GPRS sessions missing from CCN
				Mismatched voice calls duration between MSC and CCN
				Mismatched GPRS sessions volume between SASN and CCN
Fifth	TP	Problem in the mediation system	System upgrade	Great voice calls missing from Billing system
				Great SMSs missing from Billing system
				Great GPRS sessions missing from Billing system
Sixth	TP	Error in CDRS generation in (MSC, SMSC, or SASN)	node configurations change	Great records missing from MSC side (MSC vs. CCN)
				Great records missing from SMSC side (SMSC vs. CCN)
				Great records missing from SASN side (SASN vs. CCN)
				Great drop in voice calls traffic from MSC side
				Great drop in SMS traffic from SMSC side
				Great drop in GPRS traffic from SASN side
				Great drop in postpaid voice calls traffic from Billing side
				Great drop in postpaid SMS traffic from Billing side
				Great drop in postpaid GPRS traffic from Billing side
Seventh	TP	Voice calls price change-Human error	Prices change by an employee	Great increase in total voice call charges
Eighth	FP	Rounding criteria change-Poor product design	New product	Great decrease in voice calls units and total charges

Ninth	TP	Conflicting campaigns-Poor product design	Poor product	Great product's international traffic decrease
Tenth	TP	Voice calls price change to zero-Internal fraud	Prices change by an employee	Great voice calls traffic increase and total charges decrease
Eleventh	TP	SMSs price change to zero-Internal fraud	Prices change by an employee	Great SMS traffic increase and total charges decrease
Twelfth	TP	GPRS initiation traffic-External fraud	Fraudulent mobile application	Great GPRS traffic increase and total charges decrease
Thirteenth	TN	No leakage issue	No leakage issue	Normal trends and no missing records, should not raise any alert and should not be investigated.

Table 2 Revenue leakage issues

Scenario	Starting date of the problem									
	Date1	Date2	Date3	Date4	Date5	Date6	Date7	Date8	Date9	Date10
First	31-Oct	7-Nov	8-Nov	14-Nov	21-Nov	28-Nov	5-Dec	12-Dec	19-Dec	26-Dec
Second	5-Nov	6-Nov	12-Nov	19-Nov	26-Nov	3-Dec	10-Dec	17-Dec	24-Dec	31-Dec
Third	31-Oct	7-Nov	8-Nov	14-Nov	21-Nov	28-Nov	5-Dec	12-Dec	19-Dec	26-Dec
Fourth	31-Oct	7-Nov	8-Nov	14-Nov	21-Nov	28-Nov	5-Dec	12-Dec	19-Dec	26-Dec
Fifth	5-Nov	6-Nov	12-Nov	19-Nov	26-Nov	3-Dec	10-Dec	17-Dec	24-Dec	31-Dec
Sixth	1-Nov	8-Nov	15-Nov	22-Nov	29-Nov	5-Dec	13-Dec	20-Dec	26-Dec	27-Dec
Seventh	4-Nov	5-Nov	12-Nov	18-Nov	19-Nov	26-Nov	3-Dec	10-Dec	17-Dec	24-Dec
Eighth	4-Nov	5-Nov	12-Nov	18-Nov	19-Nov	26-Nov	3-Dec	10-Dec	17-Dec	24-Dec
Ninth	4-Nov	5-Nov	12-Nov	18-Nov	19-Nov	26-Nov	3-Dec	10-Dec	17-Dec	24-Dec
Tenth	2-Nov	9-Nov	19-Nov	23-Nov	30-Nov	6-Dec	7-Dec	14-Dec	21-Dec	28-Dec
Eleventh	2-Nov	9-Nov	19-Nov	23-Nov	30-Nov	6-Dec	7-Dec	14-Dec	21-Dec	28-Dec
Twelfth	3-Nov	10-Nov	17-Nov	24-Nov	1-Dec	8-Dec	15-Dec	16-Dec	22-Dec	29-Dec
Thirteenth	26 datasets with no leakage issue were used on all revenue leakage issues symptoms checks used previously to assure that no alert and no root-cause analysis will be held.									

Table 3 Datasets used for evaluation

6.4 *Results and discussion*

This section aims to show the results of introducing the generated datasets to the provenance based developed model with respect to completeness and accuracy evaluation metrics.

Table 4 illustrates the calculation of completeness and accuracy evaluation metric per symptom for the total of 10 datasets used for evaluation, except for the last case TN that used 1 dataset per symptom with a total of 26 datasets.

Scenario	Symptom	Type	Predicted				Correctly predicted	Completeness	Accuracy
			TP	FP	TN	FN		#Predicted/#Alerts	#Correctly Predicted/ #Predicted
First	Great increase in MSC trending	FP	0	10	0	0	10	100%	100%
Second	Great increase in CCN trending	FP	0	10	0	0	10	100%	100%
Third	Great voice calls missing from CCN	TP	10	0	0	0	10	100%	100%
Fourth	Great voice calls missing from CCN	TP	10	0	0	0	10	100%	100%
	Great SMSs missing from CCN	TP	10	0	0	0	10	100%	100%
	Great GPRS sessions missing from CCN	TP	10	0	0	0	10	100%	100%
	Mismatched voice calls duration between MSC and CCN	TP	10	0	0	0	10	100%	100%
	Mismatched GPRS sessions volume between SASN and CCN	TP	10	0	0	0	10	100%	100%
Fifth	Great voice calls missing from Billing system	TP	10	0	0	0	10	100%	100%
	Great SMSs missing from Billing system	TP	10	0	0	0	10	100%	100%
	Great GPRS sessions missing from Billing system	TP	10	0	0	0	10	100%	100%

Sixth	Great records missing from MSC side (MSC vs. CCN)	TP	10	0	0	0	10	100%	100%
	Great records missing from SMSC side (SMSC vs. CCN)	TP	10	0	0	0	10	100%	100%
	Great records missing from SASN side (SASN vs. CCN)	TP	10	0	0	0	10	100%	100%
	Great drop in voice calls traffic from MSC side	TP	10	0	0	0	10	100%	100%
	Great drop in SMS traffic from SMSC side	TP	10	0	0	0	10	100%	100%
	Great drop in GPRS traffic from SASN side	TP	10	0	0	0	10	100%	100%
	Great drop in postpaid voice calls traffic from Billing side	TP	10	0	0	0	10	100%	100%
	Great drop in postpaid SMS traffic from Billing side	TP	10	0	0	0	10	100%	100%
	Great drop in postpaid GPRS traffic from Billing side	TP	10	0	0	0	10	100%	100%
Seventh	Great increase in total voice call charges	TP	10	0	0	0	10	100%	100%
Eighth	Great decrease in voice calls units and total charges	FP	0	10	0	0	10	100%	100%
Ninth	Great product's international traffic decrease	TP	10	0	0	0	10	100%	100%
Tenth	Great voice calls traffic increase and total charges decrease	TP	10	0	0	0	10	100%	100%
Eleventh	Great SMS traffic increase and total charges decrease	TP	10	0	0	0	10	100%	100%
Twelfth	Great GPRS traffic increase and total charges decrease	TP	10	0	0	0	10	100%	100%

Thirteenth	No alert, No Problem	TN	0	0	26	0	0	100%	100%

Table 4 Completeness and Accuracy evaluation per symptom

Completeness evaluation was done by dividing the number of identified revenue leakage issues per symptom per type root-causes over the number of revenue leakage issues datasets introduced to the developed model.

Accuracy evaluation was done by dividing the number of correctly identified revenue leakage issues per symptom per type root-causes over the number of correctly identified revenue leakage issues per symptom per type root-causes of the leakage issues datasets introduced to the developed model.

The model has proven its completeness and accuracy depending on the results shown in table 4, as it achieves 100% completeness and accuracy for the evaluated root-cause instances. On the other hand, the model depends to a high level on the accuracy of contextual information represented in the tables, so it needs to assure tables data accuracy to be accurate in root-cause analysis. For example to predict that an offer for prepaid subscribers was launched at the day of the revenue leakage issue, it should be correctly added and represented in the offers table for prepaid subscribers not postpaid subscribers. Including more data elements and classes will be a goal for improving the current proposed approach. Moreover, the model lacks the involvement of agent nodes to be representative of the user responsible for some actions and activities, and it does only support usage assurance and rating assurance, but lacks the ability to include subscription assurance, interconnect assurance, and partner assurance. Furthermore, it does not support RA functions on subscriber basis, depending on subscriber information. But, these limitations represent a starting point for future enhancement for the model.

6.5 *Threats to validity*

The evaluation was conducted on the defined processes, scenarios, and root causes that were systematically identified by the author, and thus consequently the approach was developed to address these identified cases, thus achieving high completeness and accuracy. However, the authors realize that identified cases or scenarios may not necessarily be conclusive and due to the nature and complexity of the RA function, other cases or scenarios may exist, that may have not been identified by the author. In this work the authors studied the scenarios and root causes of interest that could be measured accurately and have a strong relation to the research problem. It is worth noting that as shown in the implementation chapters, RA detective processes follow nearly a similar workflow, and which gives an indication that our results may be generalized to other RA processes, scenarios and root causes.

Chapter 7 Conclusions

This chapter concludes the thesis. A brief description of literature review, proposed model, contributions, results, limitations and future work is presented here.

7.1 *Introduction*

In our literature review, we focused on current approaches for leakage detection including big data analytics and rule-based systems. The main drawback of these approaches that they are mainly human driven in the investigation process and do not support drill down and root-cause analysis features.

Chapter 4 proposed a new approach for revenue leakage detection that supports drill down and root-cause analysis in telecom industry based on data provenance approach using semantic and contextual information related to the main revenue leakage detection processes and main telecom network nodes.

The rest of this chapter presents the conclusion including contributions, summary of the results, limitations and assumptions related to the proposed approach and potential future work.

7.2 *Contributions*

The main contribution of this novel approach is the development of an approach that supports root-cause analysis and drill down capabilities in current rule-based RA systems. To achieve this we have followed the below steps:

- The development of a rule-based RA system model that supports backward tracing consisting of critical revenue assurance functions on large datasets for testing and evaluation purposes.

- The development of a query model able to generate a data workflow for critical revenue assurance functions. The workflow can automatically be generated once the function is executed, consisting of nodes representing source nodes and SQL processes, and links representing the relationships between the nodes.
- The development of a Provenance model. The model consists of information and the relationships as mapped attributes and used filters. Additional nodes, such as incidents, logs, offers, and public events, are connected to both source nodes and results nodes to enrich the provenance information.
- The development of an alerting mechanism where once a threshold is reached the system generates an alert with possible reasons and additional information, a data workflow graph explaining how the data has changed and tables of all affected data at each step to give the analyst the ability to drill down among the raw data.

7.3 *Results*

The evaluation evaluated the completeness and correctness of our approach as below:

1. Completeness test was done to insure that our proposed approach could perform root-cause analysis and debugging for all different use cases that generate alerts. In order to evaluate the completeness of our approach, we use a gold standard test with 12 revenue leakage issues (with their root causes) and 26 possible revenue leakage symptoms as a gold standard discussed in the implementation chapter. Completeness evaluation was done by dividing the number of

identified revenue leakage issues per symptom per type root-causes over the number of revenue leakage issues datasets introduced to the developed model.

2. Accuracy (Correctness) test was done to evaluate how many of the predicted root-causes for detected cases were correct, whether these cases were real leakage issues or just fake alerts indicating an abnormal behavior. In order to evaluate the accuracy and correctness of our approach, we use a gold standard test with 12 revenue leakage issues (with their root causes) and 26 possible revenue leakage symptoms as a gold standard discussed in the implementation chapter. Accuracy evaluation was done by dividing the number of correctly identified revenue leakage issues per symptom per type root-causes over the number of correctly identified revenue leakage issues per symptom per type root-causes of the leakage issues datasets introduced to the developed model.

The model has proven its completeness and accuracy as it achieves 100% completeness and accuracy for the evaluated root-cause instances. On the other hand, the model depends to a high level on the accuracy of contextual information represented in the tables, so it needs to assure tables data accuracy to be accurate in root-cause analysis. For example to predict that an offer for prepaid subscribers was launched at the day of the revenue leakage issue, it should be correctly added and represented in the offers table for prepaid subscribers not postpaid subscribers. And it needs to be further enhanced to involve agent nodes and include subscription assurance and interconnect assurance as business cases with the involvement of subscribers' information.

7.4 *Limitations and assumptions*

This section covers some of the assumptions and limitations of our work:

- The model depends to a high level on the accuracy of contextual information represented in the tables, so it needs to assure tables data accuracy to be accurate in root cause analysis, for example to predict that an offer for prepaid subscribers was launched at the day of the revenue leakage issue, it should be correctly added and represented in the offers table for prepaid subscribers not postpaid subscribers.
- The model lacks the involvement of agent nodes to be representative of the user responsible for some actions and activities.
- The current model does only support usage assurance and rating assurance, but lacks the ability to include subscription assurance, and interconnect assurance.
- The current model does not support RA functions on subscriber basis, depending on subscriber information.

7.5 *Future work*

This section presents the main areas for future work as below:

- The involvement of agent nodes in the provenance model to be representative of the user responsible for some actions and activities for auditing and accountability purposes.
- Include subscriber database in the approach to enhance current RA functions that work on subscriber basis to support drill-down and root-cause analysis.
- Include RA functions related to subscription assurance, interconnect assurance, and partner assurance in enhanced approach.

References

[1] Adler, S. C., Curbera, F. P., Doganata, Y. N., Li, C. S., Martens, A., McAuliffe, K. P., ... & Slominski, A. A. (2008). U.S. Patent Application No. 12/265,986.

[2] A powerful and effective answer to revenue leakage. http://www.ey.com/Publication/vwLUAssets/Revenue_leakage/$FILE/A5_revenue_EN.pdf. Retrieved Apr 22, 2018.

[3] Baamann, Katharina. "Data Quality Aspects Of Revenue Assurance." ICIQ. 2007.

[4] Belhajjame, K., Deus, H., Garijo, D., Klyne, G., Missier, P., Soiland-Reyes, S., & Zednik, S. (2012). Prov model primer. URL: http://www. w3. org/TR/provprimer.

[5] Buneman, P., Khanna, S., & Tan, W. C. (2001, January). Why and where: A characterization of data provenance. In ICDT (Vol. 1, pp. 316-330).

[6] Business rules management systems. https://www.infoworld.com/article/2665148/techology-business/business-rules-management-systems.html. Retrieved Jan 07, 2018.

[7] Chapman, A., & Jagadish, H. V. (2009, June). Why not?. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data (pp. 523-534). ACM.

[8] Che, P., Bu, Z., Hou, R., & Shi, X. (2008). Auditing Revenue Assurance Information Systems for Telecom Operators. Research and Practical Issues of Enterprise Information Systems II, 1597-1602.

[9] Cheney, J., Chiticariu, L., & Tan, W. C. (2009). Provenance in databases: Why, how, and where. Foundations and Trends® in Databases, 1(4), 379-474.

[10] Chiticariu, L., Tan, W. C., & Vijayvargiya, G. (2005, June). DBNotes: a post-it system for relational databases based on provenance. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data (pp. 942-944). ACM.

[11] Curbera, F., Doganata, Y., Martens, A., Mukhi, N. K., & Slominski, A. (2008, November). Business provenance–a technology to increase traceability of end-to-end operations. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 100-119). Springer, Berlin, Heidelberg.

[12] Davidson, S. B., Boulakia, S. C., Eyal, A., Ludäscher, B., McPhillips, T. M., Bowers, S., ... & Freire, J. (2007). Provenance in scientific workflow systems. IEEE Data Eng. Bull., 30(4), 44-50.

[13] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.

[14] e Silva, A. C. (2011). Metrics for evaluating performance in document analysis: application to tables. International Journal on Document Analysis and Recognition (IJDAR), 14(1), 101-109.

[15] Fine, J., Deshong, E., Lim, M. J., Ailene, K. I. M., LeGro, E., & Kumar, S. (2010). U.S. Patent No. 7,720,759. Washington, DC: U.S. Patent and Trademark Office.

[16] Foster, I., Vockler, J., Wilde, M., & Zhao, Y. (2002). Chimera: A virtual data system for representing, querying, and automating data derivation. In Scientific and Statistical Database Management, 2002. Proceedings. 14th International Conference on (pp. 37-46). IEEE.

[17] Fraud and revenue assurance in telecoms overview. https://www.techzim.co.zw/2012/03/fraud-and-revenue-assurance-in-telecoms-overview/. Retrieved Apr 22, 2018.

[18] Global revenue assurance survey 2013. http://www.ey.com/Publication/vwLUAssets/Global_telecoms_revenue_assurance_survey_2013/$FILE/Global_revenue_assurance_survey_2013.pdf Retrieved Jan 07, 2018.

[19] Greenwood, M., Goble, C., Stevens, R., Zhao, J., Addis, M., Marvin, D., ... & Oinn, T. (2003). Provenance of e-science experiments-experience from bioinformatics.

[20] GSM protocol analysis. https://www.gl.com/gsm-network-protocol-analyzer.html. Retrieved Jan 07, 2018.

[21] Gupta, M., Norton, N., & D'souza, A. (2013). U.S. Patent Application No. 13/973,044.

[22] Hadoop. http://hadoop.apache.org/ .Retrieved Jan 07, 2018.

[23] Harris, S., Seaborne, A., & Prud'hommeaux, E. (2013). SPARQL 1.1 query language. W3C recommendation, 21(10).

[24] Hospital billing optimizer. https://www.beckershospitalreview.com/pdfs/white-papers/Opera_Solutions_WP_Hospital_Billing.pdf. Retrieved Jan 07, 2018.

[25] How to develop a CDR generator. https://paul.kinlan.me/how-to-develop-a-cdr-generator/ .Retrieved Jan 07, 2018.

[26] Husted, N., Quresi, S., & Gehani, A. (2013, April). Android Provenance: Diagnosing Device Disorders. In TaPP.

[27] Ikeda, R., Cho, J., Fang, C., Salihoglu, S., Torikai, S., & Widom, J. (2012, April). Provenance-based debugging and drill-down in data-oriented workflows. In Data Engineering (ICDE), 2012 IEEE 28th International Conference on (pp. 1249-1252). IEEE.

[28] Imran, M., & Hlavacs, H. (2012, July). Provenance in the cloud: Why and how. In The Third International Conference on Cloud Computing, GRIDs, and Virtualization (pp. 106-112).

[29] Interlandi, M., Shah, K., Tetali, S. D., Gulzar, M. A., Yoo, S., Kim, M., ... & Condie, T. (2015). Titian: Data provenance support in spark. Proceedings of the VLDB Endowment, 9(3), 216-227.

[30] Kim, Y., Lee, H., & Perrig, A. (2014). Information security applications: 14th International Workshop, WISA 2013, Jeju Island, Korea, August 19-21, 2013: Revised selected papers. Heidelberg: Springer.

[31] Logothetis, D., De, S., & Yocum, K. (2013, October). Scalable lineage capture for debugging disc analytics. In Proceedings of the 4th annual Symposium on Cloud Computing (p. 17). ACM.

[32] Luc Moreau and Paolo Missier et al. PROV-DM: The PROV Data Model. Online. Retrieved: 29.08.2016. 2013. url: https://www.w3.org/TR/prov-dm/.

[33] Mattison, R. (2005). The telco revenue assurance handbook. Lulu. com.

[34] Mattison, R. (2009). A Framework for Assessing Revenue Assurance Capabilities.

[35] Mattison, R. (2009). A Financial Framework for Revenue Assurance Decision-Making.

[36] Mattison, R. (2009). Revenue Assurance Standards.

[37] Miller, J. J. (2013, March). Graph database applications and concepts with neo4j. In Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA (Vol. 2324, p. 36).

[38] Missier, P., Belhajjame, K., & Cheney, J. (2013, March). The W3C PROV family of specifications for modelling provenance metadata. In Proceedings of the 16th International Conference on Extending Database Technology (pp. 773-776). ACM.

[39] Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., ... & Plale, B. (2011). The open provenance model core specification (v1. 1). Future generation computer systems, 27(6), 743-756.

[40] Moreau, L., Groth, P., Miles, S., Vazquez-Salceda, J., Ibbotson, J., Jiang, S., ... & Varga, L. (2008). The provenance of electronic data. Communications of the ACM, 51(4), 52-58.

[41] Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., ... & Plale, B. (2011). The open provenance model core specification (v1. 1). Future generation computer systems, 27(6), 743-756.

[42] Park, H., Ikeda, R., & Widom, J. (2011). Ramp: A system for capturing and tracing provenance in mapreduce workflows.

[43] PROV model primer. https://www.researchgate.net/profile/Stian_Soiland-Reyes/publication/236237923_PROV_Model_Primer/links/59de44ca0f7e9bcfab23f701/PROV-Model-Primer.pdf. Retrieved Jan 07, 2018.

[44] Ram, S., & Liu, J. (2006, November). Understanding the semantics of data provenance to support active conceptual modeling. In International Workshop on Active Conceputal Modeling of Learning (pp. 17-29). Springer, Berlin, Heidelberg.

[45] Revenue assurance how to stop bleeding and start leading. https://clarity.sutherlandglobal.com/blog/accounting-minute/revenue-assurance-how-to-stop-bleeding-and-start-leading/. Retrieved Jan 07, 2018.

[46] Revenue leakage overview. http://rafm360.blogspot.com/2013/11/revenue-leakage-overview.html. Retrieved Jan 07, 2018

[47] Rozsnyai, S., Slominski, A., & Doganata, Y. (2011, July). Large-scale distributed storage system for business provenance. In Cloud Computing (CLOUD), 2011 IEEE International Conference on (pp. 516-524). IEEE.

[48] Schouten, P. (2013). Big data in health care: solving provider revenue leakage with advanced analytics. Healthcare Financial Management, 67(2), 40-43.

[49] Simmhan, Y. L., Plale, B., & Gannon, D. (2005). A survey of data provenance in e-science. ACM Sigmod Record, 34(3), 31-36.

[50] Small, N. (2015). The Py2neo 2.0 Handbook—Py2neo 2.0. 7 documentation.[Online]. Retrieved July 10, 2015.

[51] Tech trends in telecommunications. https://www.ariasystems.com/blog/5-tech-trends-telecommunications/. Retrieved Jan 07, 2018.

[52] Townend, P., Webster, D., Venters, C. C., Dimitrova, V., Djemame, K., Lau, L., ... & Taylor, N. (2013, March). Personalised provenance reasoning models and risk assessment in business systems: A case study. In Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on (pp. 329-334). IEEE.

[53] GPRS Core Network. (n.d.). In Wikipedia. Retrieved Jan 07, 2018, from https://en.wikipedia.org/wiki/GPRS_core_network.

[54] Network Switching Subsystem. (n.d.). In Wikipedia. Retrieved Jan 07, 2018, from https://en.wikipedia.org/wiki/Network_switching_subsystem.

[55] Online charging system. (n.d.). In Wikipedia. Retrieved Jan 07, 2018, from https://en.wikipedia.org/wiki/Online_charging_system.

[56] Provenance. (n.d.). In Wikipedia. Retrieved Jan 07, 2018, from https://en.wikipedia.org/wiki/Provenance.

[57] Revenue Assurance. (n.d.). In Wikipedia. Retrieved Jan 07, 2018, from https://en.wikipedia.org/wiki/Revenue_assurance.

[58] Short Message service center. (n.d.). In Wikipedia. Retrieved Jan 07, 2018, from https://en.wikipedia.org/wiki/Short_Message_service_center.

[59] Telecommunications billing. (n.d.). In Wikipedia. Retrieved Jan 07, 2018, from https://en.wikipedia.org/wiki/Telecommunications_billing.

[60] Zoldi, S. M., & Balon, M. P. (2014). U.S. Patent No. 8,824,648. Washington, DC: U.S. Patent and Trademark Office.

Appendix

Dataset generation process using CDR generation tool

The original version of the CDR generation tool was obtained from the following url link (https://paul.kinlan.me/call-detail-record-cdr-generation-tool/). The updated version, used in this study, its description, command line usage, API usage, and configuration options can be found here (https://github.com/mayconbordin/cdr-gen/). To generate the dataset, 1) the parameters are given values in the configurations file, see figure 100; 2) run the CDRGen using the command prompt using MVN commands, see figure 101. The datasets will then be generated in Excel files, see figure 102.

```
{
    "callsMade": {
        "mean": 20,
        "stdDev": 5
    },
    "incomingCalls": {
        "mean": 250,
        "stdDev": 5
    },
    "numAccounts": 40000,
    "startDate": "29/12/2017",
    "endDate": "30/12/2017",
    "callTypes": ["Free", "Local", "National", "Intl", "PRS", "Mobile"],
    "dayDistribution": {
        "sun": 0.185535,
        "mon": 0.18222,
        "tue": 0.18222,
        "wed": 0.18222,
        "thu": 0.18222,
        "fri": 0.0428,
        "sat": 0.0428
    },
    "offPeakTimePeriod": {
        "start": "18:00",
        "end": "08:00"
    },
    "outgoingCallParams": {
        "Free": {
            "callCost"     : 0,
            "callDur"      : 5,
            "callStdDev"   : 5,
            "callStdDev2"  : 5,
            "callOPCost"   : 0,
            "callOPDur"    : 5
```

Figure 100 Configuration file

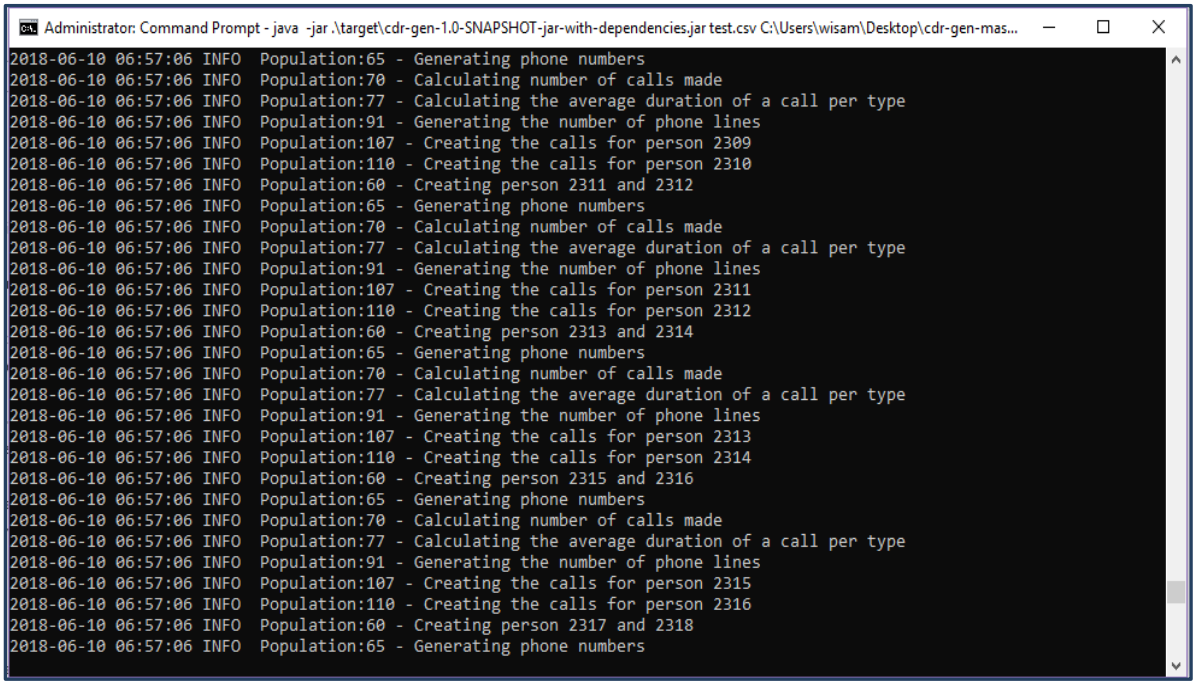


Figure 101 CDRGen in command prompt

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	ID	A#	Lines	B#	Sdate	Edate	Stime	Etime	Type	Cost	SDTM	EDTM	Time Difference	Call Duration (Seconds)	Call Duration (Minutes)
2	0	1866111111	1	7836422222	12/10/2017	12/10/2017	16:05:00	16:11:00	Mobile	150	12/10/17 16:05:00	12/10/17 16:11:00	0:06:00	360	6
3	1	1866111111	0	1867777777	12/9/2017	12/9/2017	19:30:00	19:47:00	Local	68	12/09/17 19:30:00	12/09/17 19:47:00	0:17:00	1020	17
4	2	1866111111	0	1867777777	12/6/2017	12/6/2017	18:55:00	19:12:00	Local	68	12/06/17 18:55:00	12/06/17 19:12:00	0:17:00	1020	17
5	3	1866111111	0	1867777777	12/9/2017	12/9/2017	23:05:00	23:16:00	Local	44	12/09/17 23:05:00	12/09/17 23:16:00	0:11:00	660	11
6	4	1866111111	1	8456777777	12/7/2017	12/7/2017	19:30:00	19:34:00	National	32	12/07/17 19:30:00	12/07/17 19:34:00	0:04:00	240	4
7	5	1866111111	1	1867777777	12/4/2017	12/4/2017	12:00:00	12:05:00	Local	20	12/04/17 12:00:00	12/04/17 12:05:00	0:05:00	300	5
8	6	1866111111	0	1867777777	12/7/2017	12/7/2017	5:55:00	6:06:00	Local	44	12/07/17 05:55:00	12/07/17 06:06:00	0:11:00	660	11
9	7	1866111111	1	8456777777	12/6/2017	12/6/2017	16:20:00	16:33:00	National	104	12/06/17 16:20:00	12/06/17 16:33:00	0:13:00	780	13
10	8	1866111111	0	1867777777	12/6/2017	12/6/2017	20:25:00	20:45:00	Local	80	12/06/17 20:25:00	12/06/17 20:45:00	0:20:00	1200	20
11	9	1866111111	1	1867777777	12/5/2017	12/5/2017	23:05:00	23:25:00	Local	80	12/05/17 23:05:00	12/05/17 23:25:00	0:20:00	1200	20
12	10	1866111111	0	8456777777	12/7/2017	12/7/2017	21:45:00	21:58:00	National	104	12/07/17 21:45:00	12/07/17 21:58:00	0:13:00	780	13
13	11	1866111111	1	1867777777	12/10/2017	12/10/2017	15:25:00	15:57:00	Local	128	12/10/17 15:25:00	12/10/17 15:57:00	0:32:00	1920	32
14	12	1866111111	0	7836422222	12/6/2017	12/6/2017	18:00:00	18:10:00	Mobile	250	12/06/17 18:00:00	12/06/17 18:10:00	0:10:00	600	10
15	13	1866111111	1	1867777777	12/7/2017	12/7/2017	18:10:00	18:34:00	Local	96	12/07/17 18:10:00	12/07/17 18:34:00	0:24:00	1440	24
16	14	1866111111	1	1867777777	12/7/2017	12/7/2017	18:40:00	19:04:00	Local	96	12/07/17 18:40:00	12/07/17 19:04:00	0:24:00	1440	24
17	15	1866111111	0	1867777777	12/4/2017	12/4/2017	17:50:00	18:34:00	Local	176	12/04/17 17:50:00	12/04/17 18:34:00	0:44:00	2640	44
18	16	1866111111	1	7836422222	12/7/2017	12/7/2017	20:50:00	21:04:00	Mobile	350	12/07/17 20:50:00	12/07/17 21:04:00	0:14:00	840	14
19	17	1866111111	0	22822222222	12/5/2017	12/5/2017	18:50:00	19:18:00	Intl	840	12/05/17 18:50:00	12/05/17 19:18:00	0:28:00	1680	28
20	18	1866111111	1	1867777777	12/7/2017	12/7/2017	1:35:00	1:44:00	Local	36	12/07/17 01:35:00	12/07/17 01:44:00	0:09:00	540	9
21	19	1866111111	0	1867777777	12/6/2017	12/6/2017	17:40:00	17:49:00	Local	36	12/06/17 17:40:00	12/06/17 17:49:00	0:09:00	540	9
22	20	1866111111	1	7836422222	12/6/2017	12/6/2017	19:25:00	19:25:00	Mobile	0	12/06/17 19:25:00	12/06/17 19:25:00	0:00:00	0	0
23	21	1866111111	1	1867222222	12/4/2017	12/4/2017	19:20:00	19:47:00	Local	108	12/04/17 19:20:00	12/04/17 19:47:00	0:27:00	1620	27

Figure 102 Final generated dataset