Feasibility Analysis for Forecasting Inflow of Unplanned Work using Machine Learning Techniques

FAHAD SHAKEEL

SUPERVISOR
Prof. Andreas Prinz, UiA
Saad Abbas Abbasi, Ericsson AB

University of Agder, 2021
Faculty of Engineering and Science
Department of Engineering and Sciences

Master

Obligatorisk gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

1.	Vi erklærer herved at vår besvarelse er vårt eget arbeid, og at vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	Ja
2.	**Vi erklærer videre at denne besvarelsen:** • Ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. • Ikke refererer til andres arbeid uten at det er oppgitt. • Ikke refererer til eget tidligere arbeid uten at det er oppgitt. • Har alle referansene oppgitt i litteraturlisten. • Ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	Ja
3.	Vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§ 31.	Ja
4.	Vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.	Ja
5.	Vi er kjent med at Universitetet i Agder vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.	Ja
6.	Vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.	Ja

Publiseringsavtale

Fullmakt til elektronisk publisering av oppgaven Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).
Oppgaver som er unntatt offentlighet eller taushetsbelagt/konfidensiell vil ikke bli publisert.

Vi gir herved Universitetet i Agder en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:	Ja
Er oppgaven båndlagt (konfidensiell)?	Nei
Er oppgaven unntatt offentlighet?	Nei

Acknowledgements

I would like to thank my family, for always supporting me to pursue my studies, especially my father for constantly motivating me towards completion of my Masters degree. I would also like to thank my supervisors, Prof Andreas Prinz for always being available and helping me with his weekly sessions, and Saad Abbas Abbasi for helping me through out the process and giving his valuable input that eventually shaped this thesis.
Finally, a special thank you to my partner, Aafeen Asif Alam, for constantly being by my side, and encouraging me to get it all done within the time frame.

Fahad Shakeel, Stockholm, January 2022

Abstract

Ericsson Radio Software (RSW) works and develops radio network products for its customers, at the same time handling bugs or maintenance issues that may arise in its products. These issues need to be handled by the same teams, balancing development work along with the maintenance that may arise. High maintenance issues may arise causing the development timelines to be affected consequently affecting the overall project completion plans.

In this thesis, an attempt is made to look into and explore the usage of machine learning (ML) algorithms to help predict future workloads for the teams working to balance development work along with resolving maintenance issues that may arise to help teams make better and more accurate plans. In addition to looking at the feasibility of using ML algorithms for better and accurate prediction, it will attempt to look into what are the present methods and techniques employed by the organisation to cope with this issue and what are its merits and demerits.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter gives information to help give a context to the issue which this thesis addresses.

1.1 Background

The issue at hand here is that during planning of product development at Radio Software, there needs to be ample time dedicated to the provision for solving issues that may arise. These issues can arise not only in the radios being developed currently, but also previous products that have passed through the development phase and are currently under operation at the customer or otherwise.

1.2 Problem Statement

Whenever one such issue is reported or detected in one of the products, that issue is logged in the system against the issuance of a ticket. These logged reports are referred to as Trouble Reports, usually referred to as a (TR). The TR is then routed to the system to the appropriate organisation responsible for looking into the contents of the TR. Once it has reached the relevant organisation, the teams that have been involved with the product in question are called in to investigate the issue and come to a proposal of how this issue can be resolved. The team taking care of the TR needs to shelve responsibility from its on going development activity. This affects the overall plan for delivering the on going project.

In the absence of a team that can be related to the product, as can be possible for older or more smaller products, those TRs are made part of the organisation's central TR backlog. It is the combined responsibility of the different product development cross functional teams to make sure that they help take responsibility for these TRs and help bring them to their logical conclusion. A TR can end up in the central backlog also when it is unclear where the issue actually arises from. In this case, the developer working on the TR needs to first analyse where the actual issue lies and try to seek out the relevant person to resolve it. This can result in the TR being rerouted out of the organisation as well if it is determined that the issue lies in the scope of another organisation.

1.3 Report Structure

Chapter 2 provides information related to operation of the RSW organisation and the generation of the TRs. Chapter 3 discusses briefly the factors that affect TR resolution. Chapter 4 gives an overview of the buildup of the backlog. Chapter 5 looks at the different ML techniques considered for this thesis. Chapter 6 looks at the results of the ML experimentation.

Finally, the report is concluded with with some closing remarks and suggestions for future work.

Chapter 2

Workplace Mode of Operation

The mechanism in which the RSW organisation operates is analyzed, these analysis are presented with the frame of reference of TRs and its handling; and how these fit into the Ericsson workplace.

2.1 Development Teams' Ways of Working

The Radio Product development unit consists of several cross functional teams, each team containing 5-8 members, following the agile process framework. The team convenes every three weeks to plan for the following sprint, pulling in relevant tasks to the sprint based on priority.

There have been several approaches from the organisation to help teams efficiently reduce the central TR backlog. Previously every team was asked to resolve one TR during the course of each sprint. On occasions of high numbers of backlogged TRs this quota would be doubled to two (2) TRs.

The fundamental issue with this approach was that it is hard to predict the complexity of a given TR. Only after once a developer starts to investigate it, there can be some comments as to the amount of effort required in its resolution.

This meant that the teams would be spending an indefinite and varying amounts of time on TR resolution. A team that has been lucky could perhaps resolve a TR in a single day and be done with their quota. Whereas on the other hand, sometimes a single TR resolution can take months to get to the bottom of . This would lead to undue pressure on the teams as they were far behind their allotted quota but were in fact investing a whole lot of more time and effort in trying to find a solution. This can cause them to lose valuable time that was originally planned to be dedicated towards product development, delaying the product delivery plans as well.

The current mechanism placed in order to help teams prepare for this unknown workload, is that each team is asked to allocate 20-30% of their workload during their sprint plans towards TR resolution. This is a ballpark figure as the effort eventually needed can only be known after investigating it.

Inaccuracy is yet another issue found using this approach. It is possible that the workload is far greater than 20%, consequently causing the project plans to be further pushed forward. Whereas on the other hand, if the workload from TRs is lower than 20%, then the teams end up planning and performing to lower than the teams' capacity.

Here it must be stated that these planning mechanisms employed by the teams are only to help resolve the central backlog TRs of the organisation. In the event that TRs incoming are relevant to the team's previous or current work, they are obligated to take ownership of it. All of this unplanned work is often hard to predict and ends up slowing down existing and ongoing projects.
If there was a mechanism that can be fairly accurate in predicting the amount of Trouble Reports to expect, and the accompanying workload, it would help make the sprint plans of the teams a lot more accurate.

2.2 Sources of TRs

Trouble Reports can be generated from a number of sources. This may either be internally within Ericsson, or it can also be generated externally. External TRs usually are sourced from customers raising problems faced by them during operation of Ericsson equipment on their networks.

2.2.1 Internally Generated TRs

For internally generated TRs, there can still be a number of further places of origin.

Development Phase

During development, the software being developed is constantly tested by running it in the automated testing loops where the software is regularly tested for flaws in functionality. If a test case starts to fail on a specific radio several times, it can then be considered to be written as a Trouble Report. These TRs are usually generated by the same organisation that will resolve it.

Post-Development Lab Testing

After a radio has gone through its initial development, it is then possible for the labs to start testing its connection with the rest of the products it is supposed to operate with, such as Base-band and end user equipment. This testing is usually outside the Radio organisation, any TR generated here is then determined and routed to the appropriate organisation. If it is not a Radio TR it has to be rerouted to its relevant destination.

Customer Equipment Integration

Before a mobile operator starts to install and integrate a new radio product into its network, the usual route is that people from Ericsson start testing the product along with the operator equipment in a lab environment, usually at the customer premises. This helps find issues that may arise while running the product on the operator's network. These TRs that indicate integration issues are of higher priority, as they can directly impact final operation of the product. They are still considered as Ericsson generated TRs.

Live Network Initialisation

Once a product has started being installed onto the customer network, they start facing all of the real-world elements that may have been overlooked in the lab conditions. There can be several problems faced during this stage of integration into the operator's network. This is usually the part where most of the problems arise and can be a very busy and stressful time for the development teams, as both the volume of TRs generated at this point are significantly high, and so are their priority. These TRs can be both either customer generated or generated by Ericsson itself.

After a product has been integrated into a live network, and the network starts operating, if the network operator sees any problems in the operation of the product, the customer can then raise a Trouble Report with Ericsson. These TRs are usually of the highest priority as they signify a problem with products working on a live network, and any delay in their resolution can result in monetary loss for the mobile network operator.

Live Network Operation Statistical Feedback

Another scenario from which a trouble report may be generated also concerns the live networks of the customer. During operation of Ericsson products at the customer end, Ericsson regularly receives operational statistics from its products regarding their performance. Analyses on these statistics can unveil possible problems that are present and need to be resolved. These usually are issues that have not yet come to light as they haven't had any significant impact so far. The customer is also usually unaware of the existence of the problem, and the idea is to resolve the issue before it can cause some significant problems in the live mobile network.

2.3 Internal RSW TR Handling

The bulk of the TRs generated within the Radio Software organisation are the TRs based on issues that occur during the initial development and testing phase of the radio product. As this constitutes the major contributor of RSW TRs, the mechanism in place for this TR generation and resolution can be looked at further.

At the integration meetings arranged daily, situations, problems and hindrances faced during the development and testing phases by a host of parties ranging from developers, testers, CI teams/ DevOps team and Integration teams etc are raised, discussed and occasionally resolved.

In case of unresolved instances, the next course of action is to generate a TR against the problem, and serves to track progress and discussions related to the issue. This generated TR is then assigned to an appropriate developer to further analyse and resolve.

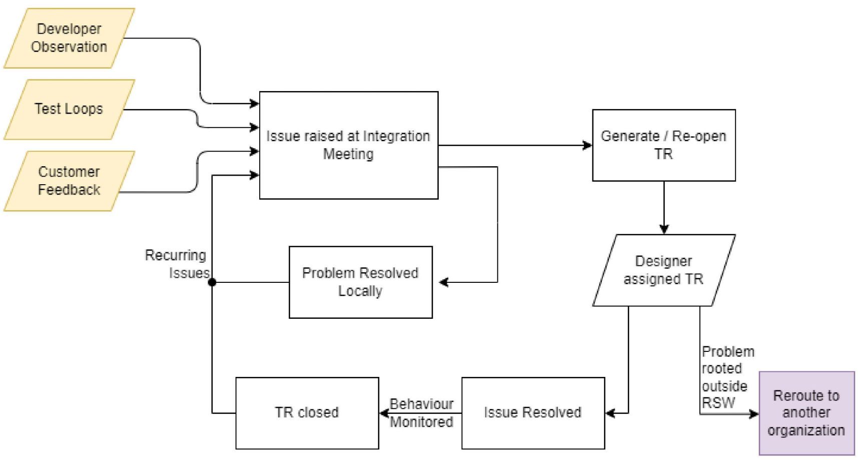


Figure 2.1: TR Handling within RSW

The range of time these TR resolutions require is very diverse. It can range from anything between half a day to over a month's worth of work. Once resolved, the TR is closed off and the designated developer proceeds to the next TR or development assignment planned.

The TR system in general lacks details on the complexity and depth of the situation at hand. The assignee is unaware of the effort and time required for the issue.

This limitation of the TR system bars other team members as well from understanding the depth of work and the time being dedicated to a task, unless a breakdown by the assignee is available after some deliberations.

Chapter 3

Factors Affecting Backlog Buildup

3.1 Effort

One of the problems in assigning the teams with a quota of Trouble Reports, is that using the metric of TRs tends to assume that each TR is equal in the amount of effort needed to resolve it. Each Trouble Report is dealing with a problem of a different nature, therefore the effort put in until a resolution is reached is also different.

A TR of rather less complexity should usually be easier and quicker to resolve, whereas if a problem of increasingly complex nature presents itself it would seem that it takes more time and is tougher to resolve.

This merely translates into a points system. One member of the organisation may have addressed several TR's in a week, and may present a good average on their turnover time while the other developer who may take several days of the week, on a problem of more complex nature may end up with a very low score on the turnover time.

If a solution can come up with a formula to work out this shortcoming, a host of problems may be avoided– including apprehension on the developer's end who could anticipate possible rebuking for a low TR average despite the magnitude of work put in.

3.1.1 Problem in Measuring Effort

For a developer, the effort put into resolving any matter can mostly be attributed in relation to the time spent by that developer into resolving it.

Unfortunately, this is an extremely subjective area which brings light to the problem that each developer is different and possesses a different level of knowledge and experience of working with the product. An issue that takes a developer, that is relatively new to the area, around 5 working days might be resolved by an experienced developer, in 1.5 working days. This huge difference is down to the fact that the complexity of a product at Ericsson is usually quite high and in order to understand the context of a problem, one needs to go through several documents and traverse the several layers of the code that exist. It is usually not possible for a developer to comprehend the code implementation swiftly and then reach a conclusion to the root of the problem, whereas for someone already familiar with the product implementation and its code structure, this journey can be travelled more quickly.

Considering this, one can come to the conclusion that if a binary division is made where a developer is either an experienced one or an inexperienced one, these two categories can have a separate estimation of effort. Where an estimation for an experienced developer is made and an algebraic expression can relate it to the additional time needed by an inexperienced developer.

This simple solution has a lot of flaws in it. For starters, it is quite difficult to categorise a developer as experienced regardless of the number of years they may have worked. This is due to the fact that a person may have spent quite a few years in the department and have developed quite a deep knowledge base in an area. But if they are to start work in a different area, due to the huge difference in each area, they no longer possess the same depth in knowledge. They might have a better idea of the area and its structure, and be able to make sense of the information quicker. But even then it will still take some time for this “experienced” developer to resolve the issue as compared to a more seasoned developer of the area.

Eventually these titles will come to affect their inevitable evaluation and assessment, therefore, it would be unlikely that an experienced employee would be willing to diminish their status due to a change in assignment.

3.1.2 Perception of Efficiency

A big ethical issue with attaching an outsider estimate to a TR is that this estimate, in a corporate environment that always wishes to improve turnaround times, will eventually be the yardstick used to measure the efficiency of the staff who is in charge of resolution of that TR. This will end up becoming another stress instigator for the assignee of the TR.

3.2 Dated Backlog Items

As the TRs are distributed and resolved across the organisation, there are certain outliers which either due to their complexity or perhaps are dependent on cooperation from several organisations to bring to their conclusion, end up staying unresolved for an extended period of time.

The general perception of these TRs is that the older TRs are usually more complex and more difficult to resolve. But as this can not be visible to anyone other than the assignee, it is usually regarded as another TR that should be taken care of in the same time as others. This makes it very unfavourable for a developer to assign themselves such a TR resulting in an even longer time in the backlog.

Chapter 4

Maintenance Backlog Buildup

In this chapter, a look into the different elements that contribute to the buildup of TRs in the RSW backlog. As well as the different experiences by the organisation of factors that contribute to the overall maintenance workload.

4.1 Change of Product Generation

Every few years, technological advancements require the release of a new generation of mobile radio products. These products are usually vastly more capable than their predecessors and utilise the advancements in telecommunication technologies that were not able to be integrated into previous products due to technology, requirement, compatibility and other restrictions.

These new generations have a lot of new technology built into them that have not had the chance to be tested in the real world. So whenever a new generation of radios are being rolled out, it is safe to say that it is expected to run into a host of problems.

4.1.1 Technological Evolution vs Technological Revolution

The new generation products can be broadly categorised into two parts. They are either built on top of the existing technology or framework, or the previous approach has been abandoned and an entirely new structure and framework has been built to support the needs of the newer technology.

For instance, the Third Generation of mobile radio products or 3G as they are commonly known by, were an advanced version of the 2G mobile radio products. They were considered to have an evolved approach. Whereas in 4G, most of the previous approach was done away with and new sets of technologies were developed to support the needs of the new LTE products.

If the new generation of products is an evolution of the existing technology, it can be expected to perform better in the real world as part of its technology is already stable. The part that is newly evolved can be expected to face problems upon operations.

The situation with revolutionary generational products is entirely different. There can be high amounts of instability in the entire system and the issues that arise can pertain to any region whereas the whole system has been untested in the real world environment. This can cause issues to arise in very complex areas and these may require a lot of time and effort to resolve. Another reason, is that the developers working on the system are also usually new to the complexities of the new environment. Comparatively, all the personnel are also inexperienced with the system in operation.

4.2 Issue Occurrence Trend

The frequency of how many issues a product faces can be dependent on several factors. This can be on whether the product is part of a new line of radios utilising a new platform or perhaps a non-standard customer specific product is being developed that is unlike other existing radios. All of this can cause an increase in the number of issues that arise in the radio, and subsequently, the number of TRs that need to be resolved.

To break it down, two primary scenarios shall be considered.

4.2.1 Scenario 1:

First is the case of a radio product that is to be developed for a big customer with a huge network. The idea is that a huge number of radios will be deployed by the mobile operator across its network.

For big deployments like these, smaller test portions are selected where deployment starts. These small deployments are made part of the live network gradually as they start to appear stable in their test runs.

The expected build up of TRs in this case is predicted to increase in numbers as more radios are deployed. As the network grows, the TRs also grow exponentially. Many new issues are uncovered simultaneously across several locations. In many instances, several TRs appearing to be different issues may actually stem from the same basic problem. This will result in multiple people to look into the same problem perhaps across different organisations. With time it would be realised that the issue stems from the same source and happens to be duplicates of one another.

These can be very demanding times from the teams. Since the product has dedicated development teams, all of these issues are expected to be resolved by the relevant teams. They can quickly build up with several issues being worked on simultaneously and still have a backlog of trouble reports to catch up with.

As the issues continue to be resolved by the teams; this growth in TR numbers starts to slow down, eventually topping off from where quite quickly the number of TRs starts to drop off. Gradually, the radio starts to move towards a more stable state where a few trouble reports remain.

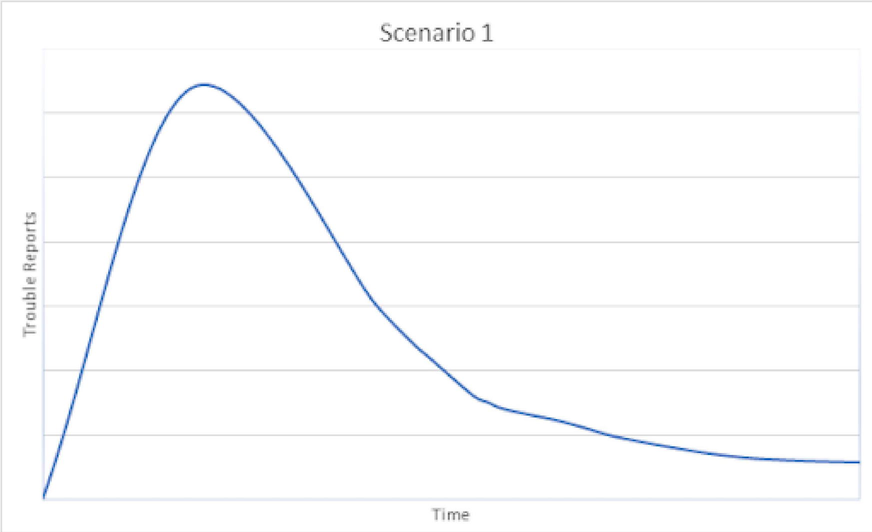


Figure 4.1: Representative graph to illustrate trend in Scenario 1

4.2.2 Scenario 2:

The second scenario is for a radio product that is to be developed for a relatively smaller network. This means that the number of radios to be produced will be fewer and there might still be differences in the areas where the radio products are used but generally the conditions in which they are to be operated are not drastically different.

Unlike the previous scenario, in this case most of the radios will become part of the radio network at the same time. At the most, they may be divided into two halves and then each half is on boarded to the live network.

The incoming trouble report frequency here is relatively lower due to the lower number of active devices. There will also be instances of duplicate trouble reports here, the frequency of that happening is lower and are usually limited to two single duplications. These duplicated TRs are also spotted earlier and quicker.

The build up of the TR backlog here is gradual but it continues to grow over an extended time. But unlike the first scenario, this number does not drop off as quickly. On the contrary the peak here tends to continue for a longer period and even after that, the TR numbers, while reducing, do continue to sustain themselves over a long period.

Although the scenario does not result in such extreme pressures on the teams as the first one due to the huge backlog of TRs, instead in this situation a much longer commitment and engagement from the development team is required towards the product. That means that the teams involved with the products that are designed for smaller networks need to be engaged with the product for a longer time before they are able to start work on the next one. And even after they do start work on the next project, it must be expected of them to have to spend some portion of their time towards possible incoming TRs.

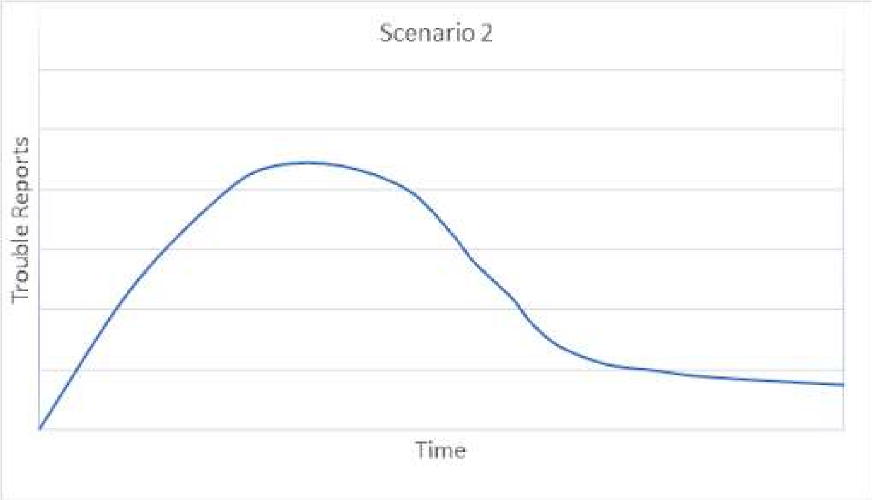


Figure 4.2: Representative graph to illustrate trend in Scenario 2

Based on these primary scenarios, two further scenarios are possible if after the initial client, follow up customers for the same product install the equipment into their networks

4.2.3 Scenario 3:

This scenario is when a product, which is already operational with an operator with a sizable network, is installed by other operators as well.

The number of issues that the Radio product encounters faces a slight surge as a subsequent operator network is added to its customer list. These are usually not a big number of TRs that are generated in these subsequent installations as the major issues faced by the radio product have already been ironed out.

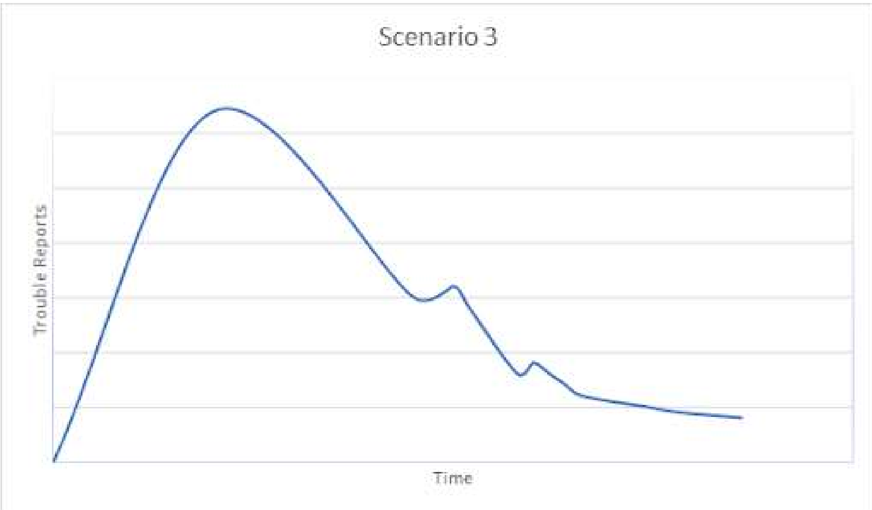


Figure 4.3: Representative graph to illustrate trend in Scenario 3

4.2.4 Scenario 4:

Similar to the previous case, when more customers are added to a product that had previously only been used by a customer in their limited network, the adding of newer operators causes fluctuations in the previously lowering rate of issues.

As newer network operators are added, the radio product gets exposed to newer environment variables and challenges in the host networks. These new operating conditions result in problems to be reported and can thus require more time from the teams to spend on resolving issues of the product before they are available to be moved to the next development project.

As observed, it usually happens that follow up customers for such a product also look to install the radio products in smaller networks.

Figure 4.4: Representative graph to illustrate trend in Scenario 4

4.3 Subjective Variation Based on Customers

There has been a significant difference in the way mobile operators in different geographical locations of the world conceive something to be a problem. This cultural factor often plays a major role in the type and amount of complains that are received by Ericsson for a particular product.For example, one mobile operator might be happy as long as the product operates and completes the job it has, but another customer might be more specific in the way that operation takes place by the radio product.

For this purpose, certain markets have a general image of the manner in which mobile operators run their respective businesses. For example, operators in Japan are quite well known for the fact that they require the radios to operate flawlessly. An occasional restart of any radio might not be a problem by an American operator, but in the case of Japan, a single restart or operational freeze will be reported, and then deep follow ups will be conducted going down to the root cause of the issue. This makes the TR backlog to be more filled up if the customer is in Japan for a relative problem. Furthermore, each TR may require more effort as it is common for the operator to ask for individual logs and the difference in the log file before and after resolution of the problem.

Chapter 5

Machine Learning Techniques Considered for this Thesis

5.1 Basics of ML

Machine Learning is the term used to describe a set of techniques and methods used to implement algorithms that help make predictions and organise input data according to their common traits. The goal of Machine Learning is that to study and improve mathematical models that enable decision making in the absence of complete data and knowledge of all factors [2].

5.2 K Nearest Network

KNN is a classification algorithm. The idea is that based on characteristics, data can be classified.

Once the characteristics are available, they are mapped out in order so that items with similar characteristics fall closer to each other, and as these characters differ the distance between the neighbours starts to increase.

As a new item needs to be classified, its classification is decided by the neighbourhood in which its characteristics land them. The K is the parameter of the number of nearest neighbours. The majority of neighbours in this K is how the classification of this new item is decided.

The idea to use KNN in this thesis was that it could help in mapping out the different radios at Ericsson based on their differing characteristics. Then a prediction can be made on them based on the probability that if a certain radio, on a certain platform, with x and y number of branches is to be developed, what should the expected number of TRs be for it based on information of similar radios in the past.

5.2.1 Gradient Boosting

Gradient Boosting is one of the ensemble learning algorithms. In an ensemble, several underlying models work collectively to give a more favourable model.

Here several weak learners are used together with the idea that together they can perform as a strong model. Therefore, it is an iterative method where the data points go through the different trees that are created based on the performance of the previous trees, optimising it one step at a time for bringing a better prediction.

The idea in gradient boosting that sets it apart is that in Gradient Boosting the subsequent models are focused on minimising the error value. By focusing on the errors, the algorithm gives more importance to misclassified data and tries to place them correctly thereby reducing the error. Each successive tree created tries to minimise the error where the previous tree performed poorly.

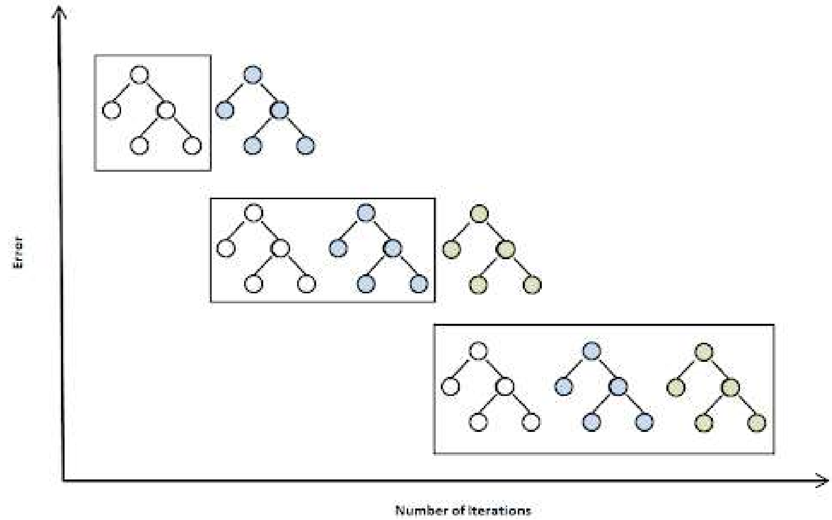


Figure 5.1: Ensemble learning in Gradient Boosting

During experimentation, it was felt that Gradient Boosting was better suited to the scenario that was faced. The most common Gradient Boosting algorithm for classification is LSBoost. The LSBoost Algorithm works by keeping the sum of the squares of the ensembles to a minimum. To do that, it creates a new tree at each iteration and uses the base learner and learning rate to keep the difference between the predicted output and actual value to a minimum. As iterations are increased, the chances of over-fitting increases[1]. To avoid this situation, the parameters such as learning rate and number of iterations need to be tuned accordingly.

Chapter 6

Experimental Results

6.1 Data Collection

The issues that arise in most Ericsson systems as Trouble Reports are stored in a common database, with TRs that arise from across the whole company. Here, one can access the various TRs matching a given search query.

At the start of this thesis, the assumption for this work was that the central database would make the collection of data quite streamlined and swift. As the work for the thesis progressed, it was realised that data collection would not be as simple and straightforward as first suspected.

For the purpose of this thesis, the radio products that were developed by Ericsson Radio Software in the last ten years were collected. The TR data for these products needed to be collected from the central database; this was a particularly arduous activity as it was not possible to automate this process and, particulars for each radio produced in the past decade had to be individually entered to create queries for the relevant data to be retrieved. This data was then combed through to find the relevant TRs for the purpose of this thesis.

Towards the end, the final data set consisted of all the Ericsson Radio products developed at the different sites within Sweden, Canada and China. Data collection became one of the major tasks to be completed for the thesis.

More relevant information regarding the individual radios products was gathered to help provide more data points that can be used to find relations between technical specifications and the probability that issues may arise in the operation. The final attributes decided upon for the data set were as follows:

Product	No.of RX Branches	No.of TX Branches	Frequency Band	Product No.	R-State	Release Date	Site	No.of TRs on record
A	2	2	B2	KRC 123456/1	R1-A	2010-08-22	Stockholm, Sweden	4
B	0	1	B8	KRC 123457/2	R1-B	2015-06-01	Beijing, China	2
C	1	0	B40	KRC 123458/3	R1-D	2018-03-15	Ottawa, Canada	0
D	6	1	B85	KRC 123459/4	R1-E	2020-09-01	Lund, Sweden	11

Table 6.1: Data set representation

Here in table 6.1 all the products are to serve as the input variables, whereas the No. of TRs is to be predicted by the Machine learning algorithm.

6.1.1 Data Set Attributes

Product:

The Product is the name that has been used by the company to market a particular model developed. The model number can also denote some information regarding the Tx and Rx branches in use by the particular product.

No of RX Branches:

A basic single branch radio to receive transmissions will use at least one RX branch to receive data transmission over a certain frequency. The Rx branch provides vital components that process data received from the antenna to the radio. Having more branches increases the capacity and gives more operational flexibility to the radio.

No of TX Branches:

A basic single branch radio to use for transmissions will use at least one TX branch to transmit data over a certain frequency. The Tx branch provides vital components that process data received from the antenna to the radio. Having more branches increases the capacity and gives more operational flexibility to the radio.

Frequency Band:

A radio product is designed to operate within certain frequency bands, these are usually customer and equipment specific. Operating in a certain frequency band can possess its own complexities and problems. For example, radio products operating in the higher frequency bands have more focused beams than have limited range as higher frequency signals have limited penetration power.

Product Number:

A product number is used to mark certain models of a radio model that have some differing product specification. These product number can also be used to denote the use of alternate design or parts.

R-state:

For each Radio that has a certain product number, it is possible for it to have some hardware revisions. These hardware revisions are denoted as R-states.

Release Date:

This is the date when a radio is developed to the point where it can be released to the customer, mostly on a preliminary basis.

Site:

This denotes the primary development site that was responsible for driving the development of the radio product. Other sites may have also been involved as well.
The environment used for this thesis was selected based on the existing tools that the team hosting the thesis had used up till that point. The host team had good experience of the use

of MATLAB for their daily tasks and needs. As MATLAB was also a good platform for use in Machine Learning, it was decided that the same tool will be used in this thesis as well.

6.2 Output Results

KNN

Initially the approach made was to attempt and utilise the K Nearest Neighbour algorithm for the purpose of this thesis. The idea behind using KNN was that the algorithm uses the characteristics provided in the dataset to classify each individual item. This would lead to the use of this algorithm to start categorising the data that is present regarding the radio products and help place them in groups that will signify the probability of the product running into a problem.

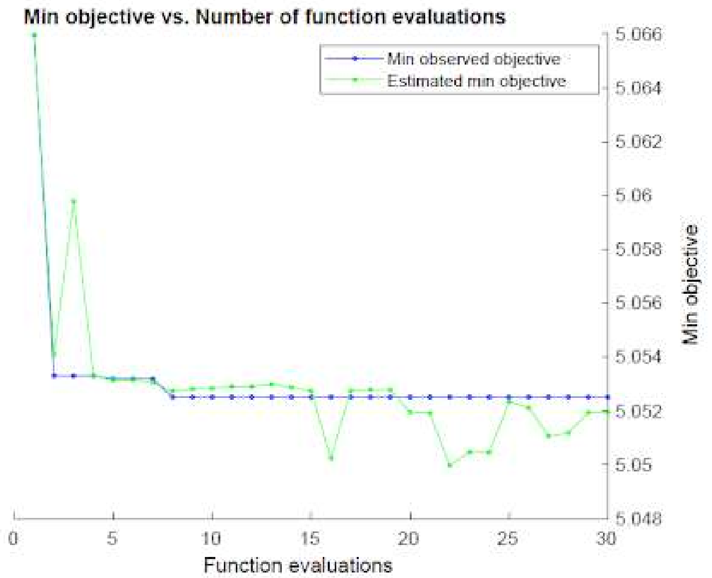
In essence, the idea was that the algorithm, based on past data provided, will create a classification that will sort radios based on its product specifications respectively. Any new product that is planned to be developed, can be sorted by the algorithm, and the placement into the respective classification will determine the likelihood of how frequently it may face issues and consequently, how frequently the development teams can expect to face TRs due to the specific radio product.

Upon running the KNN algorithm, it was realised that this particular algorithm is not suited to the task currently at hand. The tuning of parameters and other steps were not helpful as the result received did not signify any improvement in the prediction model.The assessment regarding use of KNN was unfortunately, incorrect as it did not yield a favourable result.

With some discussion among the developers present at Ericsson, the suggestion was made that the nature of this thesis would mandate use of an ensemble learning algorithm. Thus the transition to the use of Gradient Boosting method was made.

Gradient Boosting

The best results were achieved by utilising the Gradient Boosting method. LSBoost algorithm was used to run experimentation on our datasets. Several iterations with differing parameter values were used.



Upon tuning of the parameters, the following values fetched the best results for the given dataset:
Leaf Size: 30
Learning Cycles: 3000
LearnRate: 0.05

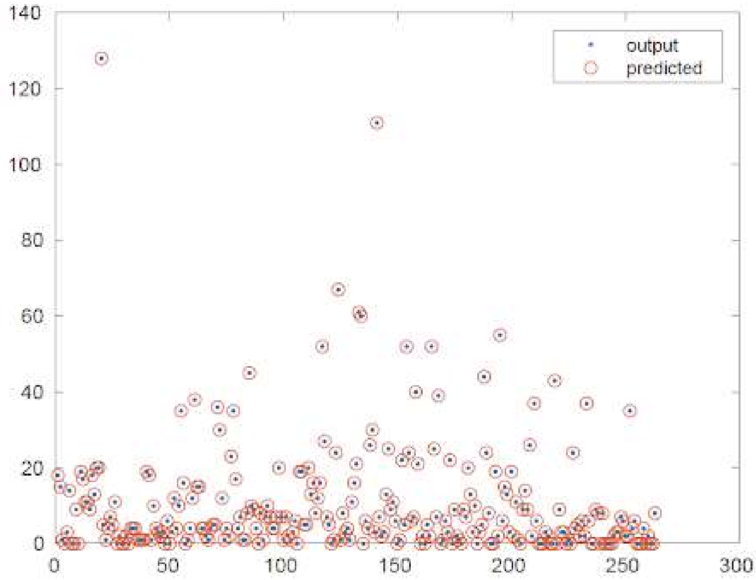


Figure 6.1: Mapping output to prediction

Initial experimentation were made with the data split as 80% training data and 20% test data. These were adjusted to 70% training data to 30% test data. Eventually the optimum ratio achieved between the two was at an equal of 50% training data against a 50% test data. With these set parameters, an accuracy of around 87-92% was reached.

Chapter 7

Conclusion

The initial impressions regarding the nature of the trouble reports were revisited once work had started and a more detailed look at the working of the organisation was taken.

The metric used, where Trouble Reports are the sole point of reference and their binary status of either being open or closed, appear to be a limiting factor for the actual developers who are responsible for resolution of the problem.

Merely predicting the occurrence of Trouble Report can be helpful in charting of the performance of a product, but as far as the development team's ability to plan for these TRs, it will not be very useful without the information of how much expected time and effort is needed for the resolution of the problem. The planning of the team cannot be very accurate if it lacks the vital information of how long the expected effort is in this particular problem.

The reason a metric of possible effort required, in the resolution of the problem does not exist currently, is the predicament that a person of technical expertise is needed in the first place to ascertain the complexity of the issue at hand and required effort to resolve it.

The suggestion put forward at the conclusion of this thesis, as demonstrated during the course of work, it is possible to create a model that can predict the occurrence of issues in particular radios that will result in generation of a Trouble Report. But the current approach of the team to allocate a particular slice of the time available to cater to those issues will not be vastly improved on adaption of such a TR prediction model. This is due to the fact that the number of expected TRs does not assist much in terms of planning of time spent as effort on diagnosis and resolution.

The suggestion put forth as possible future work as succession to the thesis at hand; a prediction model for an average time spent by a team member be developed. This model can come up with a rough time estimate that is part of the TR information, combined with the current TR prediction model, the effort prediction can give a better picture to the teams of what may be expected of them by the organisation in the upcoming sprints.
To tackle the perception that this effort estimate might have on the efficiency of the developer, the estimate can be made transparent to the management by making it as part of the details that usually are only of interest to the developer resolving the issue.

Chapter 8

Bibliography

[1] Agapitos, Alexandros Brabazon, Anthony O'Neill, Michael. (2017). *Regularised Gradient Boosting for Financial Time-series Modelling.* Computational Management Science. 14. 10.1007/s10287-017-0280-y.

[2] Bonaccorso, Giuseppe. (2017). *Machine Learning Algorithms.* Packt Publishing Ltd

Tarbani, N. (2021). *How the Gradient Boosting Algorithm works?* Retrieved November 15, 2021 from https://www.analyticsvidhya.com/blog/2021/04/how-the-gradient-boosting-algorithm-works/

Abbasi, S. (2021). *Behavioral Modeling of Power Amplifiers with Machine Learning on Multi Carrier and Multi Band Scenarios.* Chalmers University of Technology