Master's thesis

Master's Programme in Computer Science

Automatic Detection of Mass Outages in Radio Access Networks

Milla Lintunen

March 13, 2023

Faculty of Science

University of Helsinki

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Faculty of Science	Master's Programme in Computer Science

Tekijä — Författare — Author

Milla Lintunen

Työn nimi — Arbetets titel — Title

Automatic Detection of Mass Outages in Radio Access Networks

Ohjaajat — Handledare — Supervisors

Dr. Gopika Premsankar, Dr. Ashwin Rao, Prof. Sasu Tarkoma

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	March 13, 2023	61 pages

Tiivistelmä — Referat — Abstract

Fault management in mobile networks is required for detecting, analysing, and fixing problems appearing in the mobile network. When a large problem appears in the mobile network, multiple alarms are generated from the network elements. Traditionally Network Operations Center (NOC) process the reported failures, create trouble tickets for problems, and perform a root cause analysis. However, alarms do not reveal the root cause of the failure, and the correlation of alarms is often complicated to determine. If the network operator can correlate alarms and manage clustered groups of alarms instead of separate ones, it saves costs, preserves the availability of the mobile network, and improves the quality of service. Operators may have several electricity providers and the network topology is not correlated with the electricity topology. Additionally, network sites and other network elements are not evenly distributed across the network. Hence, we investigate the suitability of a density-based clustering methods to detect mass outages and perform alarm correlation to reduce the amount of created trouble tickets. This thesis focuses on assisting the root cause analysis and detecting correlated power and transmission failures in the mobile network. We implement a Mass Outage Detection Service and form a custom density-based algorithm. Our service performs alarm correlation and creates clusters of possible power and transmission mass outage alarms. We have filed a patent application based on the work done in this thesis. Our results show that we are able to detect mass outages in real time from the data streams. The results also show that detected clusters reduce the number of created trouble tickets and help reduce of the costs of running the network. The number of trouble tickets decreases by 4.7-9.3% for the alarms we process in the service in the tested networks. When we consider only alarms included in the mass outage groups, the reduction is over 75%. Therefore continuing to use, test, and develop implemented Mass Outage Detection Service is beneficial for operators and automated NOC.

ACM Computing Classification System (CCS)
Networks → Network types → Mobile networks
Information systems → Information systems applications → Spatial-temporal systems → Data streaming

Avainsanat — Nyckelord — Keywords

alarm correlation, fault management, mobile networks, spatio-temporal clustering

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Software study track

Acknowledgements

I would like to thank my Elisa supervisor Henri Tenhunen, whose knowledge helped me move forward and whose explanations clarified many complex issues. I would also like to thank my other supervisor Jaakko Rautiainen from Elisa for his support and advice. I thank my supervisors Ashwin Rao, Gopika Premsankar and Sasu Tarkoma who have given me valuable feedback throughout this process. Also many thanks to my family, friends and colleagues who have supported and encouraged me.

Contents

1 Introduction

Mobile Radio Access Networks (RAN) are facing ever-growing challenges in their fault management due to the increased number of alarms [27] and the growing complexity of heterogeneous networks consisting of base stations with different-sized cells [7]. Faults constantly occur in the network, and alarms are generated to inform network operators of such failures [1]. The performance of the RAN is a key factor for a good user experience. Mobile network can have degraded quality of service or it is not available at all and problems appearing in the mobile network correspond to the number of user complaints [39]. When a failure is detected, a large number of alarms can be generated from multiple different network elements within a short period of time. The failure can be significant and require actions to correct network behavior. It is also possible that the failure is self-correcting but causes alarms [20]. Alarms are sent as messages to describe a failure type, but network elements have very limited knowledge of the network and the cause of the error [8].

As shown in Figure 1.1, the alarm generated by a network element propagates through a set of intermediaries before arriving at the Network Operations Center (NOC). Each network element records the alarms and they are reported to a Element Management System (EMS). The EMS communicates higher to a Network Management System (NMS). The NMS must have the capability to display inventory and alarm data and manage multiple networks. The NMS receives alarms from detected events, and alarms are handled in the NOC where trouble tickets are created for issues that require further actions [27]. The NOC can receive multiple alarms from different base stations and other network elements and create a trouble ticket for each of them. However, in the case of large power outages, some of these alarms can be a part of the same incident and have the same root cause.

Fault management in telecommunication includes detection and analysis of the issues together with fixing network problems. Several studies [27, 23, 8, 24, 7, 20, 43, 41, 21, 16] have researched alarm correlation and automatic methods to discover rules for fault management. Many different approaches and techniques have been introduced to provide better service quality and improve fault detection. Still, multiple alarms and trouble tickets burden the NOC and field maintenance as manually analyzing an extensive number of generated alarms is time-consuming [39]. Thus, the development of new correlation

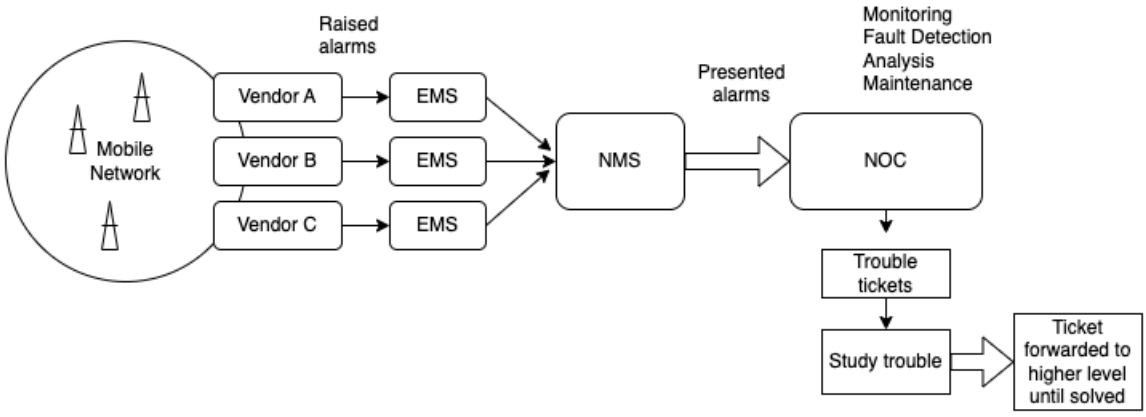


Figure 1.1: Alarms arriving from each vendor in the mobile network is reported to the Element Management System (EMS) and forwarded to the Network Management System (NMS). The NMS transmits alarms to the Network Operations Center (NOC), where the network failures are handled.

solutions for fault management is important to reduce the costs of running the network, improve network quality, minimize service degradation, and enhance fault resolution.

A network can have failures due to many different reasons including transmission, power, and configuration failures. Power failures and power outages occur regularly, as a consequence of storms [40], some other weather conditions (for example high wind), natural disasters [45], or simply human errors [40]. Operators have usually multiple electricity providers and the electricity network topology is not correlated with the mobile network topology. Additionally, power distribution is usually decentralized between sites. Transmission alarms can be detected using topology information, but in the case of power-related problems, location information of the network elements is more essential. Modeling and analyzing power failures is difficult, as they evolve over time in a complicated manner and spread randomly [45].

In this thesis, we consider that a mass outage occurs when several alarms of a similar type arrive from multiple network elements during a short period of time presumably because of the same failure. We consider alarm types that indicate more severe power or transmission problems. When a mass outage occurs, a single failure disrupts the smooth operation of the different nodes affecting the network's ability to carry traffic. Mass outages also appear in different types and in this thesis we focus on transmission mass outages and power mass outages. As an example of a mass outage, let's consider a scenario where a problem in the power distribution causes multiple alarms to be generated. In such circumstances, the NOC might create multiple trouble tickets. While a relevant power alarm is received,

NOC must begin to process the failure, create a trouble ticket and solve the root cause of the alarm to perform sufficient actions. The NOC is equipped with complex monitoring systems to manage alarm flow. The generated alarms need to be analyzed so that decisions for the trouble tickets can be performed. Relationships and dependencies of other relevant alarms and network elements may need to be collected manually during investigation [1]. If dependencies are missed, trouble tickets may be created for the sites separately, even if faulty alarming sites would contain the same root cause. Investigation of separate tickets consumes time and increases costs. The field maintenance technicians can be sent to multiple sites if problems are not detected as having the same root cause. Technicians can travel large distances which result in unnecessary costs [29].

The objective of this thesis is to automatically detect possible correlated power and transmission alarms and define mass outages from the alarm stream. An automated process will help resolve the failures faster by minimizing the number of trouble tickets and providing information on correlated alarms for faster decision-making for the operator. We want to get direct information on alarms that have the same root cause. To the best of our knowledge, existing literature and research do not present solutions for receiving a stream of alarms associated with cell sites, where each group of alarms is associated with a particular type of alarm. We propose a solution where each cell site in a group is located within a threshold distance from at least one of the cell sites in a group. We have filed a patent application based on the work done in this thesis.

We design and implement a Mass Outage Detection Service to be used as well in Virtual NOC product at Elisa Polystar [39]. We focus on assessing how correlated alarms can be detected, and analyze selected approaches' ability to detect power or transmission mass outages in real-time in the radio access network. Concerning power mass outages, a major challenge is that operators have different electricity providers. Correlating alarms are attached to a site and particular sites may not be direct neighbors. Still, there can be found a significant location and time relation between alarms. An additional challenge is caused by the need for real-time detection from continuous data streams. Data and situations evolve continuously and analysis must be done in real-time taking into account limitations in terms of memory and time.

Density-based clustering algorithms group data points that have high density. Density-based clustering detects clusters without a predefined number of clusters, detects outliers, and can find arbitrarily shaped clusters. We propose a method using our custom density-based algorithm designed specially to cluster correlated alarming sites to detect mass

outages from continuous data streams in real-time. In our case, current algorithms as such are not directly suitable for predicting mass failures because the need is online and the algorithm must be fast enough when handling large amounts of data. There is also a need for specific steps, that current density-based algorithms do not have. We introduce a new clustering method that sets a unique search radius for each site. The value of the search radius may be calculated based on a configured number (k) of neighboring cell sites and the scaling factor. A new detection method is proposed in the form of Proof of Concept (PoC) implementation, referred to as Mass Outage Detection Service, that is used to detect power or transmission mass outages online with the network automation product Virtual NOC. The implementation is written with Kotlin programming language and it uses the PostgreSQL database with PostGIS extension. Apache Kafka is used for real-time data streaming.

The implemented service is deployed to two real networks and to a test network to monitor the alarm stream and detect possible mass outages. We evaluate the performance of the Mass Outage Detection Service and the ability to detect mass outages in two real mobile networks and in the Test Network. We generate the dataset for the Test Network and perform testing with different parameter settings. Our results show that we are able to detect power and transmission mass outages from real mobile network data streams in real time. We note that calculating the search radius with neighboring sites (kNN) 5 with multiplication factors 3 to 4 and kNN 10 with multiplication factors 2 to 3 provide the most promising results. The ability to find mass outage groups creates value for the operator by reducing the number of trouble tickets and assisting with root cause analysis. We detect several mass outage clusters in the real networks. Considering the alarms in the mass outage groups alone, we are able to reduce the number of tickets in these two real networks by 75.21% and 82.76%. Considering all power- and transmission-related alarms, the reduction in the number of tickets is between 4.74 and 9.25%.

The chapters of the thesis are organized as follows. Chapter 2 provides required background of mobile networks, focusing on RAN, alarm enrichment, processing data, and alarm correlation and fault management. Chapter 3 discusses data stream clustering methods, focusing on density-based clustering, providing background of benefits and weaknesses using clustering and unsupervised learning methods. Chapter 4 describes the data analysis methods, research hypotheses, our developer algorithms, and implementation of the Mass Outage Detection Service. Chapter 5 discusses the results and evaluates the performance of the implementation. Chapter 6 draws a conclusion and discusses the future work.

2 Fault management in mobile networks

Fault management is one part of mobile network management. Fault management is responsible for the detection, diagnosis, isolation, and resolution of problems occurring in the mobile network. Traditionally fault management and troubleshooting are manual processes done by radio access network experts. Concerning the different network management functionalities, fault management is the most demanding considering labor, time usage, and demand for experts [7]. Thousands of alarms occur every day and alarms arrive from different network elements from different regions.

Generated alarms are handled in Network Operations Center (NOC) where alarms are monitored constantly. The NOC receives thousands of less relevant alarms per day as only some contain essential information [1]. The NOC needs to collect and aggregate alarms belonging to the same incident and create a trouble ticket with a priority to handle the problem [23]. Different definitions are used concerning concepts in the field of network fault management. Some specifications of key definitions are defined as they are used in this thesis as follows:

- *Fault* is kind of a problem. A fault can be permanent or intermittent. Fault occurrence is informed as alarms for example in the form of a message such as SNMP trap. A single fault may cause multiple alarms. One example of a fault is power outage of a connectivity loss.
- *Error* is discrepancy between an assumed to be correct condition and an observed condition. A fault may cause multiple errors but may not need to be directly corrected and it can be invisible. An example of an error is when IP packets are discarded in a router because of bad/malformed headers.
- *Event* is an occurrence of an exceptional condition at the certain time in the managed network. There can be fault events informing of the start of a problem and clear events notifying of the end of a problem.
- *Alarms* are information of the faults generated by network elements which are transmitted to the NOC. An alarm can contain multiple different states and events.

Alarms may originate via management protocol messages such as SNMP trap as a notification.

- *Alarm correlation* is the process of finding relationship and correlations between alarms and grouping alarms which refer to the same problem [35, 41].

Resolution time of the faults is crucial, especially with the alarms impacting the usability of the network and end users as it has the biggest financial effects on the service provider [24]. Time spent on resolving the root cause, and separately solving problems that might be correlated consumes resources and money.

2.1 Mobile networks and RAN

Contemporary mobile networks consist of radio access networks (RAN) and core networks (CN). The RAN implements radio access technology and its functionalities include managing radio resources, channel allocation, and data rate. The RAN consists of radio base stations. A radio unit (RU) processes and transmits signals to the base station. The RU communicates with a baseband unit (BBU) using Common Public Radio Interface (CPRI) and signals are processed to be forwarded to the core network as seen in Figure 2.1. In 5G, the BBU is separated into a central unit (CU) and a distributed unit (DU). The RAN provides access and connects user equipment (UE) to the network through radio connections and coordinates the management of resources across the radio sites. The base station has the responsibility to connect to UE so that mobile endpoints can communicate with the rest of the network. The CN is responsible for routing, user authentication, high-level traffic aggregation, and call control/switching [24] [31]. The traffic from sites is carried by transmission (or transport) network. Transmission network can be divided into core (backbone) network, access (backhaul) network, and regional network but division can differ between operators [6].

The cell site is the location where the mobile operator installs a radio base station. Coverage, capacity, and throughput provided by the base station vary. Different types of base stations (macro, micro, pico, or femto) have two main component blocks. The first block has a cooling system and a microwave link. The second block consists of a power amplifier (PA), a transceiver (TX), and a digital signal processing (DSP) which are presented in Figure 2.2 [33]. The radio base station is powered by electricity. If the base station is without a power source, it will not be able to carry traffic. An alarm is triggered when

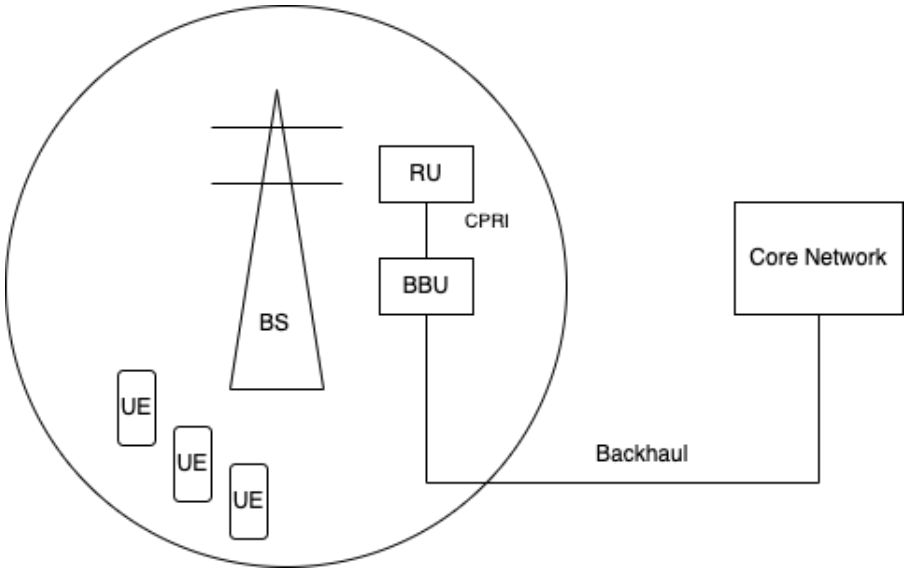


Figure 2.1: Traditional architecture of the RAN. The circle denotes the RAN where the base station (BS) connects to the user equipment (UE). Signals are transmitted through the radio unit (RU) and the baseband unit (BBU) using Common Public Radio Interface (CPRI). Traffic to the core network (CN) is carried by Backhaul, which is a part of the transmission network.

a power failure is detected. Each network element records the alarms and the alarms are stored centrally. The alarms are reported to the Element Management System (EMS) when a device informs about a fault.

A mobile network consists of several interconnected areas called cells. One site can have one or multiple cells. Cell sites are powered by electricity and they have various signal ranges. Users are handed over from one cell to the nearest available cell when they are moving. This allows the user to stay connected. A cell represents the coverage area that the operator provides and high-power base stations transmit signals at longer ranges. Cells do not receive or transmit signals in all directions. Signals are transmitted in some specific direction and usually rural areas have larger cells and less densely located base stations than cities [13]. Implementations of base station components vary and in practice, most base stations are split into multiple nodes interconnected by different interfaces. Network interfaces often run on transport networks consisting of various technologies. Multiple-input and multiple-output (MIMO) is a technology that uses multiple transmitters and receivers to transfer more data simultaneously. 5G Massive MIMO delivers a significant increase in network capacity, throughput, and coverage. With massive MIMO, users can expect a high data rate even on the cell's edge [31].

The transmission (or transport) network in mobile network transfers the traffic from cell

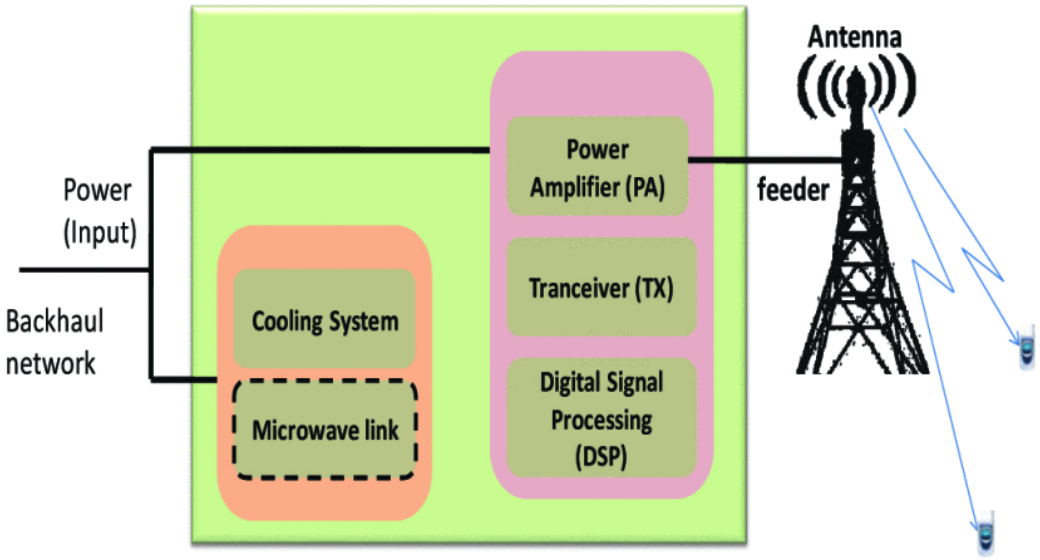


Figure 2.2: The macrocell base station with two main component blocks. The first block has a cooling system and a microwave link. The second block consists of a power amplifier (PA), a transceiver (TX), and a digital signal processing (DSP) [33].

sites forward connecting RAN and CN. As mentioned, the transmission network can be divided into a core (backbone) network, an access (backhaul) network, and a regional network. The backhaul network connects base stations to one or more traffic aggregation sites and delivers connectivity to the core network. Transport technologies differ and the decision of the technology selection is affected by multiple factors such as distances between backhaul and core transport networks. Also, variations of transport distances vary depending on spectrum availability and cell site density among other things [6].

Using the complexity and costs of both the development and deployment of the network as a metric, the biggest component is the Radio Access Network (RAN), which consists of base stations being responsible for providing coverage and connections to users. A number of the base stations is usually high and RAN results in most of the costs of running and deploying the mobile network. The RAN uses radio frequencies to provide wireless connectivity to the user equipment. Radio access technology differs among base stations, as some provide perhaps 5G and 4G and some only 3G [31]. Differences can be also found by provided

coverage, capacity, and throughput, ranging from macro with larger coverage to micro base stations with more limited coverage and smaller energy consumption. Responsibilities in the RAN include also strategies and algorithms for controlling power, channel allocation, and data rate [24].

2.2 Network Operations Center

A Network Operations Center (NOC) controls and monitors the mobile network from one or more locations. The NOC handles emerging faults across the network or only on some layers of the network such as RAN. Traditionally running the NOC requires labor for monitoring and analyzing faults around the clock. Running the NOC is crucial for the operation and management of the network and multiple tasks increase operational costs (OPEX). The NOC's responsibility is to manage the time used reacting to network alarms and problems. Additionally, the NOC is responsible for the time used for required site visits by mechanics. Total OPEX can rise to 40% of the total expenses running the network. Multiple activities performed in the NOC contribute to OPEX, such as time spent reacting to the network events and usage of time for routine tasks [39].

Multiple levels of support exist in the NOC and one way to handle network faults is to categorize support into 3 levels. The thesis work is based on division as described. In the first level (tier 1), professionals monitor the alarms and analyze problems to define the root cause. Most of the alarms are handled at this level. Second-level (tier 2) technicians are assigned to problems if issues cannot be solved on the first level. Tier 2 manages more complicated tasks, including but not limited to system restarting and installation problems, which need a deeper understanding of the network. Issues that require very deep knowledge need third-level (tier 3) support. Tier 3 has specialized technicians and support can be handled in-house or outsourced with subcontract [36].

During fault detection and the decision process of trouble ticket creation, NOC experts need to combine different information, query databases, and find dependencies between alarms. Even though experts possess years of experience, the process is error-prone and time-consuming. To handle faults adequately, the NOC has to create trouble tickets from specific alarms and situations. Understanding the relations of the received alarms in the NOC is a difficult task for experts because of the complexity of the network and complex interactions between the network elements [27].

2.3 Improving the network

The number of alarms generated from the network increases with network growth. One major goal for operators is to reduce maintenance costs. Other important issues are the improvement of the network quality, predicting future incidents, and improving fault resolution while reducing operational costs. Usually, in the ideal cases, only one trouble ticket for field maintenance from one high-level failure would be created by the NOC. For example, when malfunctioning equipment causes correlating alarms, the operator will benefit if only one ticket is created instead of multiple ones.

The occurrence of a fault in the radio access network will decrease the quality of service and customer satisfaction is compromised. It is also important to notice effect on service impact and for example areas, where the effect of the fault might be more severe. The health of the RAN layer has a direct effect on quality of service (QoS) and customers [7]. In addition to a huge number of alarms and the complexity of the network, the difficulty of solving the fault is caused by the huge number of network elements and their geographical distribution. Automating the troubleshooting process will have multiple benefits such as decreased downtime, shorter time identifying the problem, better QoS, and reduced maintenance costs [24].

2.3.1 Fault management automation

To achieve high performance and quality of service, operators need to detect and diagnose network problems fast while handling an excessive number of alarms. There is a huge interest in automating the detection and diagnosis of RAN problems. The radio access network problems are one of the hardest to diagnose due to multiple different vendors, several network elements, and the complicated correlations and details [19].

The need to identify optimal areas to introduce automation exists in the network and for most operators RAN constructs a large number of the operational costs. Handling RAN-related faults need constant monitoring of the alarms and possible occurring errors. For example, the NOC must quickly detect and handle situations when an alarm is generated due to the cell being out of service. Multiple possible actions might be required, from creating trouble tickets and remote reset to the need for a field visit, where costs rise significantly. Faults need to be reacted to quickly, but during problem resolution, several different procedures might need to be attempted. With automation, corresponding actions

can be triggered by the event and manual intervention can be eliminated or reduced [39].

2.3.2 Virtual NOC

Elisa Polystar has developed an automation system, Virtual NOC [28], for RAN fault management aiming to eventually achieve zero-touch network automation. With the automated solution, service providers are able to automate, capture, enrich, and perform root cause analysis of the alarms and handle faults in tier 1, open and update trouble tickets and solve them automatically. Trouble tickets can be created per cell site, even though in some cases there exists correlation. Our implemented Mass Outage Detection Service is aimed to be a part of the Virtual NOC product to address this issue.

Automation handles most network faults and all multivendor network alarms can be tracked, filtered, and processed after analysis. Virtual NOC does several types of actions based on tracked records of the network. Some incidents still require field visits, but unnecessary visits are reduced with the solution. For most operators, RAN causes the largest number of operational costs. The main benefits of automating NOC are to reduce the cost of running the network and minimize the service effect. These lead to design points, one of which is to minimize the number of trouble tickets created by the NOC [39]. With our Mass Outage Detection Service, correlated power alarms can be detected and information can be used to create only one trouble ticket instead of multiple separate ones for each site.

2.4 Alarm enrichment and processing

Alarm data needs a lot of processing beforehand to extract relevant information from the content. Simple Network Management Protocol (SNMP) is a widely used industry-standard protocol for network management. SNMP system consists of four key components: Network Management System (NMS), SNMP agent, managed object, and management information base (MIB). The NMS manages the network providing different functionalities. One of them is fault management which has the responsibility of detecting and fixing faults [24]. Managed devices contain an SNMP agent and the NMS interacts with the agent which executes operations on the managed device. SNMP agent sends notification messages called SNMP traps to inform the NSM of different events generated by the device. MIB has a tree structure and every node in the tree represents the managed

object. Object Identifier (OID) consists of numbers (e.g., 1.3.6.1.4.1.2682.1) started from the root of the tree and it uniquely identifies managed objects and are used to navigate through variables. OID and the value information can be found in SNMP traps and for example own OID exists for the type field of the alarm. OIDs can be mapped to relevant alarm fields [17].

Alarms contain several information fields in the raw data records, such as time of occurrence, unique identifier, type of alarm, managed object name, and extended information. Typically most of the alarm fields are empty or do not provide any useful information. Figure 2.3 describes part of the alarm attributes presented in Huawei alarms providing relevant information. Similar fields can be extracted from other vendors' alarms. RAN alarms do not contain explicit site information but site name can be parsed and extracted as a part of certain alarm fields [7]. In this manner, site name information is available in the alarm data and with the help of the site location data, it is possible to map every triggered alarm to a certain site. The site is always serving some geographical area and the size of the area depends on related cells.

Alarms include probable cause information where operators might be able to receive some information about what happened in a specific network element. Alarms do not usually reveal the actual root cause of the failure but they inform that there is a fault. The message field can contain hints about the reasons for a failure, but a network element in the error state has a very limited view of the network [8].

The duration of the alarm being active is the time between the start and end event. The start event is sent when the network element is in the error state. The end event clearing the alarm is sent when the malfunction ends in the network element. Alarms known as flapping alarms or self-solving alarms appear only for a short period of time and are cleared quickly. Network malfunction may not be permanent and end within seconds and clear the generated alarm [20]. For example, a problem in a cable connection can occur and notify a malfunction and activate an alarm. A cable can appear as functional after seconds and clear the alarm. After a while the problem can be repeated multiple times generating new alarms.

Self-solving alarms and flapping alarms can be detected by keeping alarms in a queue for a certain time period, and classifying them as solved with certain parameters. When a notification of a clear alarm is sent and detected, the alarm is counted as solved [20]. Alarms need to have all relevant information available and for example, alarms missing the occurrence time information usually are ignored [27]. Also scheduled maintenance work is

causing alarms to trigger, as some of the network elements might be needed to turn off for work time. These alarms will clear when maintenance work is done [20]. It is important to note that domain expert knowledge is essential when extracting alarm content or sorting out the most relevant alarms [1].

Field name	Example value	Explanation of the field
alarmCSN	1234123	Indicates the alarm sequence number that uniquely identifies an alarm.
alarmCategory	1	1 = fault 2 = clear 3 = event 4 = ack 5 = unack 9 = change Especially 1 is raise-alarm, 2 clear-alarm, 3 are random events and 4–5 are manual intervention.
alarmOccurTime	2021-03-29 17:29:51	gmt. Occurred, not current(trap) time
alarmMOName	ABC_123	managed object name, e.g., OSS, controller, pico serial identifier and base identifier
alarmNEDevID	NE=987	the id of the device where alarm is generated.
alarmID	12121	alarm unique identifier, e.g. 301
alarmType	2	1 Power 2 Environment 3 Signaling 4 Trunk 5 hw 6 sw 7 running sys 8 comm sys 9 QoS 10 Processing error 11 OC 12 Integrity 13 Operational 14 Physical 15 Security 16 Time domain
alarmLevel	1	1 = Critical 2 = Major 3 = Minor 4 = Warning 5 = Indeterminate 6 = Cleared
alarmAckTime		Acknowledgement time
alarmRestoreTime		Clearance time
alarmProbablecause	S1 Interface Reset	"identifies the cause"
alarmAddionalInfo	RAT_INFO=GUL, AFFECTED_RAT=GUL, DID=NULL	RAT's (Radio Access Technologies) G/SM U/MTS L/TE
alarmExtendInfo	Cabinet No.=0, Subrack No.=0, Slot No.=19, Port No.=1, Board Type=AAA	Location information about an alarm

Figure 2.3: Example fields of a Huawei alarm. Left column has the name of the field, middle column has an example value of the field, and the third column gives an explanation of the field and possible values. Note that only a subset of fields occurring in an alarm are presented here [17] .

The same type of alarms originating from the same network element within a short time period have the same root cause with a high probability [1]. Usually, some specific type of alarms indicates provided power disappearance of the site. Relevant alarm selection for detecting power outages requires RAN expert knowledge and can be also noticed for example with alarm pattern recognition [40].

2.5 Relation discovery and alarm correlation

Currently, to the best of our knowledge, no published research on detecting correlated power or transmission type of alarms between sites exists. There exists related alarm correlation methods and research [7, 8, 23, 24, 27, 43] which we utilize to gather information for the analysis and implementation for our Mass Outage Detection Service. We develop an alarm correlation method that can be used with real-time data streams and with multiple different operators. We try to achieve finding correlated sites which contain relevant active alarms and form a mass outage. Understanding how network elements and problems are related is crucial as it offers the possibility to correlate alarms and form clustered groups to handle failures instead of processing each of them separately. Spatial and temporal aspects to cluster network elements can be used to achieve automatic relation discovery between network nodes. Hidden relations between network elements can be found using sequential pattern mining and detecting network node relations [23].

Using graphs to find correlations, the site can be seen as a vertex and the next wireless hop or other traffic between sites can be used as an edge. This kind of method can reveal hidden topological properties and is especially useful when determining transmission-related faults. Also, connectivity of the neighbors and the density of the local area can be evaluated with the help of graphs [7]. Network graph analysis offers important insights for alarm data mining and rule discovery. A graph can be obtained for different levels, for network problems, or network nodes. It can reveal unknown relationships or dependencies and the mutual effects of the network elements. Trying to find relations manually requires a lot of time and effort in addition to expert knowledge [23]. Also, connectivity of the neighbors and the density of the local area can be evaluated with the help of graphs [7].

One much-researched approach for correlating alarms or investigating the availability of the network is through the use of topology information [27, 24]. Several alarm correlation methods use topology as a part of the solution trying to discover relations in the network. Topology knowledge and mapping alarms to certain network elements provide a possibil-

ity to find relevant alarm correlation rules and relations between network elements. From the alarm logs, it is possible to extract network topology following how information has transited through the network. A path can be created as an ordered list of the network elements where the alarm message was triggered. Correlation rules can be found within a single device, between devices in the same domain (eg. RAN), and also between devices in different domains. However, the method used by the Fournier-Viger et al. [27] requires information on many domain-level alarms, including microwave and router, and cannot be used only within the RAN domain. Topology is not always already available and data is needed from many domain levels to build it. It is notable, that alarms in different domains (layers) in the hierarchical network have very different types of alarms [27]. However, topological information changes constantly with the rapid evolution of the network hardware and configuration policies and topology knowledge is not fully reliable [8].

Most features in alarm logs are categorical which indicates that applying clustering algorithms is difficult based on only alarm data lacking distance measurements. Alarms generated from closely located network elements occurring closely in time indicate possible major conditions even though logs typically contain only coarse timestamps. Alarm patterns can be detected in time and space, for example, in a time window containing alarms and clustering network elements based on geographic coordinates. Interesting situations can be found with a spatial correlation which can be confirmed after domain expert analysis [43]. From extracted attributes of the alarm, alarm occurrence time is a major correlation factor. When an alarm occurrence time is used as a clustering attribute and hypotheses are formed with the identified clusters, the hypotheses can be further introduced to domain experts [24]. Often alarms can occur at the same time and be highly correlated and clusters can represent these sorts of groups. Relevant clusters provide information for NOC to offer more essential knowledge of the correlating alarms and help them solve the fault.

2.6 Network failures

In a mobile network, an outage occurs for multiple reasons. Large outages can affect the networks ability to carry traffic, for example, due to a problem in the power distribution or transmission network. Concerning disaster scenarios, power failure is the most dominant failure. Physical damage failures and power failures account for more than 99% of all

network failures among disaster-occurred failures [45]. In such cases, for the network operators and NOC handling alarms and creating trouble tickets, more value is provided if we offer information on mass outages and which possible alarms in certain areas might be caused by the same incident.

With a site outage (Figure 2.4), none of the cells transmit signals and it will affect the ability of the network to carry traffic. When more sites located closely have an outage, there are even more coverage holes. In the case of power mass outages, detecting is a challenge, as operators do not own the electricity network and power distribution might be decentralized between sites. As the density of the base stations increases, power network failure will affect even more sites. Cells can be overlapping, especially in the city area, when sites have different frequencies. If an outage concerns part of the sites, service is still available with other frequencies but the quality of service level is decreased. Power failures need to be recognized as early as possible with correlated sites. To achieve high availability for the network, it is essential to have very short network downtime [29]. The same type of alarms originating from the same network element within a short time period have the same root cause with a high probability [1]. Usually, some specific types of alarms inform provided power disappearance of the site. Relevant alarm selection for detecting power outages requires RAN expert knowledge and can be also noticed for example with alarm pattern recognition [40].

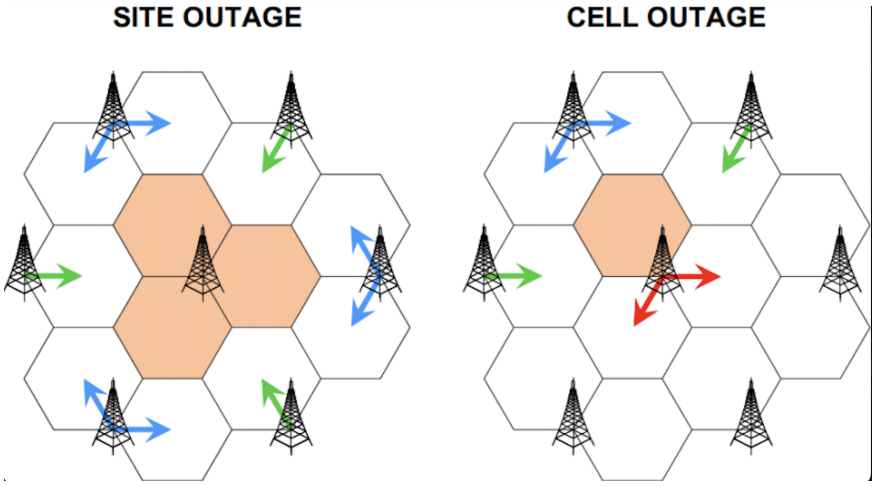


Figure 2.4: Site outage and cell outage. Outage cells are marked as beige color. If the whole site has an outage, there may exist coverage holes or the level of service quality is decreased [4].

Severe weather, human errors, or vandalism can cause large power outages which affect base stations in the area. Most base stations have battery banks installed as backup power. Since power outage duration can be long, large power outages can cause service

interruptions and in rural areas, outages can be frequent [40]. Different base stations of the same operator can also use different power distributors. Even if some power distribution issues occur for one distributor, some base stations use a different distributors and are not affected. This poses an additional challenge for power mass outage detection.

The majority of the base stations are equipped with batteries and when a base station loses power, it will automatically switch using backup power although, there can be a quite large variation in how long backup power will last [45]. The NOC is informed of the power failure in form of alarms when the power of the base station is lost. When the NOC receives an alarm informing of lost power, it indicates the base station has begun to use backup power. The subsequent alarms can occur when for example battery power is beginning to be low. Power outage detection is mandatory for operators as providing service depends on electricity. Outages need to be detected fast so that there is a possibility to restore the connections and the whole system. Actions include correcting the cause of the failure or for example providing more backup power to the site. If faults are noticed only after users report the problem, it has already affected more service quality and customer experience affecting possible revenue losses [29].

Power failure is one of the biggest causes of disrupted base stations. There are geographical areas where the possibility to lose all network coverage is increased [45]. When a cell site has a full outage, none of the cells of that site is in operation. Detecting power outages in the mobile network is done by gathering measurements, such as alarms that inform of a certain kind of failure. For example, when a power failure occurs, an alarm is generated that reports that the electricity has been cut [4]. Storm can spread randomly and faults arise slowly moving from one area to another damaging infrastructure along the way. Analyzing complicated evolving storm situations is very difficult, and disaster-related failure incidents and their relation to geographical features have been scarcely researched in radio access networks [45].

A transmission network consists of transmission lines and a critical failure in the transmission network affects multiple sites. Fiber optics network technology is widely used for transmission. Optical fiber has huge bandwidth and signals at different wavelengths do not interfere with each other [6]. A cut fiber affects negatively the network quality and connectivity. A damaged cable incident disrupts normal operations and causes network outages. Interruption in a connection between a cable and network element can be expected close to the locations where optical cables are deployed. When there is an impact on multiple sites, each site will send alarms of transmission type [15].

Knowledge of the network availability, and changes in the network spatially and temporally are crucial for network management. Network connectivity in a specific geographical area at a certain time after a natural disaster has been considered by L. Zhong et al. [45]. The authors criticize that some studies make ideal assumptions that network failures are spatially independent and networks are ideally distributed. For example, large weather disasters or power failures are highly correlated with geographical factors. For power outages in mobile networks, one important aspect is the base station's spatial distribution and location.

Base station location and density information are important to be taken into account detecting mass outages. In the city area, as cell size is much smaller than in rural areas, base stations are usually located close to each other. Regarding geographic location, we can define a network denser and distances with different sites smaller in the city area. A full site outage in 5 base stations has different effect for example in a large city, as multiple available connection points still may exist, than in the countryside, where that number of base station outages can cause a full loss of network coverage.

In this Chapter, we discussed fault management in the RAN and concepts related to the thesis. We showed the importance of improving and automating the development of correlation methods. We discussed alarm enrichment in more detail and introduced different alarm correlation methods that may benefit detecting mass outages. In the next Chapter, we will discuss different data stream clustering methods and their connection to data analysis and our Mass Outage Detection Service.

3 Data stream clustering and management

A Data Stream Management system (DSMS) is a system managing continuous data streams and executing continuous queries. The dataset is not static and a new situation in the dataset can change the results [12]. Any number of streams can enter the system. One key research decision for a Data Stream Management system is to decide on the query language and the data model. Additionally, results need to be processed while new data arrives continuously and the possible request for the system needs to be handled. Usually, all past data cannot be saved and memorized for the future and some data must be discarded [12]. Our Mass Outage Detection Service uses Apache Kafka for handling data streams and Postgres as a database to efficiently query data.

Clustering is an unsupervised machine learning method trying to assign unlabeled data points into homogeneous, meaningful groups called clusters. Alarms can be correlated and classified with pre-defined rules [20]. In our work, we classify alarms based on certain alarm features and fields such as alarm identification numbers. For forming groups that are defined as mass outages, there is a need for the classification of alarms and clustering of the groups. As the need for the Mass Outage Detection Service and detecting the power and transmission mass outages is in real-time, clustering and detection are needed online. Additional alarm filtering is done based on configurations. Our preliminary analysis investigates the suitability of density-based algorithms detecting mass outages. We analyze historical data and propose a method for detecting mass outages and assisting in the root cause analysis. The analysis uses also a Density-Based Spatial Clustering Algorithm with Noise (DBSCAN) and estimates the suitability to detect mass outages with a density-based approach.

We use density-based clustering due to its many features that suit the nature of the mobile network. In the mobile network, base stations are not evenly distributed and cell site density can be notably higher in urban areas. Transmission and power mass outage groups can form for example oblong shaped clusters and the ability of the density-based algorithm to recognize arbitrarily shaped groups is beneficial. A merit of the density-based clusters in our case is also that there is no need to specify the number of clusters

in advance. We additionally have sites, that will not belong to any cluster so we need the ability to identify outliers. We need to detect groups within a dataset that have large differences in densities and consider an algorithm taking into account the varying densities.

3.1 Data stream clustering

A data stream can be defined as data that is being produced incrementally over time so it is not available fully as a bounded dataset before the need for processing. Network fault management alarms from various network elements are a good example of a continuous data stream that needs to be collected and analyzed in the NOC. Certain type of data is periodically saved for offline analysis but to react in real-time to alarms requires continuous monitoring and analyzing of alarms while considering scaling and constraints of the computational resources. The information on evolving datasets is incomplete while performing clustering of the data streams. Data points are assumed to be presented one at a time and the previous state can affect the next [12].

A few of the major challenges in data stream clustering are memory limitations, limitations of the amount of queries, and the continuously evolving data. If the amount of data is large, data cannot be queried often. As dataset is not static and the data arrives continuously, clustering the data streams demand continuous monitoring. Most important considerations in clustering data streams are execution time and memory restrictions [14].

Spatial data mining attempts to discover patterns from large spatial databases containing geographical data. Spatiotemporal data mining discovers spatial and temporal relationships. Spatial clustering generally uses latitude and longitude clustering data values whereas spatiotemporal clustering introduces time dimension in addition to spatial data [5]. We use spatiotemporal clustering and consider both time dimension and location information.

In dynamic environments, the number of the clusters can change, and formed clusters in real-time data streams have to adapt to the change. Data stream clustering methods include Hierarchical stream-based, partitioning stream-based, Growing Neural Gas (GNG) based, Grid-based, and density-based methods [14]. As mentioned, our main focus is on density-based clustering methods which are discussed in more detail in the rest of this Chapter.

3.2 Density-based clustering

Density-based clustering can form arbitrarily shaped clusters. Some conventional clustering algorithms such as K-means require the number of clusters to be defined beforehand and they cannot be used for situations where the number of the clusters might vary. Density-based clustering methods are able to detect clusters without a predefined number of clusters. In addition, density-based methods also can specify outliers not belonging to any cluster [3]. Multiple density-based clustering methods, including well known Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, use single static density for discover clusters [11] but other methods for varying densities have also been developed [9]. In a mobile network, each of the cell sites serves cells from a single location. Coverage, capacity, and throughput provided by a site vary. Cell density and cell site density are therefore higher in urban areas and lower in rural areas.

3.2.1 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm based on connected regions with high density. It is developed for large spatial database clustering with noise and the density is defined based on the surrounding number of points called a neighborhood. The algorithm uses two parameters, the distance measure in the neighborhood (epsilon) and the number of the objects in the neighborhood (MinPts) [2]. DBSCAN has core points, which have at least *minPts* points within a distance r, and border points, which have at least one core point within a distance r. Other points are defined as noise [10]. A point x is directly density-reachable from the point c if the distance of x is within the defined epsilon and c is a core point. A point x is density-reachable from the point c if there exists a chain of points (p_1, p_2, p_3..., p_n) and p_1=x and p_n=c such that p_{i+1} is directly density-reachable from p_i. Connectivity defines whether data points belong to the same cluster and reachability determines whether data points can be accessed from another point. A point x is density-connected from a point c if a point o exists such that both x and c are density reachable from o. [10].

The parameters of DBSCAN are user-defined. The selection of values influences the created clusters. For instance, if the epsilon is too small, a part of relevant data will not be clustered but defined as noise. Too high value poses false cluster merges and includes possible noise. Epsilon selection should reflect the distances in the dataset [10]. DBSCAN

creates an epsilon radius surrounding each data point as illustrated in Figure 3.1. Points are classified as core, border, or noise. The core data points contain at least a minPts number of points within a radius. DBSCAN algorithm visits all the points in the data and considers points included in the same cluster if at least minPts points exist within a defined distance (epsilon). A point forms a cluster with all the points that are reachable from itself. Each cluster must contain at least one core point and border points will not try to reach new points.

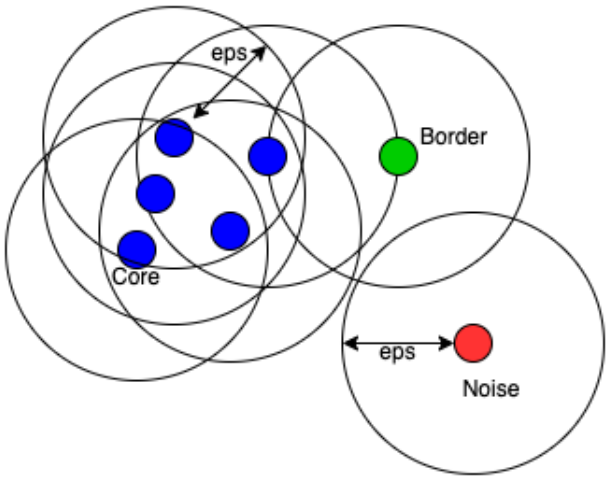


Figure 3.1: DBSCAN presentation of noise point (red), border point (green) and core points (blue) with minPts 3. The core data points contain at least a minPts number of points within a radius. The border point is included in the cluster but do not try to reach new points. The noise point is not included in the cluster.

3.2.2 Density-based stream clustering algorithms

DBSCAN is not suitable for stream clustering and we therefore use it for static analysis (see Section 4.4). A lot of density-based stream clustering algorithms have been developed, such as DenStream [9], OpticsStream [37], SOStream [18], D-Stream [38] and PKS-Streams [30]. Stream clustering algorithms must consider the data evolving and arriving continuously [2]. All density-based stream clustering algorithms have the ability to handle evolving data streams using window models such as fading [9, 18] and sliding [30, 38] window. In a sliding window model, data is considered in a specific time between the present and a set time value. In a fading window model each data point is given a weight and recent data is given more weight compared to older ones. A sliding window model is beneficial when only the most recent data is needed. As a disadvantage, it ignores part of the data stream.

Fading window model is able to consider older data and consider smaller effects for older data. The drawback is that model captures historical data and the size of the data grows with time. Our approach uses a sliding window model and includes alarms from a specific time range. We do not need to consider older alarms and want to define a certain time range to include alarms in the cluster. Density-based stream algorithms are used in many real-life applications such as environment observations, medical systems, social network analysis, and network intrusion detection systems [2].

The main challenge with density-based algorithms with data streams is the efficient use with space limitations, as the clustering can be computationally intensive with a large amounts of data. Multiple density-based stream clustering algorithms use online (summarizing data) and offline (creating actual clusters) phases and access data only once [14]. We perform all clustering online as we can limit the amount of data by filtering only certain alarm types and removing cleared alarms.

One well-known density-based stream clustering algorithm is a Density-based Clustering Over an Evolving Data Stream with Noise (Denstream) -algorithm which creates micro-clusters. Micro-clusters have summary information of the data and they compress the data effectively. Clusters are formed based on micro-clusters. Denstream uses associated weight which decreases exponentially with time. An incoming point is considered belonging to potential-micro-clusters or outlier-micro-cluster. The arriving point is tried to merge into the nearest clusters based on radius. In the offline phase, the final clusters are formed from potential micro-clusters with a variant of DBSCAN [9, 14]. DenStream recognizes the potential clusters from outliers but does not release memory space. Removing outliers is a time-consuming process in the algorithm [2].

Self Organizing Density-Based Clustering over Data Stream (SOStream) algorithm is based on self-organizing maps and DBSCAN. The algorithm is influenced by points neighborhood. SOStream creates clusters using dynamically learned threshold value using neighborhood information. The algorithm uses micro-clusters and a new point is added based on Euclidean distance if a dynamically set threshold is smaller than the distance to the micro-cluster [14]. Many stream clustering algorithms have an offline component which might not be a desired feature. SOStream has online operations to dynamically create, merge, and delete clusters [18]. In contrast to online, offline clustering is unable to cluster incrementally and clusters are updated only after receiving a batch of new data. Online clustering can adapt quickly to changing conditions. SOStream is able to adapt its thresholds to the data stream which can be seen merit to the algorithm. We use a similar

approach as SOStream defining individual density values for each site using neighboring information. We also use only online operations to dynamically form clusters.

3.3 Density estimation

Each alarm contains the name of a site. We attach each alarm to a site and thus to a certain location. Base stations in the network are not evenly distributed and using density-based clustering algorithms with constant density values would create too much noise or include noise points depending on the area. In a distance-based method, density is calculated with the number of points inside a neighborhood. Threshold radius r is used to define the neighborhood and the amount of data points are calculated inside the defined area.

Non-parametric density estimation is a technique from the field of statistics to estimate the probability density function of a random variable and it can be used with arbitrary distributions. One of the most common method for the non-parametric density estimation is k-Nearest Neighbor (kNN). In non-parametric statistics, there is no assumptions about the underlying distribution of the data. The kNN estimates the density by calculating the distance between the sample and the k-th nearest neighbor. [44]. In the case of the Mass Outage Detection Service, we use kNN to define a unique search radius for each site. Our analysis where we investigate the effect of different density calculations, use both kNN and calculation the number of data points in the area using different distances.

3.4 Clustering validation

One of the most difficult tasks in clustering is validation. We do not always know the ground truth or true labeling of the data. Clustering validation can be divided into external, internal, and relative criteria [25]. Internal validation criteria measure the quality of the clustering using only the data available during clustering. External clustering validation criteria approaches use information that is not available during clustering, such as external class labels. Results are evaluated based on a pre-defined structure. This can mean ground truth labels or beliefs of the clusters. Relative criteria compare clustering structure to other clustering schemes with different parameter values. Relative criteria do not involve statistical tests [25]. One of the most common internal criteria measures is Silhouette width. The other widely used internal evaluation metric criteria with a

density-based algorithm is the sum of squared distance (SSQ). Customarily used external evaluation quality metrics criteria used in clustering data streams are purity and Rand Index [3].

SSQ is used extensively in density-based data stream clustering when class labels of data are not available. SSQ determines the dispersion of the data points. Silhouette compares the similarity of a data point in its cluster to a neighboring cluster and computes a score between -1 and 1. Each object of a cluster is assigned a quantitative measure sw_i. The silhouette can be calculated with different distance metrics and similarity is measured based on distance. Let i be any data point in the dataset. The similarity of a data point i with the other data points in its cluster is computed as a_i. Let b_i be computed similarity of i with the other data points in other clusters [26].

The silhouette width for data point i is defined as:

$$sw_i = \frac{b_i - a_i}{max(a_i, b_i)}.$$

If the score is close to 1, the data point is closer to its own cluster. A score close to -1 score implies that clusters are assigned wrong [26].

As clustering is an unsupervised technique, it does not generally contain any ground truth labels. External criteria validation methods attempt to evaluate clustering results by creating labeling and assuming that the ground-truth clustering is known a priori [25, 42]. One possibility with external criteria is also to consult a domain expert whether the created clusters have been formed correctly. Rand Index measures the similarity between two data clusters. It receives a value between 0 and 1. It returns value 1 when clusters are the same. Value 0 means that clustering has clustered correctly none of the data points [9]. It measures the fraction of true positives and true negatives over all data points. Rand is defined as:

$$rand = \frac{TP + TN}{\binom{n}{2}}$$

Another external clustering validation method is a purity of clusters, where the proportion of data points that were correctly clustered are calculated given true labels of the data. Let $D = \{x_i\}_{i=1}^{n}$ dataset consist of n points partitioned into k clusters. Let $y_i = \{1, 2, ..., k\}$ denote ground-truth label information of the data points. Let $C = \{C_1, ..., C_r\}$ denote a clustering of the same dataset into r clusters. The ground-truth cluster T_j consists of all

the points with label j. The count n_{ij} denotes the data points that are common to both cluster C_i and T_j:

$$n_{ij} = |C_i \cap T_j|,$$

Purity is defined as:

$$purity = \frac{1}{n} \sum_{i=1}^{r} max_{i=j}^{k} \{n_{ij}\},$$

which is the weighted sum of the cluster-wise purity values [42].

We can measure the clustering algorithm performance also with testing for example execution time, memory usage, or sensitivity. Execution time is the total time used by the algorithm to process the data. Synthetic datasets can be used for the evaluation. Memory usage informs the amount of memory used by the algorithm. It can be measured based on real and synthetic datasets. Measurement is highly dependent on the used data structure. The sensitivity analysis reveals how the parameters of the algorithm affect the clustering quality and evaluates the best ranges for the parameters.

Since we use a synthetic dataset with known ground truth, we use external clustering validation to evaluate the performance of implemented Mass Outage Detection Service. In our case, we choose to use Rand Index as it considers both true positives and true negatives. With Rand Index, we can easily compare the usage of different kNN values. We also measure execution time in the real networks to verify enough efficiency in performing clustering online. We benefit from performing the sensitivity analysis and provide the possibility for operators to fine-tune parameter values.

4 Mass outage detection

In this Chapter, we form a definition for a mass outage. We perform alarm enrichment and extract site name, alarm occurrence time, alarm restore time, and other relevant information from the alarm data and merge the site location data. We then form research hypotheses, perform a data analysis, and detect situations defined as mass outages. We additionally describe our developed algorithms in Section 4.5. Section 4.6 describes our implementation as well as the flow of data through the service.

4.1 Mass outage definition

We are interested in detecting power and transmission failures that occur in more than n affected sites and are correlated with a high probability. At some unknown time, a large power or transmission incident occurs affecting a number of nodes in the network. Failures are informed in the form of alarms. We presume that the incident is dynamic, and it propagates along the network affecting more sites within a certain time window, especially in the case of power mass outages. Our goal is to design a density-based stream clustering algorithm to identify power and transmission failures. The proposed algorithm is sequential over a data stream as it has to monitor and have knowledge of the current situation. The objective is to detect clusters where more than n sites are affected and can grow with time. Additionally, we implement a Mass Outage Detection Service that can be used to detect mass outages in real-time and perform a root cause analysis.

First, we wish to estimate in the analysis which situations are defined as mass outages to form a better picture of the detection methods for the Mass Outage Detection Service. The motivation for finding mass outage clusters and correlating alarms stems from the need to minimize the number of created trouble tickets. We have formed definitions for mass outage as follows:

Definition 1 *Mass outage occurs when relevant alarms of the same type arrive from a large number of network elements within a certain short period of time and have the same root cause for failure with a high probability.*

Definition 2 *Correlated alarms in power or transmission outages cannot be located a*

long distance apart. The distance is dependent on the site density in the area. Other base stations without failure can be located in between base stations that belong to a mass outage.

Mass outages can occur with different types. Alarms that inform of transmission failure are used to detect transmission mass outages and power-related alarms can be used to detect power mass outages.

Definition 3 *The type of mass outage can be defined based on the type of the alarm and from the alarm identification number. This information is available directly in the alarm as stated in Section 2.4.*

To compare whether the situations contain mass outages, we use our definitions of the mass outage and investigate which alarm fields contain relevant information. While alarm data contains a huge amount of information, one challenge is that the alarms have a limited view of the overall network. The content of the fields is mostly categorical and does not offer good possibilities to cluster alarms. For this reason, as stated in Chapter 2, it is hard to define distance measures for applying clustering algorithms. Clustering just based on alarm type, alarm time and alarm category forms groups of alarms, that can have multiple different root causes. Another challenge arises from real-time detection. We have to maintain information on alarm states; is the alarm event still in a fault state or has the alarm been cleared. Real-time detection of the groups of sites, which contain active alarms having the same root cause needs to be detected from data streams and updated accordingly. In addition, we need to define, whether the groups will be defined as mass outages, is defined mass outage active and the state of the alarms in the mass outage. The end of a mass outage is achieved by monitoring, i.e., when all mass outage-related alarms have been cleared.

We can filter specific alarms based on alarm identification numbers before clustering. For example, in the case of power-related alarms implying that power is lost, alarms that have been active for over 1 minute, accounted for only a few percent of the total number of all alarms in the network in analysis data. Hence the amount of data to save and query from the database and to use in the clustering algorithm decreases as we need to store only relevant alarms for mass outage detection.

4.2 Extracting data and enriching alarm events

Before utilizing any information in the alarms, information is extracted from the incoming alarm. The information available inside the alarm event is unstructured and varies based on the vendor or alarm. Extraction rules for different fields differ and for example, the name of the site is found using different parsing rules for different types of alarms. Interesting fields for analysis of the alarm data in the context of mass outage detection are alarm category, alarm serial number (alarm CSN), alarm occurrence time, alarm managed object name (MO name), alarm id, alarm restore time, and alarm extend info. More detailed information on these fields was provided in Figure 2.3. We can monitor the state of the alarm from the event occurrence or clearance time (restore time). We also consider other possible relevant fields in the alarm data; alarm type, alarm additional info, alarm level, and probable cause. However, with the information provided merely by the alarm, reliable correlation conditions cannot be formed to define mass outages. When observing the occurrence time field, some of the alarms may be related, but we cannot eliminate the ones which are outliers only with the information provided by the alarm.

To determine whether alarms are related and have the same cause for failure, we also use geographical information. The same type of alarms which have the same root cause is expected to have significant location and time relation in both power and transmission-related faults. Despite the unstructured nature of some of the alarm fields, useful information can be obtained by parsing the alarm fields. We can extract site information in the alarms either from alarm MO name or alarm extend info with certain parsing rules. With this extracted information we may map each alarm to a specific site. We receive the location (latitude, longitude) of each site from the operator inventory data. Thus, each alarm may be mapped to a certain location.

4.3 Research hypotheses

As L. Zhong et al. [45] stated, power failures are correlated with geographical factors. They mention that power failures and physical damage account for over 99% of all total network failures during disaster scenarios. Motivated by the knowledge that the alarm occurrence time is a major correlation factor [4] and site location may have a relation to the same high-level power or transmission failure, an analysis is done based on temporal and location factors.

Challenges arise from the facts that network operators have different power distributors, and the power distribution network is not known to the mobile network operator. We may have groups of sites containing power failures which could be defined as power mass outages and appear within the defined radius. There may also exist sites without any issues within the same area. Therefore, we cannot form groups of sites only based on nearest neighbors. We cannot assume that all mass outage alarms belonging to the same group would be always directly connected if we consider only for example 10 nearest neighbors. Still, we can utilize time and location correlation and density-based clustering algorithms. As network elements are installed in a certain location, the failure has usually a spatial connection to the network elements. As stated in Chapter 3, a cell site density is higher in urban areas and lower in rural areas and density-based clustering methods have characteristics suitable for detecting mass outages. In our case, we use adaptable density-based clustering for detecting clusters with varying densities.

Hypothesis 1 *Mass outages can be identified by clustering the alarms according to temporal and spatial factors.*

Our Mass Outage Detection Service should be able to recognize both power and transmission mass outages. As both have geographical and time correlation between the sites, we make a hypothesis in the case of mass outages that:

Hypothesis 2 *Different types (power and transmission) of mass outages can be detected with the same method and algorithm, just using a different sets of relevant Alarm IDs.*

Addressing the described problems include the recognition of relevant features for the detection of correlated power outages from the data that is available for Virtual NOC use. After data analysis, we will implement the system detecting mass outages. We have formed a hypothesis:

Hypothesis 3 *It is possible to form groups to detect possible mass outages considering relevant alarm IDs or types while utilizing the site information and features available in the alarm data.*

A mobile network operator may have multiple electricity providers and the electricity network topology is not correlated with the mobile network topology. For transmission mass outage detection, usage of the mobile network topology is more beneficial. For power mass outage detection, geographical information of the network elements is more

relevant than the mobile network topology, as the electricity network is not connected to the mobile network topology and the power distribution can be decentralized between the sites. Nevertheless, power failures have a dependency on location and multiple sites containing electricity input failure pose a risk to the network of losing coverage.

4.4 Data analysis

To validate our hypotheses, we first analyze the information contained in the alarms, such as identification numbers and alarm types. The values in the fields, that we use to filter the alarms, are obtained by the domain experts. We can use an alarm identification number that reveals what the problem is related to. Accordingly, we filter power outage relevant alarms based on these fields.

The analysis and experiment setting is the following. We use a dataset of alarm data from a few months of time period with almost 2 million alarms. For confidentiality reasons, we do not provide the information of the accurate time period. The used alarm data is limited to alarms from Huawei network elements in the analysis. The chosen dataset represents a subset of a group that reflects the characteristics of the entire population as it contains all alarms from that time period from one operator. We note that the data of the other operators and vendors might vary, and site locations have variations between countries. Still, the information considered in the thesis is also available in other vendors alarm data and we believe our insights would be valuable to all operators. The alarm dataset contains the information available in the alarms. The site dataset contains the information available in the inventory data; the site name and location information in the form of latitude and longitude. Each alarm contains the name of the site. In our analysis, the alarm dataset and site dataset are merged based on the site name and we do this to map the alarm to a certain location. In the implementation, we use inventory data to query the site location based on the site name. Site name and timestamp are extracted from the alarm data fields. We calculate based on alarm raise time and clear time the duration for which the alarm has been active. Alarm datasets are analyzed with JupyterLab using Python programming language, a data analysis library Pandas, and machine learning library Scikit-learn.

For the analysis, we filter power-related alarms that indicate that the site has lost power. Domain expert knowledge is used to obtain the relevant alarm identification numbers. The total number of power alarms in the dataset after filtering the power-related alarms is

minutes	alarms	%
-	41206	100
10	3520	8,54
5	4362	10,59
2	5756	13,97
1	7052	17,11

Table 4.1: Dataset contains all alarms from a continuous time period from Huawei. After filtering power alarms, the dataset contains 41206 alarms. Alarms are additionally filtered based on the time that they have at least remained active (minutes).

41206. We consider different scenarios and include alarms in the analysis that have been active at least over a certain duration of time (at least x minutes). The parameters used in the analysis are 1, 2, 5 and 10 minutes. Filtering results can be seen in table 4.1.

In the context of power related alarms in the analysis, we include permanent failures and do not consider flapping alarms. We define the duration of minutes that the alarm needs to be active. Alarms that have shorter duration are considered as a flapping alarms in our case. If the site has lost its power for a longer period of time (e.g., 10 minutes), we can assume the failure and problem is more permanent and that possible future actions might be needed (e.g., sending a maintenance team to the site or resetting a base station). In some situations, if a so-called alarm storm of multiple incoming flapping alarms is detected, it may possibly be short term problem that does not need immediate actions. Hence, we do not consider faults caused by flapping alarms.

We consider different scenarios, with two parameters; the activity time of the alarm (1 to 10 minutes) and the time-window in which the fault alarm arrives. We consider different sized time-windows in the analysis. We assign alarms to their own window if they appear inside the defined time interval. We analyze data with time windows of 2 minutes, 5 minutes, 10 minutes, and 20 minutes. We use windows that contain more than 5 alarms in the analysis. We iterate through windows and plot all sites with active alarms of the same type in that window. We also plot all sites in that defined area that do not have active alarms of that type. One of these situations is plotted and presented in Figure 4.1. Red dots represent sites with active alarms which have been active for over 5 minutes. Light blue dots represent other sites located in the area. To perform a preliminary analysis of situations considered as a power mass outage, we investigate the plots and identify sites

as a part of a mass outage or outlier with the help of domain experts. In the situation in Figure 4.1, all alarms are raised within a few seconds of each other. Additionally, all alarms are cleared almost simultaneously. Thus we can conclude with high confidence that the alarms are caused by the same high-level failure. Note that in real-time detection, the clearance time of the alarm is not available during detection as a clear event is sent only when the fault is solved. Still, we may utilize the clearance time of the alarm validating found groups.

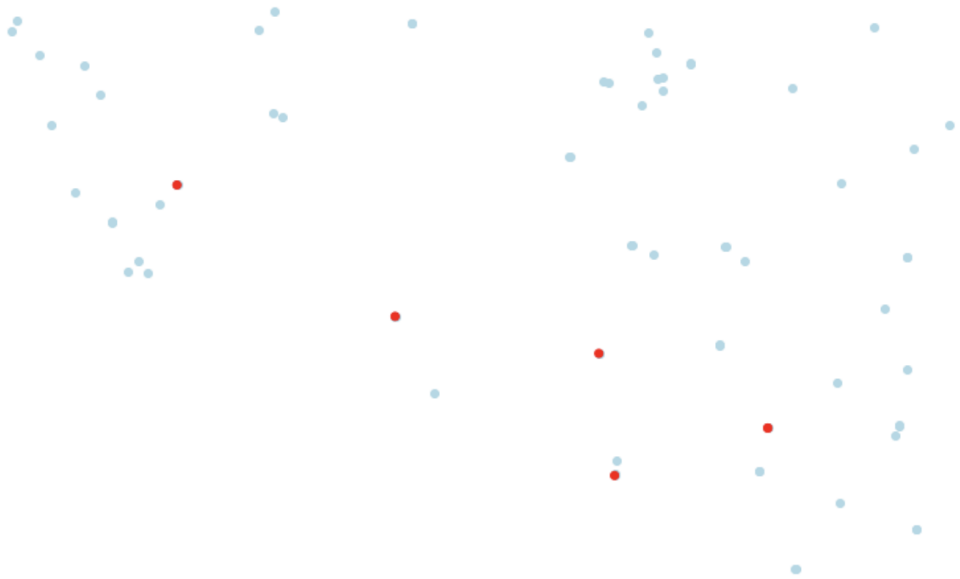


Figure 4.1: Example of plotting one group in the alarm analysis. Red dots demonstrate sites with active power alarms. The current situation presents a power mass outage. The site has a nearest neighbor which is not a part of the mass outage. X- and y-axis latitude and longitude values are removed to protect proprietary information.

First, we use the DBSCAN algorithm with different distance and minimum points values in the analysis. We also use the Scikit Learn ball tree algorithm [32] with distance metric haversine. Other settings are set to default. Ball tree is computationally less expensive

for finding the nearest neighbors compared to the Brute force method and we use it in the analysis to query for k-nearest neighbors. Density-based clustering in offline state manages to find some groups to be considered as power mass outages. However, the results depend on the selected parameters, in particular, the distance value. We have to adjust the value based on the density and distances between sites. Generally, DBSCAN is more suitable for evenly distributed data sets [3].

We analyze the data with DBSCAN with different parameters. Results of one scenario where alarms have been active for at least 5 minutes and groups formed from alarms that appear within a time-window of 5 minutes can be seen in table 4.2. With this example scenario, a dataframe contains 146 groups consisting of 1411 alarms. We used different distance values (epsilon) and minPts running DBSCAN. Table 4.2 shows found clusters and outliers from the whole alarm dataset with different parameters. Naturally, a larger distance was able to find more alarms to be included in the clusters. However, with higher epsilon values, the algorithm includes alarms into the cluster that should not be defined to the same mass outage. In these cases, there exist too many sites, that do not contain power issues that remain in between. We have confirmed this observation with RAN domain experts. We notice, that in multiple cases, groups defined as a mass outage have alarm occurrence time within a minute's time interval. However, timestamp and alarm content alone cannot reliably define the group. We notice investigating the data plots with the domain experts, that some formed groups contain data points that should not be included in the mass outage cluster.

Figure 4.2 illustrates problems with the static distance value and the importance of taking the density of the sites in the current area into consideration. In the sparse area marked with yellow and red circles, DBSCAN manages to find the correct mass outage cluster if the distance value used is large enough. Here epsilon value is set large, as it detects a mass outage group in the rural area. If we would investigate similarly located sites in a denser city area with the same epsilon value, the algorithm would find a cluster. However, the found points should be defined as outliers.

Estimations with the domain experts provide two main observations; how we define a mass outage and how in neighboring sites in the area, one may have a power failure and the other not. First, situations where multiple closely located sites with active power alarms exist, and which do not have other sites in between and occurrence time interval is under 10 minutes, should be defined as a power mass outage. Situations, where the clearance time of the alarms is almost simultaneous, reinforces the observation. We are not able

Epsilon	MinPts	Labels	Clusters	Outliers	% in cluster
e	3	1411	37	1277	9,50
2e	3	1411	126	918	34,94
3e	3	1411	141	818	42,03
4e	3	1411	155	742	47,41
5e	3	1411	157	705	50,04
e	4	1411	15	1345	4,68
2e	4	1411	51	1163	17,58
3e	4	1411	71	1035	26,65
4e	4	1411	84	956	32,25
5e	4	1411	86	921	34,73
e	5	1411	5	1385	1,84
2e	5	1411	17	1312	7,02
3e	5	1411	35	1191	15,59
4e	5	1411	51	1089	22,82
5e	5	1411	56	1047	25,80
e	6	1411	1	1405	0,43
2e	6	1411	5	1378	2,34
3e	6	1411	17	1292	8,43
4e	6	1411	26	1219	13,61
5e	6	1411	29	1185	16,02

Table 4.2: Results of DBSCAN algorithm in data analysis with different parameters. Epsilon value is defined as a constant value e and run with 1*e, 2*e, 3*e, 4*e, and 5*e. Accurate distance is not available here due to data confidentiality.

to utilize alarm clear time in real-time detection. Still, the clearance time of the alarm can be verified from the data and can be used to validate found mass outage groups in the analysis. With longer time intervals, we can find cases where the mass outage has possibly expanded. Secondly, power mass outage clusters may be formed of areas that contain sites in between which do not have active alarms. Sites without active alarms are not included in the mass outage cluster but indicate the fact that due to the division of power distributors, the nature of the mass outage cluster shape is arbitrary and can be fragmented.

We investigate the nearest neighborhoods of sites of the found groups. The goal is to define whether arbitrarily shaped groups with active alarms are connected through the closest sites. As a result of different power distributors and the cluster's points defined as mass

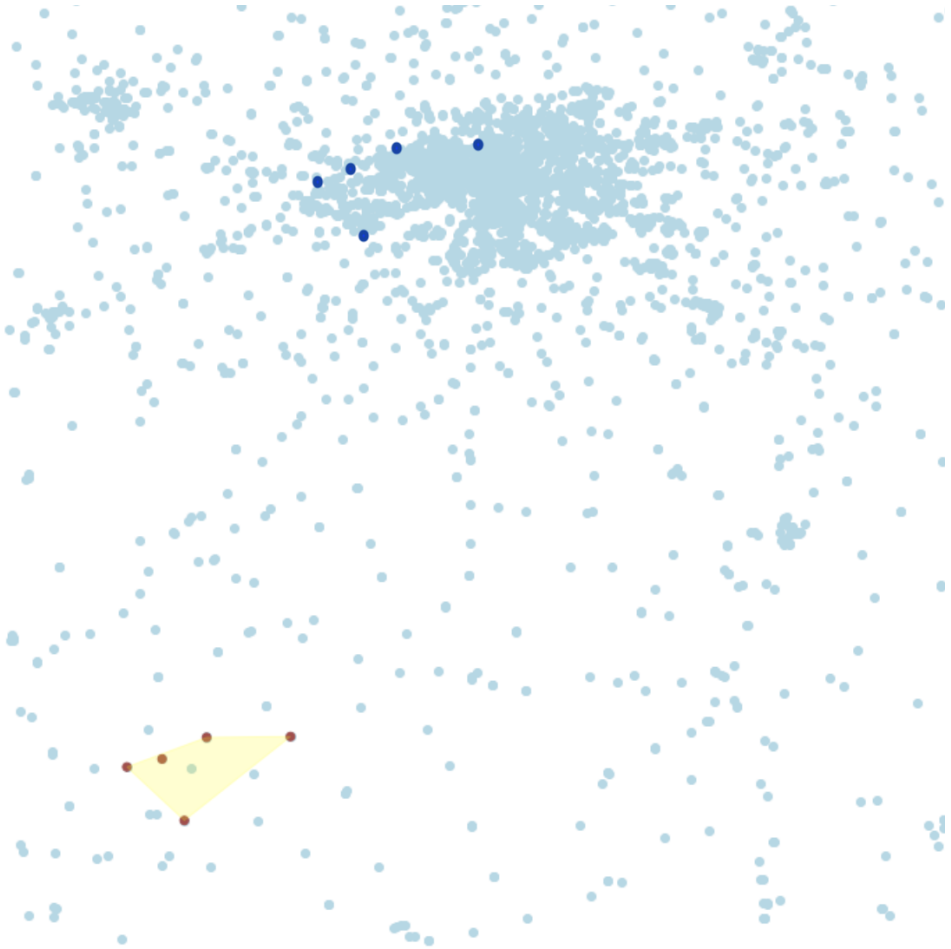


Figure 4.2: A figure illustrates how density-based algorithms with a static distance value cannot be used to find mass outages. The yellow area presents correctly found power mass outage cluster. Dark blue dots represent similarly located sites in the rural area. As the epsilon value is static and large, a false positive mass outage is also detected in the rural area. With a smaller value, the found mass outage would not be detected. X- and y-axis latitude and longitude values are removed to protect proprietary information.

outage groups are not always direct neighbors (with a small enough k value), connecting sites directly through neighborhoods is unreliable.

In Figure 4.3, dark blue dots represent sites with active power alarms present and form a mass outage cluster. We notice that with the nearest neighbor count 10 (k=10), we are not able to discover correlated sites traveling through neighbors. In the mobile networks, the groups that we define as mass outages may contain sites in the intermediary area that do not have active alarms. These sites are not naturally included in the mass outage cluster as the group is formed based on sites that have generated a fault alarm informing of a power failure. Increasing the number of neighborhood count would find also mass outage

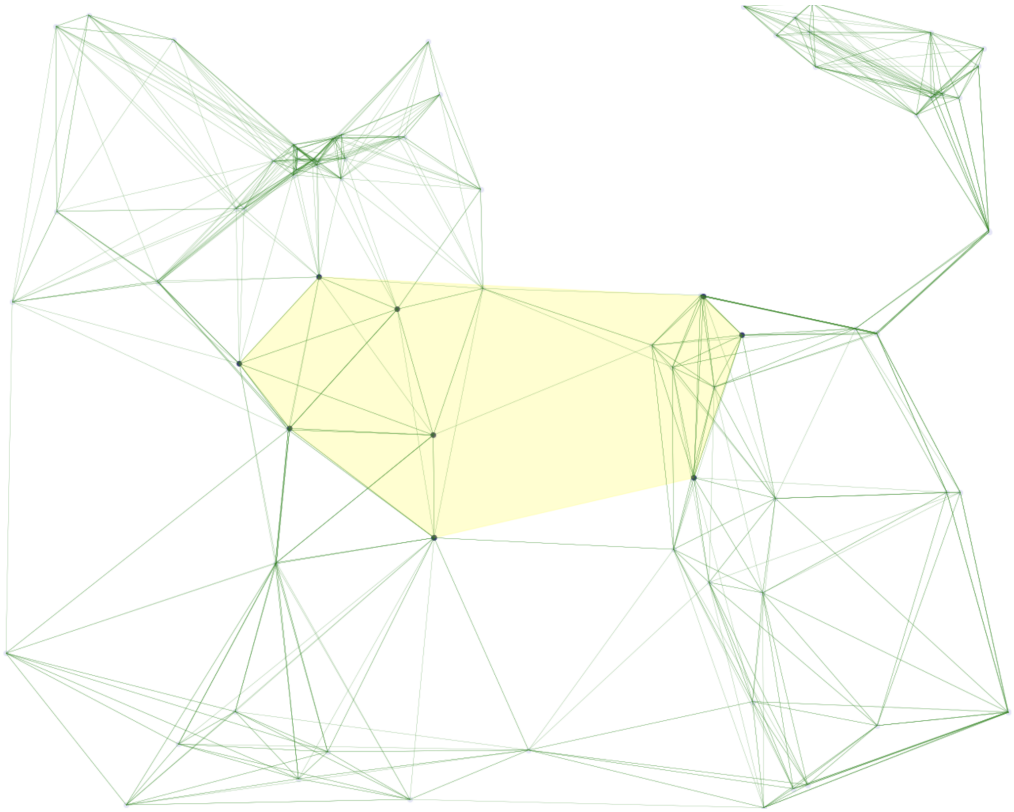


Figure 4.3: Correlated alarms within a time-window which are not direct neighbors. Lines present the nearest neighbors of each site (k = 10). Light gray dots represent sites that contain no active alarms and dark blue dots represent sites with active power alarms present. Detected mass outage (yellow area) is not found searching directly 10 nearest neighbors. X- and y-axis latitude and longitude values are removed to protect proprietary information.

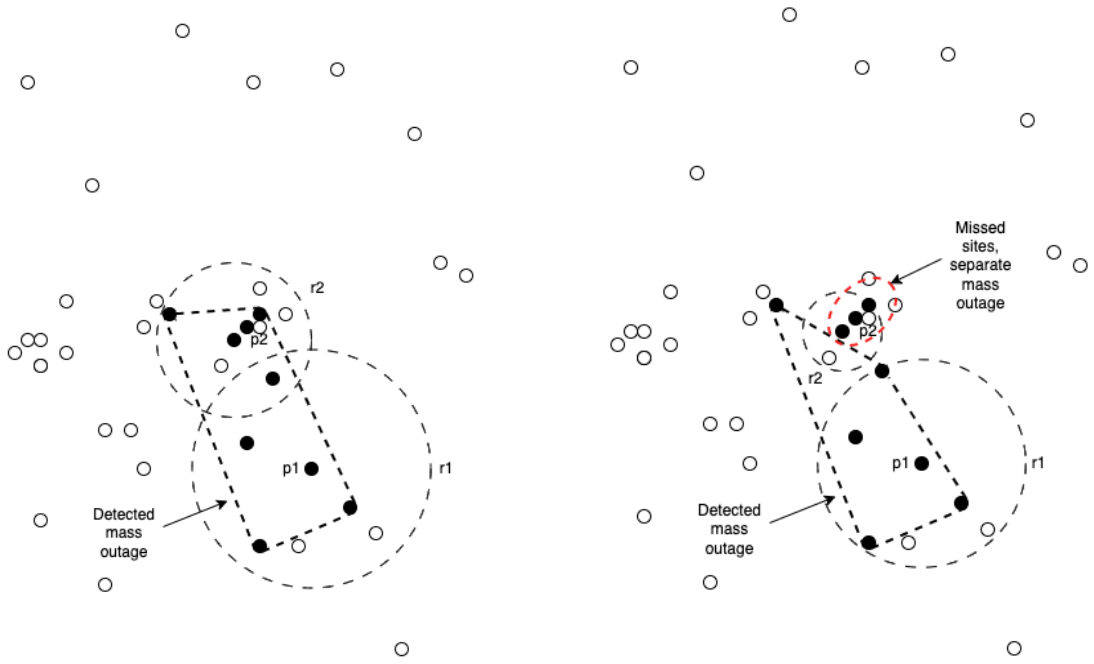
groups that can be seen in Figure 4.3. Raising the neighbor count would cause issues especially in rural areas. A high number of k include neighbors without considering the distance. With density-based clustering, we may control the distance and explore different distance scaling factors to use to define the clusters. Additionally, too large a value of k could increase the number of false positives and is computationally more expensive.

To achieve more accurate results, we then consider individual density for each site and analyze each time-window group separately. We try to assess the optimal manner of creating groups while considering the requirements and limitations of the real-time service and features of the alarms, in particular activity status and time. The shape of the group area is arbitrary and we consider each site arriving individually to the group. In the analysis, each active alarm mapped to a site is handled separately and we attempt to merge them with each other based on individually calculated distance values. In addition, we merge found smaller groups after each iteration.

We achieve good results by creating groups with distance values based on each site's density. For the definition of the radius value, the different regional density value is calculated for each site. We consider two different techniques for counting the density value for a site; k Nearest Neighbor and the number of sites within an area with different radius parameters. The nearest-neighbor distance has a natural determination of high- and low-density regions but a high value of k is averaging the values over a greater area. We use different k values (k=5, k=7, k=10) with kNN in the analysis. Parameter selection is based on discussions with RAN domain experts. We are able to observe the density of sites in the area with defined k values. We define an average distance value for a site by calculating the distance to k nearest neighbors and diving the sum of the distances by k. Then we define a search radius value for a site by multiplying the average distance value of a site by a scaling factor m. The scaling factor m varies between 1.5 and 5. The scaling factor is necessary as when counting the search radius value, we need to consider that there may occur sites in the intervening area that are not part of the mass outage. The scaling factor is higher than 1 as all sites with active alarms in the cluster may not be neighbors.

We also consider a second technique defining the search radius value for each site. We calculate the radius value for sites considering different distances as kilometers with varying values assuming a uniform distribution. The number of the sites within the distance r is counted for each site. Density D is calculated by counting the total number of sites N divided by area ($\pi * r^2$). We expect the average distance d of sites by calculating with n=2 with uniform distribution $d \sim \sqrt[2]{(1/D)}$. When an area is expanded, naturally average distance grows also with sites located in more dense areas if areas contain different densities. Density estimate with the number of sites inside the area does not take into account the borders areas of a certain operator. Thus, the calculated average distance value may offer a more false estimate as the measured area might be only half used.

Defining the average distance for each site to be used in the grouping, kNN detects better differences between dense and sparse areas and individual site differences. Naturally, as the number of neighbors grows, the average distance that has been obtained from the calculation increases. We also consider the calculated mean of source and target site kNN distances with the same scaling values. This approach is computationally more expensive and not used in the implementation. The radius measurement selection strongly influences the estimate of the sites belonging to the power mass outage cluster. For too big radius threshold, outliers will be included. For too small threshold, a cluster is separated into



(a) Radius defined correctly finding correlated sites.

(b) Unique defined radius for each site with too small search radius values fails to find connecting sites with alarms.

Figure 4.4: The search with different scaling values affecting search radius.

multiple different clusters.

Figure 4.4 depicts the situation with a different defined radius when searching with customized density values defined for each site. Figure 4.4 (a) illustrates an example of a successfully detected mass outage. Black dots represent cell sites from which a fault alarm is active. White dots represent other cell sites in the mobile network without any active power alarms. The calculated search radius varies in different geographical regions and is unique for each site.

Two examples of cell sites, p1 and p2, are considered in more detail in Figure 4.4 (a) and 4.4 (b). Site p1 is associated with search radius r1 and site p2 is associated with search radius r2. In Figure 4.4 (a), four other alarming sites can be found within a search area for p1. For p2, we can also find 4 alarming sites. Since site p2 density is higher than in site p1, the search radius is shorter for site p2 than for site p1. In Figure 4.4 (a), both neighborhoods contain a common site and are assigned to the same mass outage cluster. Figure 4.4 (b) illustrates an example of a separation of a mass outage. Search radius r2 is too low, and a group of alarms with site p2 arriving later are not added to the same

found mass outage cluster. Search radius r2 may be too low for example because of not applying a sufficiently high scaling factor.

4.5 The algorithms

We create a sequential algorithm that handles a continuous data stream and creates clusters of possible found mass outages of different types. The cluster creation uses a density-based approach and the algorithms are created specifically for detecting power and transmission mass outages. Observations are made sequentially as the generated data is affected by the previously formed groups and group identification numbers. Addressing the memory issues, we save only relevant alarms to the database. After a certain time period or when the alarm is cleared, the alarm is removed from the database table. Different types of alarms, such as power or transmission, are handled separately. We consider only alarms that are active after m minutes, which is a configurable value. Alarm merging is attempted after a wait time m to avoid taking flapping alarms into the detection.

Algorithm 1 Merging alarms into groups

```
1: Merging(incomingAlarm)
2: radius ← averageDistance(incomingAlarm) *scaling factor
3: if radius > definedMaxRadius then
4:     radius = definedMaxRadius                          //We limit the radius
5: end if
6: alarmList ← find alarms within radius and time-window and type
7: if alarmList is not empty then
8:     assign incomingAlarm into same group as alarmList[0]      //first alarm in a list
9:     assignedGroup ← alarmList[0].group
10: else                  //there exists no group within radius and time-window and type
11:     Assign new group to alarm
12:     assignedGroup ← new group
13: end if
14: for alarm in alarmList do                          //Merge possible separate groups
15:     if alarm.group not assignedGroup then      //if in found area exists other group
16:         alarm.group ← assignedGroup
17:     end if
18: end for
```

Merging each fault event into a group is implemented using the custom distance value with a scaling factor as a search radius for each alarm and site. We calculate the average distance value for a site based on a configured neighbor amount (k) value. When an alarm is attempted to merge into a specific group as seen in Algorithm 1, the search radius is defined based on the found average distance value of a site (with kNN) and configurable scaling factor. The maximum radius is set in the configurations to avoid too large search areas in sparse locations. We define values for the configuration parameters based on the test results from the data analysis and the real networks. Other active alarms are searched based on the defined radius, in a specified time-window and alarm type which are configurable values. If alarms match the search, we merge arriving alarm to the same group. In cases where we cannot find any alarms matching the search criteria, a new group is created for the arriving alarm. If several groups are found for the arriving alarm, groups are combined.

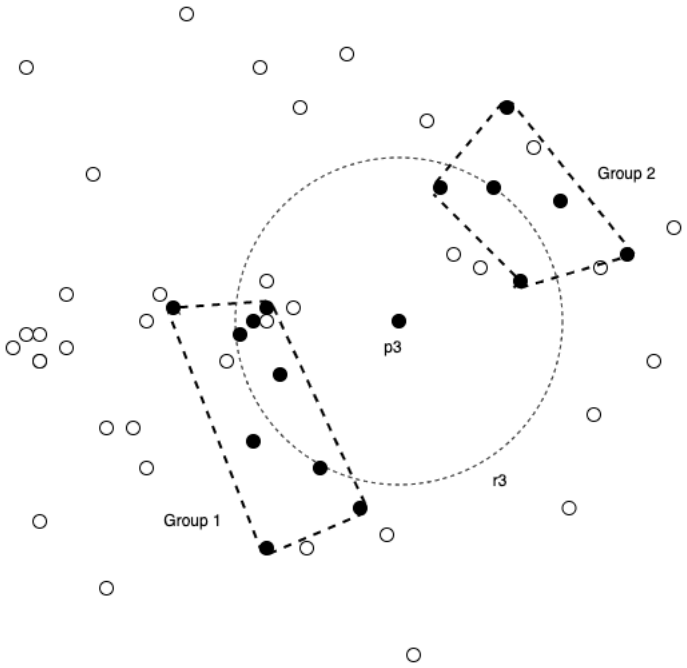


Figure 4.5: Merging of two groups. When a point *p3* arrives, it is merged to the Group 1 because sites with active alarms are found within the search area *r3*. As *p3* reaches sites also from the Group 2, the groups are merged.

Figure 4.5 illustrates an example of merging two groups. In the example, two groups (Group 1 and Group 2) have been detected. The search radius between the cell sites of the two different groups has not been long enough and two separate groups have been created. When we receive an alarm from the cell site p3 with search radius r3, sites from

both Group 1 and Group 2 are found. This results in adding p3 to the Group 1 and combining all sites from the Group 2 to the Group 1.

Mass outages are formed from the groups based on a configurable site threshold value. Different operators may have diverse views on a minimum number of different sites to have active alarms to form a mass outage group. If the threshold is crossed, we attempt to find already defined mass outages and add an alarm to the mass outage group. Otherwise, we form a new mass outage group containing all alarms already belonging to the defined group with the same id as can be seen in Algorithm 2.

Algorithm 2 Form mass outage

```
1: HandleMassOutageCheck(groupId)
2: alarmsInGroupList ← findAlarmsInGroup(groupId)  //list of all alarms in the group
3: sitesList ← find distinct sites in alarmsInGroupList  //list of sites of the alarms
4: if sitesList.size >= siteThreshold then
5:     alreadyDefinedMassOutagesList ← find alarms from alarmsInGroupList with massOutage id  //Check if some alarms in a group already belong to mass outage
6:     if alreadyDefinedMassOutagesList is empty then
7:         id ← new id  //Mass outage id
8:         Save new mass outage
9:     else
10:        id ← alreadyDefinedMassOutagesList[0].massOutageId
11:        Add alarm to existing mass outage
12:    end if
13: else
14:    No found mass outages
15: end if
16: for alarm in alarmsInGroupList do
17:    if alarm.massOutageId not id then
18:        alarm.massoutageId ← id
19:    end if
20: end for
```

4.6 Mass outage detection implementation

We use Apache Kafka for alarm streaming in our Mass Outage Detection Service. Apache Kafka is an event streaming platform used for handling continuous data streams as it is capable of handling high-volume data and thousands of messages per second. The Mass Outage Detection Service implementation can create mass outage groups from real-time alarm data stream. We use a different sets of configurable values to achieve an adaptable system for different operators. We use our own custom algorithm to merge alarms into groups and detect mass outages. The Mass Outage Detection Service is made with Spring Boot [34] and written in Kotlin programming language [22].

To select clusters, a minimum site threshold is applied in the proposed Mass Outage Detection Service. We assume that group of 3 or more is a mass outage and we benefit from creating only one possible trouble ticket for the group. We choose a low value for this parameter to be able to detect more clusters. We define in the configurations the number of neighbors, maximum radius per search (meters), time-window to include alarms into clusters, scaling factor, wait-time (how long the alarm needs to be active before clustering), and threshold for the minimum points in the cluster. One example of the power mass outage parameter configuration in the service implementation is presented in Figure 4.6.

```
power-mass-outage:
    site-threshold: 3
    wait-time: 10m
    time-window: 5
    radius-scaling: 2.5
    max-radius: 25000.0
    neighbors: 10
```

Figure 4.6: We define different parameters in the configurations. When the site threshold is met, we form a mass outage cluster. Wait time is the time that the alarm needs to be active to be considered in a group. We include alarms within a certain time-window. The scaling factor is used to fine-tune the clustering search radius. We define a maximum radius (meters) and do not cluster alarms beyond that distance. Additionally, we define the number of the neighbors that we use in kNN to find the average distance for a site.

Our Mass Outage Detection Service uses PostgreSQL database as a data store and PostGIS extension to query geographical information efficiently. PostGIS is an open source, freely available spatial database extender for the PostgreSQL Database Management System. PostGIS has spatial functions allowing location queries. We take into consideration fast querying and access to up-to-date information in the database design. The database contains tables to store information on alarm events and mass outages with essential information. We limit the data in the database in 'active alarm' table by deleting cleared alarms and taking into consideration the occurrence time of the alarms to address restrictions in execution time and memory. The database also contains a table for the site information with the site name, latitude, and longitude. We do not describe the exact database design to protect implementation details. PostGIS has functions and spatial indexing allowing efficient geography and geometry queries.

As in SOStream discussed in Chapter 3, we want to use an automatic selection of the density threshold. We define an average distance value for each site. Each site has information on latitude and longitude in the site database table. We extract the site name from the alarm data and query distances of k nearest neighbors (kNN) with the help of the site data. Value of k is defined based on the configuration. We calculate an average distance value for the site by searching k nearest neighbors among all the sites in the network and then divide the sum of the distances by the k. The search radius value is calculated by multiplying the average distance for the site by the scaling factor. Unique radius calculation for the site is done with PostGIS query run time when receiving the alarm which is attached to a specific site.

We monitor the continuous alarm stream and extract essential information from the alarms in a separate service. From the Element Management Systems (EMSs), SNMP capturing is made and relevant fields of the alarm are saved as an event object containing essential information. Arriving data stream contains all events from the EMS. We capture the SNMP traffic and map OIDs to uniquely identify alarms. The SNMP trap has variable bindings for different fields we need to extract such as site name and occurrence time. After parsing the alarm fields, we store each alarm event object and send it to a Kafka topic. The Mass Outage Detection Service subscribes the Kafka topic and monitors the continuous alarm stream. The service implements the algorithms described in the previous section. The high-level data flow is presented in Figure 4.7.

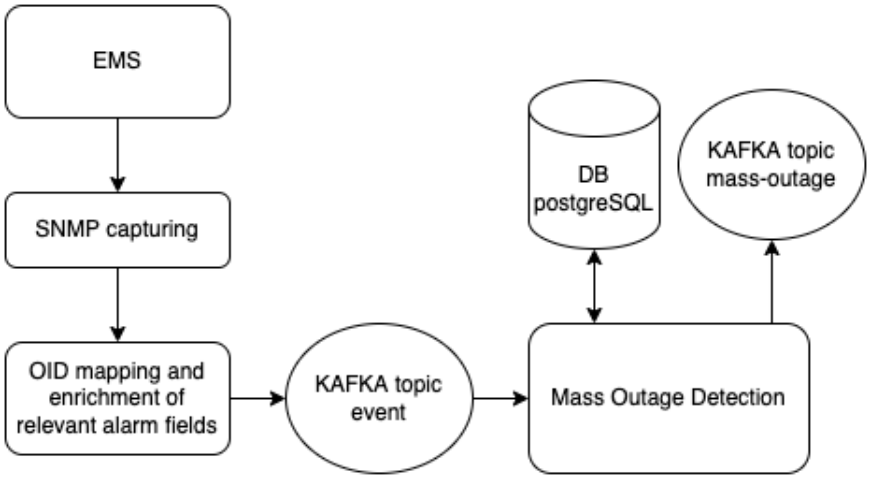


Figure 4.7: The data is mapped to user-readable form and relevant fields are extracted and sent as an event object through Kafka. Mass outage detection receives events for detection and forwards findings to Kafka and stores them in the database.

The Mass Outage Detection Service filters alarms based on the alarm identification numbers specified in the configurations. We can have multiple different types of mass outages such as transmission or power mass outage and each type contain a different set of alarm identification numbers. Identification of the preliminary filtering is done based on domain experts, operator wishes, and depending on Virtual NOC algorithms. For example, some Virtual NOC algorithms handle power-related issues and concern only certain types of alarms. We consider the states of the alarms and monitor if an event instance of the fault alarm has arrived and when the alarm is cleared. When a fault alarm is detected and has been active over a specified time threshold, we search for other alarms in existing groups as seen in Algorithm 1. Then we assign an alarm to the found group and merge other possibly found groups or create a new one. If the site threshold is crossed in the group, we create or update the mass outage group. A chart for handling an event alarm in the Mass Outage Detection Service is presented in Figure 4.8.

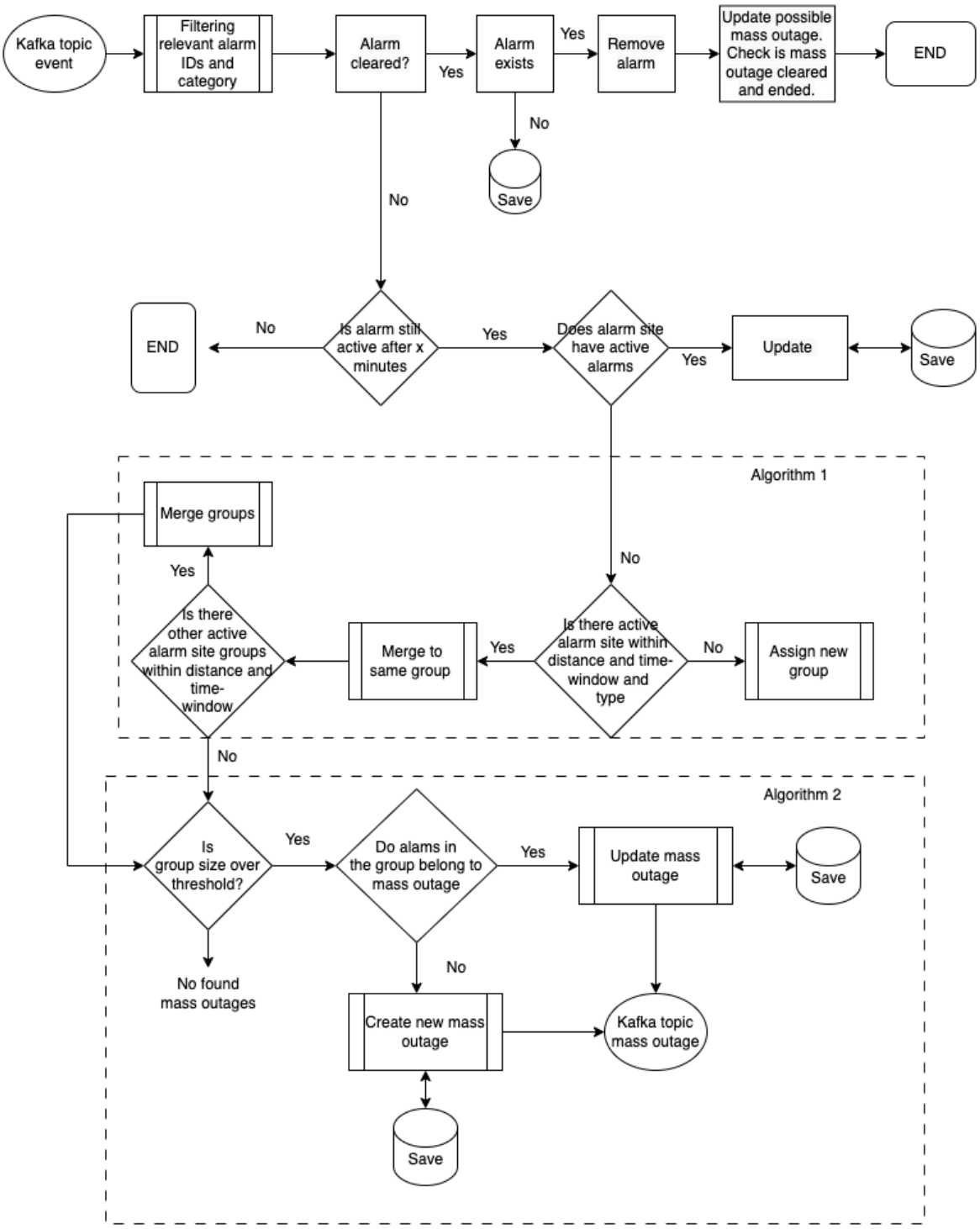


Figure 4.8: Alarm handling and merging alarms into groups in the Mass Outage Detection Service.

5 Evaluation and discussion

The motivation for the Mass Outage Detection Service stems from the need to automatically correlate alarms based on their type to provide essential information for Virtual NOC algorithms and decrease the number of created trouble tickets. We assume that identifying correlated groups of alarms will aid in creating more value for the operators and ease the root cause analysis. Our evaluation is based on the real data collected after deploying our service in two large networks in different countries. First, we evaluate our algorithm and implementation with a synthetic dataset. With the synthetic dataset, we are able to test different parameters faster and more easily. The synthetic dataset also eliminates privacy issues presenting the results. We create a test dataset and network referred as *Test Network*. Second, we implement and deploy the Mass Outage Detection Service in two real mobile networks. Each network contains several thousand of sites. We refer to these two networks as *Network 1* and *Network 2*. We test different parameters (scaling factor and neighbor amount) defining mass outages with Test Network and in Network 1 and Network 2. Due to confidentiality reasons, we present a high-level overview of the performance of the real networks, whereas the synthetic dataset provides a more detailed analysis. Both power and transmission mass outages appear in real networks. In the synthetic network, we consider only power-related alarms and power mass outages. To the best of our knowledge, our algorithm for detecting power and transmission mass outages in real-time is novel and has not been presented in the literature previously. A patent application has been filed based on this work.

5.1 Generation of the synthetic network

To evaluate the proposed clustering algorithm and to present the results more openly, we use a synthetic dataset. Alarm data is formed based on the perceptions from the real data and we use prior labeling to evaluate the created clusters. Our test data consists of alarms that imitate SNMP traps containing all relevant fields of the alarm. The included fields are alarm id, alarm occurrence time, alarm managed object name, alarm type, alarm additional info, alarm extend info, alarm category, and alarm CSN. Alarm restore time is included in the fields when a clear event is sent. We imitate real situations that we

have seen, and domain experts have verified as power mass outages. We create site data with location information. The site data is formed by generating randomly x number of sites with location (latitude, longitude) inside a defined country. We use Denmark and 10 000 sites in our scenario. We choose the particular number of sites because it provides a suitable distribution to create, and test observed situations.

We generate 64 power alarms that contain the information of the site name and alarm occurrence time. Based on our observations of data from the real mobile networks (Network 1 and Network 2), we choose the site and occurrence time for 35 of these alarms, so that they form 5 different mass outage clusters. Data points belonging to a cluster have a specific mass outage cluster id, they belong to. Alarms that are labeled belonging to a certain mass outage cluster, have a duration of more than 10 minutes and occur within 5 minutes of each other. The five different clusters are of different sizes and comprise 14, 7, 6, 5, and 3 alarms. We also trigger 29 outlier alarms, in which the site is located such as it should not be considered as a part of the mass outage. We compare labeled data against the clustering result of our Mass Outage Detection Service. Figure 5.1 shows the layout of the test scenario and the alarms generated within an hour time interval in Denmark at the adjacent site locations. While our Test Network is useful, it does have shortcomings. The distribution of sites regionally is not ideal and lacks rural and city separation. The division of regions could be improved in future work for Test Network. Additionally, the test scenario does not cover all possible cases from the different networks. Still, we can create similar situations as in real networks.

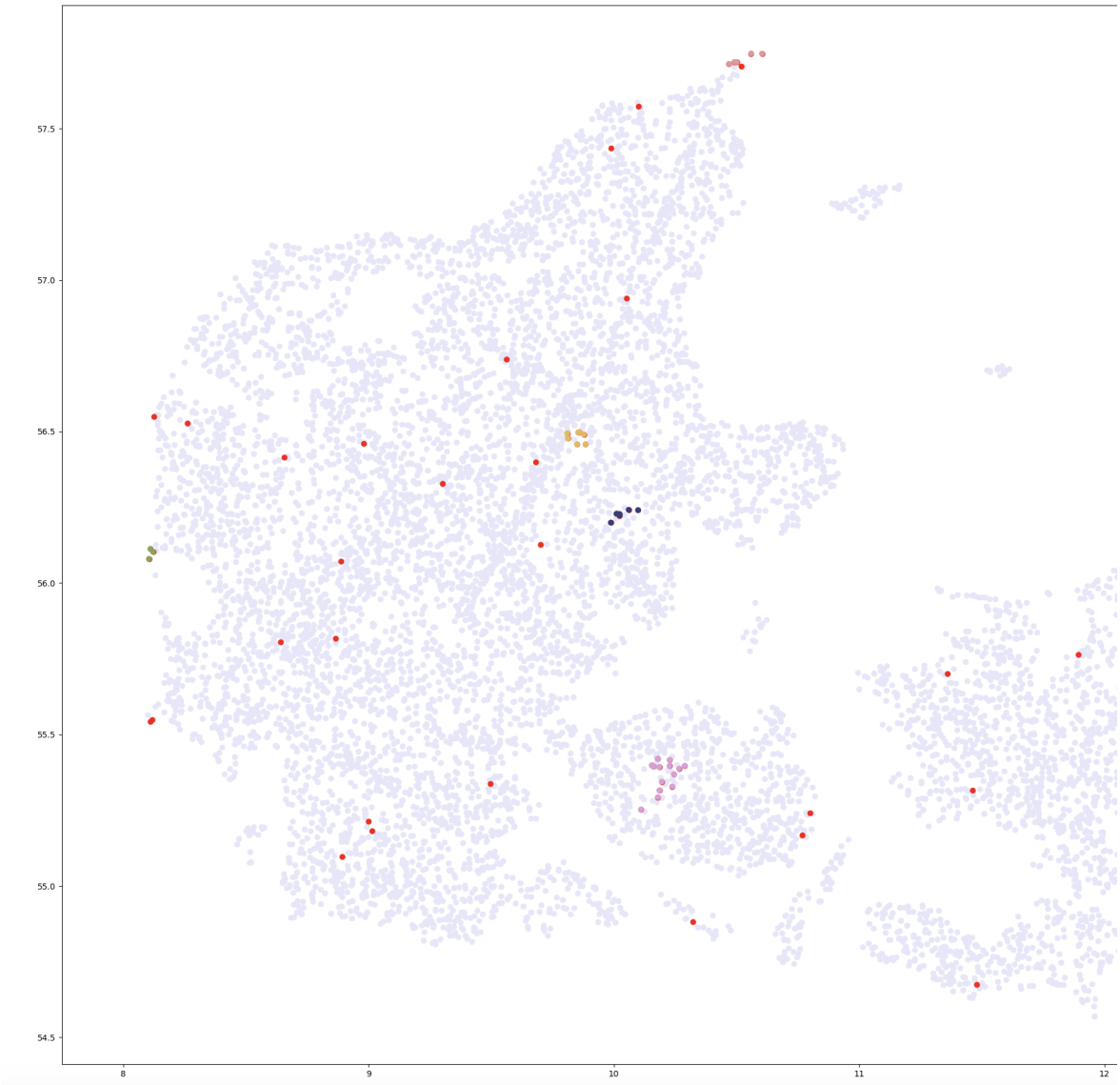


Figure 5.1: Alarms generated in the Test Network. Gray dots represent sites without active power alarms. Red color dots present outlier alarms and other alike colored data point groups represent each one mass outage cluster. The x-axis represents the longitude and the y-axis represents the latitude, the image representing the area of Denmark.

5.2 Performance of the detection service in the Test Network

We test how different parameter selection affects the clustering. We evaluate different parameters in the Test Network; the amount of neighbors and the scaling factor. We refer to data points that belong to a mass outage cluster true positive. We define data points

labeled to a certain cluster, but defined as outliers, false negatives. We calculate Rand Index for the test dataset measuring similarities between labeled data and results provided by our implementation. As discussed in Chapter 3, Rand Index gets a value between 0 and 1, where a value of 1 is received when clusters are the same. With value 0, clustering does not agree on any pair of the data points of true positives or true negatives.

We send generated alarms through Kafka. Generated alarms appear in Kafka in the same form as when monitoring real alarm data stream. After parsing the essential information from the traps, our Mass Outage Detection Service filters alarms based on the defined configurations. Parameters can be also set separately for different types of mass outages. Minimum site threshold for a mass outage cluster is set to 3. We set the wait time to 10 minutes. We test different radius scaling factors and different number of kNN.

kNN	Scaling factor	Clusters	Outliers	True Positives	True Negatives	False Negatives	False Positives
5	1	5	49	15	29	20	0
5	2	6	30	34	29	1	0
5	3	5	29	35	29	0	0
5	4	5	29	35	29	0	0
5	5	6	26	35	26	0	3
10	1	7	35	29	29	6	0
10	2	5	29	35	29	0	0
10	3	5	29	35	29	0	0
10	4	6	26	35	26	0	3
10	5	6	24	35	24	0	5

Table 5.1: Results from running the mass outage detection service with the test scenario from time-interval of one hour with different parameter values. Labeled data consists a total of 64 alarms, with 29 outlier alarms and 5 clusters containing 35 alarms.

The parameter k in kNN defines the average distance value to be used with a scaling factor defining the search radius for an alarm and its site. Naturally larger value of k increases the search radius. As stated in Chapter 4, correlated alarms are not always directly neighbors and in addition to the selection of k value, the scaling factor affects the search radius and search results. The scaling factor may be applied when we determine the search radius. This enables to adjust the search radius distance used to cluster the alarms. In addition, different types of mass outages may require a different search radius threshold value for regional clustering. A comparison of the parameters with different neighbor counts and

scaling factors defined in configurations is shown in table 5.1. When the scaling factor is defined as 1 (or not at all), it does not have any effect. The labeled dataset (ground truth) consists of 5 mass outage clusters with 35 cluster alarms and 29 outlier alarms. As seen in table 5.1, when the scaling factor is not used (sf = 1.0) with kNN=5, we discover more false negatives. Additionally, with scaling factor 1.0 and kNN = 10, two clusters are separated into 2 different (total of 4) clusters. Note that here alarms are marked as true positives if they are correctly identified as belonging to some mass outage cluster and as a part of the mass outage group. For example, clustering may assign 6 alarms to 2 different clusters with 3 alarms each when all alarms should be assigned to the same cluster. The alarms are marked as true positives as they belong to some mass outage group even if they should be assigned in the same cluster. Also, when increasing value k with kNN search, naturally we are able to use a smaller scaling factor and be able to find correlated alarms and form clusters accordingly.

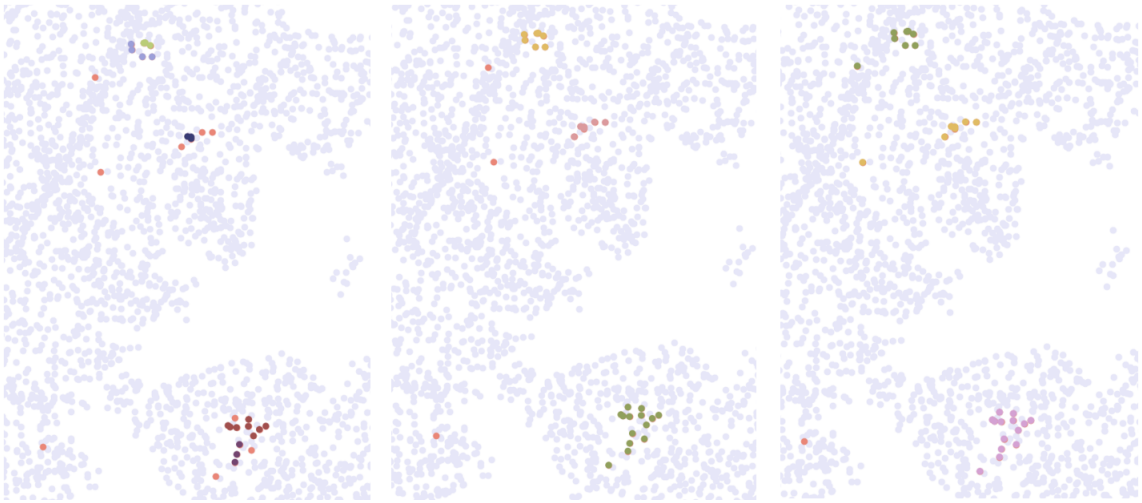


Figure 5.2: Different scaling factors (left: 1, middle: 3, right: 5) with kNN=10. Salmon color dots demonstrate outliers. Other similarly colored data points represent each one mass outage cluster. Gray data points represent sites without active power alarms. Too small factor separates clusters and produces more false negatives. Too large scaling factor includes outliers to clusters (more false positives).

However, searching even 10 nearest neighbors fails to cluster alarm correctly without the scaling factor. As seen in Figure 5.2, the search with 10 neighbors and scaling factor 1, separates 2 clusters and includes only part of true positives. Raising the scaling factor to 3, produces correct clustering results. Increasing the scaling factor to 5, includes outliers to two different clusters. Additionally, outliers are connected and an extra (incorrect) cluster is formed.

We measure the similarity of labeled dataset clusters and clustering results with different parameters by calculating Rand Index. Considering the nature of the dataset and choosing reasonable parameter values, we can expect reasonably high values for this index. The results in Figure 5.3 show that kNN 5 with scaling factor 3 to 4 and kNN 10 with scaling factor 2 to 3 provide exactly the same clustering result with the labeled dataset. If we do not use the scaling factor, we have to increase k to over 50 to achieve the correct clustering results with the test dataset (i.e., to achieve a Rand Index of 1). A larger value of k is also computationally more expensive. Additionally, increasing k value too high will cause more false positives, especially in rural areas, and does not take into account as accurately the density of an individual site in the area.

kNN	Multiplication	Rand
5	1	0,69
5	2	0,98
5	3	1,00
5	4	1,00
5	5	0,95
10	1	0,91
10	2	1,00
10	3	1,00
10	4	0,95
10	5	0,92

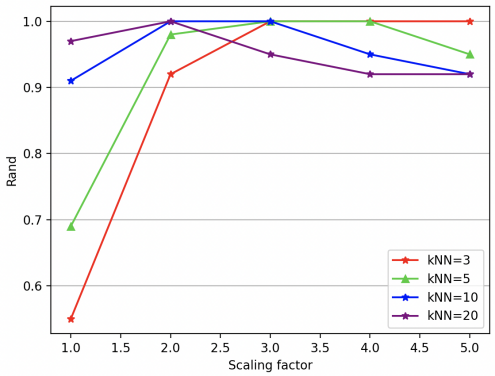


Figure 5.3: Rand index results from the synthetic dataset. The results present the impact of different scaling factors and k nearest neighbor parameters. These values are used, when we calculate the search distance radius for each site.

Results show that the parameter k in kNN may be set to 5 with scaling factor 3 to 4. Alternatively, k may be set to value 10 with scaling factor 2 to 3. Depending on the mobile network, operators can fine-tune the values and test in the real network which k values varying between 5 to 10 and scaling factors from 2 to 4 provide the most accurate results. The results confirm hypothesis 1 that mass outages can be detected by clustering the alarms according to temporal and spatial factors.

5.3 Performance of the detection service in the real networks

We observe interesting results from Network 1 and Network 2. Both networks contain thousands of sites. We are able to detect correlated groups of alarms in real-time and provide information on the clusters forward to Kafka topic to make decisions to combine trouble tickets. Validating found mass outage groups collected from Network 1 and Network 2 is performed additionally by the network operator specialists.

As mentioned with clustering validation methods in Chapter 3, we are able to measure the clustering algorithm performance with testing execution time, memory usage, or sensitivity. We monitor the execution time (the total time that it takes to process the data) in Network 1 and Network 2. We can see timestamps from the metrics we collect from the deployed implementations. The results show that our Mass Outage Detection Service is able to process data and execute the algorithm detecting mass outages in real-time. Responses are guaranteed within specified time constraints in 100 % of the cases we have seen. Different scaling factors may be configured for alarms having different alarm types. This enables the same algorithm to be used in parallel, for detecting different types of mass outages, for example, to separately detect power or transmission outages. This is beneficial as different types of outages may have different geographical correlations. For example, transmission outages may spread according to the topology of the mobile communication network, while power outages may spread according to the topology of the underlying electricity network.

When we receive an alarm and parse the site name, we then query its location (latitude, longitude) and calculate the average distance with kNN. The observed duration for the average distance calculation is between 64 and 116 milliseconds. As our implementation can filter and store only relevant data and also instantly remove not needed data points (cleared alarms or alarms that are older than time-window value), we have an extremely short search time. We are able to collect and cluster groups within a few milliseconds. As mentioned, we store only relevant information and additionally remove the data which is not needed anymore. With sensitivity analysis, operators may fine-tune the mass outage clustering parameters for each type of alarm. Different type of mass outages requires different parameter selection. Additionally, variables implemented as configurable values that we consider run time provide a continuous possibility to adjust and improve results based on operators' wishes.

In both Network 1 and 2, there appeared more transmission types of mass outages than power mass outages. Hence, our implementation found more transmission type mass outages. We also evaluate the number of tickets reduced when using our service. As an example of the ticket reduction, in Network 1, we found 29 mass outages over the considered time period. The mass outages included 117 alarms. The total number of alarms included in the detection in that particular time period is 1855. These alarms meet the conditions, they have lasted over a configured number of minutes and are of a certain type. Hence the total number of alarms that are not included in mass outages is 1738. In total, we have 88 tickets less (1855-1738) than the original 1855 (88/1855=4.74). The total number of tickets would have reduced by 4.74%. Considering only alarms that are included in the mass outage groups, ticket reduction in Virtual NOC algorithms is 75.21 percent ((117-29)/117= 0.7521). In Network 2, the corresponding results are 9.25% and 82.76%. The results from the real networks confirm hypothesis 2 that different types of mass outages can be detected with the same method using a different set of relevant alarm identification numbers. Still, different types of mass outages may need a different set of configurable k and scaling factor values to provide the most accurate results.

5.4 Benefits of the Mass Outage Detection Service

In this thesis, the Mass Outage Detection Service aims to decrease the number of created trouble tickets in the network operator's trouble ticketing system. Mass outage detection provides the possibility to combine trouble tickets or create only one ticket containing the information on relevant alarms and sites. Reducing the number of created tickets generates value for the operator.

In addition to the quantitative metrics presented so far, our solution has several additional benefits to the network operator. First, it provides faster detection of the service impact. Knowledge of the correlation of the sites that contain power issues will improve decision-making, speed up root cause analysis, and allow resolve faster the cause of the fault. This can be seen in improved quality for end-users. Second, the impact of using our service results in less time to fix issues and lower overall costs from the trouble tickets. These metrics can be captured by the network operator after a certain period of time after using the service. Finally, our implementation allows for flexibly configuring values of important parameters. With configurable values, operators may fine-tune clustering as the solution is customizable.

6 Conclusion

In this thesis, we introduced a method for discovering mass outages in mobile networks. We proposed a density-based clustering algorithm designed specially to form mass outage groups from certain types of alarms. The motivation behind the work is to ease the root cause analysis and minimize the number of created trouble tickets. We formed definitions for the mass outage and describe the process of extracting the essential information from the raw alarm data. We then constructed research hypotheses that we may form meaningful clusters by observing relevant alarms with certain identification numbers and performing clustering according to temporal and spatial factors.

To analyze situations that may be concluded as a mass outage, we first searched the data from a mobile network with almost 2 million alarms. After filtering alarms indicating power failures, approximately 41 thousand alarms were included for further analysis. We used DBSCAN with Scikit-learn to test the density-based algorithm approach when searching mass outages. We then considered individual radius distance calculated for each site with kNN and a scaling factor. A scaling factor enables to adjustment of the search radius and improves clustering accuracy. We implemented the designed Mass Outage Detection Service with the proposed algorithms and deployed it to two different real mobile networks. The Mass Outage Detection Service developed as a part of this thesis is now used as a part of the Virtual NOC product of Elisa Polystar. Results and information of formed clusters are used in Elisa Polystar Virtual NOC algorithm processes and handling of the mass outages is developed further in the continuation of this work. A patent has been applied based on the work done in this thesis.

To test how our Mass Outage Detection Service manages to find correlated alarms and performs real-time clustering, we deployed implemented service to two different mobile networks. Additionally, we used a synthetic dataset and test network for testing. Results show we are able to detect clusters correctly in real-time within time constraints. We showed how the parameter selection affects the results and suggested suitable parameter values. We calculated the reduction in created trouble tickets in two real networks. The number of mass outage groups depends highly on the mobile network and country. Still, operators benefit from each less-created trouble ticket as it reduces costs.

The implemented service creates value by reducing the number of trouble tickets and

providing direct information on correlating alarms. While we manage to find most clusters correctly, there is still a need for future work. When generating a synthetic test network for testing, the site data should have better separation of city and rural areas. It could have more realistic placement of the sites containing more dense city areas as well as sparse areas in between. This research can be extended further. Our solution focuses on finding correlations based on time and location. The possibility to include topological information could be investigated in the case of transmission alarms. A possible improvement could be to adjust the number of neighbors based on the area density. Thus, this would cause additional calculations during run time. Additional stress testing on how the service is performing under a larger alarm flow could be performed. Finally, future work will include testing the implementation as a part of Virtual NOC solution.

Bibliography

[1] M. W. A. et al. "Supporting Telecommunication Alarm Management System With Trouble Ticket Prediction". In: *IEEE Transactions on Industrial Informatics* 17.2 (2021), pp. 1459–1469. DOI: 10.1109/TII.2020.2996942.

[2] M. Aljibawi and M. Noordian. "A Survey on Clustering Density Based Data Stream algorithms". In: (Dec. 2018), pp. 147–153.

[3] A. Amini, T. Wah, and H. Saboohi. "On Density-Based Data Streams Clustering Algorithms: A Survey". In: *Journal of Computer Science and Technology* 29 (Jan. 2014), pp. 116–141. DOI: 10.1007/s11390-013-1416-3.

[4] M. Amirijoo, L. Jorguseski, R. Litjens, and R. Nascimento. "Effectiveness of cell outage compensation in LTE networks". In: *2011 IEEE Consumer Communications and Networking Conference (CCNC)* (2011), pp. 642–647. DOI: 10.1109/CCNC.2011.5766560.

[5] M. Ansari, A. Ahmad, S. Khan, G. Bhushan, and M. Siddique. "Spatiotemporal clustering: a review". In: *Artificial Intelligence Review* 53 (Apr. 2020). DOI: 10.1007/s10462-019-09736-1.

[6] S. Z. Asif. *Next Generation Mobile Communications Ecosystem: Technology Management for Mobile Communications*. 7st. Chichester, U.K: Wiley, 2011. ISBN: 9780470972168.

[7] M. Awad and H. Hamdoun. "A framework for modelling mobile radio access networks for intelligent fault management". In: *2016 Conference of Basic Sciences and Engineering Studies (SGCAC)* (2016), pp. 43–49. DOI: 10.1109/SGCAC.2016.7458004.

[8] J. Bellec and M. .-. Kechadi. "FECk: A New Efficient Clustering Algorithm for the Events Correlation Problem in Telecommunication Networks". In: *Future Generation Communication and Networking (FGCN 2007)* (2007), pp. 469–475. DOI: 10.1109/FGCN.2007.127.

[9] F. Cao, M. Ester, W. Qian, and A. Zhou. "Density-based clustering over an evolving data stream with noise". In: *In 2006 SIAM Conference on Data Mining*. 2006, pp. 328–339.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[11] C. Fahy and S. Yang. "Finding and Tracking Multi-Density Clusters in Online Dynamic Data Streams". In: *IEEE Transactions on Big Data* 8.1 (2022), pp. 178–192. DOI: 10.1109/TBDATA.2019.2922969.

[12] M. Garofalakis, J. Gehrke, and R. Rastogi. *Data Stream Management: Processing High-Speed Data Streams*. Springer Berlin / Heidelberg, 2016.

[13] A. Ghayas. *Cell Sites And Cell Towers In A Mobile Cellular Network*. https://commsbrief.com/cell-sites-and-cell-towers-in-a-mobile-cellular-network/, Accessed on 7th March 2022. 2019.

[14] M. Ghesmoune, M. Lebbah, and H. Azzag. "State-of-the-art on clustering data streams". In: *Big Data Analytics* 1 (Dec. 2016), p. 13. DOI: 10.1186/s41044-016-0011-3.

[15] T. Hayford-Acquah and B. Asante. "Causes of Fiber Cut and the Recommendation to Solve the Problem". In: *IOSR Journal of Electronics and Communication Engineering* 12 (Jan. 2017), pp. 46–64. DOI: 10.9790/2834-1201014664.

[16] S. Hou and X. Zhang. "Analysis and Research for Network Management Alarms Correlation Based on Sequence Clustering Algorithm". In: *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)*. Vol. 1. 2008, pp. 982–986. DOI: 10.1109/ICICTA.2008.263.

[17] Huawei. *Technical support*. https://support.huawei.com/, Accessed on 11th March 2022.

[18] C. Isaksson, M. Dunham, and M. Hahsler. "SOStream: Self Organizing Density-Based Clustering over Data Stream". In: vol. 7376. July 2012. ISBN: 978-3-642-31536-7. DOI: 10.1007/978-3-642-31537-4_21.

[19] A. P. Iyer, L. E. Li, and I. Stoica. "Automating Diagnosis of Cellular Radio Access Network Problems". In: *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. MobiCom '17. Snowbird, Utah, USA: Association for Computing Machinery, 2017, pp. 79–87. ISBN: 9781450349161. DOI: 10.1145/3117811.3117813. URL: https://doi.org/10.1145/3117811.3117813.

[20] O. Jukič, V. Halusek, and M. Špoljarič. “Low-level alarm filtration based on alarm classification”. In: *2009 International Symposium ELMAR* (2009), pp. 143–146.

[21] M. Klemettinen, H. Mannila, and H. Toivonen. “Rule Discovery in Telecommunication Alarm Data”. In: *J. Network Syst. Manage.* 7 (Dec. 1999), pp. 395–423. DOI: 10.1023/A:1018787815779.

[22] Kotlin. *Kotlin Programming Language*. https://kotlinlang.org/, Accessed on 19th February 2023.

[23] M. Lozonavu, M. Vlachou-Konchylaki, and V. Huang. “Relation discovery of mobile network alarms with sequential pattern mining”. In: *2017 International Conference on Computing, Networking and Communications (ICNC)* (2017), pp. 363–367. DOI: 10.1109/ICCNC.2017.7876155.

[24] A. Mazdziarz. “Alarm Correlation in Mobile Telecommunications Networks based on k-means Cluster Analysis Method”. In: *Journal of telecommunications and information technology 2* (2018), pp. 95–102.

[25] D. Moulavi, P. A Jaskowiak, R. Campello, A. Zimek, and J. Sander. “Density-Based Clustering Validation”. In: Apr. 2014. DOI: 10.1137/1.9781611973440.96.

[26] A. Musdholifah, S. Z. B. M. Hashim, and I. Wasito. “KNN-kernel based clustering for spatio-temporal database”. In: *International Conference on Computer and Communication Engineering (ICCCE’10)*. 2010, pp. 1–6. DOI: 10.1109/ICCCE.2010.5556805.

[27] F.-V. P., Z. M. He G., and N. M. “Discovering Alarm Correlation Rules for Network Fault Management”. In: *Service-Oriented Computing – ICSOC 2020 Workshops. ICSOC 2020.* (2020). DOI: https://doi.org/10.1007/978-3-030-76352-7_24.

[28] E. Polystar. *Virtual NOC*. https://elisapolystar.com/products/virtualnoc/, Accessed on 19th February 2023.

[29] S. Rao. “Operational Fault Detection in cellular wireless base-stations”. In: *IEEE Transactions on Network and Service Management* 3.2 (2006), pp. 1–11. DOI: 10.1109/TNSM.2006.4798311.

[30] J. Ren, B. Cai, and C. Hu. “Clustering over Data Streams Based on Grid Density and Index Tree”. In: *Journal of Convergence Information Technology* 6 (Jan. 2011), pp. 83–93. DOI: 10.4156/jcit.vol6.issue1.11.

[31] A. Sirotkin. *5G Radio Access Network Architecture: the Dark Side of 5G*. Hoboken, New Jersey, USA: Wiley-IEEE Press, 20120.

[32] Skicit. *Skicit learn BallTree*. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.BallTree.html, Accessed on 15th February 2023. 2007.

[33] A. Slalmi, H. Kharraz, R. Saadane, C. Hasna, A. Chehri, and G. Jeon. "Energy Efficiency Proposal for IoT Call Admission Control in 5G Network". In: *2019 15th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*. 2019, pp. 396–403. DOI: 10.1109/SITIS.2019.00070.

[34] Spring. *Spring Boot*. https://spring.io/, Accessed on 19th February 2023.

[35] M. Steinder and A. Sethi. "A survey of fault localization techniques in computer networks". In: *Science of Computer Programming* 53 (Nov. 2004), pp. 165–194. DOI: 10.1016/j.scico.2004.01.010.

[36] Sun Valley Networks. *What is a Network Operations Center?* https://www.sunnyvalley.io/docs/network-security-tutorials/what-is-network-operations-center, Accessed on 6th March 2018.

[37] D. Tasoulis, G. Ross, and N. Adams. "Visualising the Cluster Structure of Data Streams". In: vol. 4723. Sept. 2007, pp. 81–92. ISBN: 978-3-540-74824-3. DOI: 10.1007/978-3-540-74825-0_8.

[38] L. Tu and Y. Chen. "Stream Data Clustering Based on Grid Density and Attraction". In: *ACM Trans. Knowl. Discov. Data* 3.3 (July 2009). ISSN: 1556-4681. DOI: 10.1145/1552303.1552305. URL: https://doi.org/10.1145/1552303.1552305.

[39] K. Valtari and E. Vesterinen. "Automate the Network Operation Center". In: *[White paper]* (2020).

[40] F. Wang, F. W. X. Fan, and J. Liu. "Backup Battery Analysis and Allocation against Power Outage for Cellular Base Stations". In: *IEEE Transactions on Mobile Computing* 18.3 (520-533), p. 2019. DOI: 10.1109/TMC.2018.2842733.

[41] J. Wang, C. He, Y. Liu, G. Tian, I. Peng, J. Xing, X. Ruan, H. Xie, and F. L. Wang. "Efficient Alarm Behavior Analytics for Telecom Networks". In: *Information Sciences* 402 (Mar. 2017), pp. 1–14. DOI: 10.1016/j.ins.2017.03.020.

[42] M. J. Zaki and W. M. Jr. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. USA: Cambridge University Press, 2014. ISBN: 0521766338.

[43] G. Zargarian, L. Vassio, M. M. Munafo, and M. Mellia. "Mining Patterns in Mobile Network Logs". In: *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)* (2019), pp. 1–6.

[44] P. Zhao and L. Lai. “On the Convergence Rates of KNN Density Estimation”. In: *2021 IEEE International Symposium on Information Theory (ISIT)*. 2021, pp. 2840–2845. DOI: 10.1109/ISIT45174.2021.9518025.

[45] L. Zhong, F. J. K. Takano, Y. J. X. Wang, and S. Yamada. “Spatio-temporal data-driven analysis of mobile network availability during natural disasters”. In: *2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)* (2016), pp. 1–7. DOI: 10.1109/ICT-DM.2016.7857223.