Webology (ISSN: 1735-188X)
Volume 18, Number 5, 2021

Support For Telecommunication Alarming System Through Ticket Trouble Prediction

G. VIJAYASREE[1] , Dr. K. SUJITH[2]

[1]Ph.D. Research Scholar, Department of Computer Science, Annai College of Arts & Science (Affiliated to Bharathidasan University, Tiruchirappalli), Kovilacheri, Kumbakonam, Thanjavur, Tamilnadu, India.

[2]Associate Professor and Research Supervisor, PG and Research Department of Computer Science and Application (MCA), Annai College of Arts and Science (Affiliated to Bharathidasan University, Tiruchirappalli), Kovilacheri, Kumbakonam, Thanjavur, Tamilnadu, India.

ABSTRACT

A rise in the occurrence of faults from various telecom network services and infrastructures has been detected with the expansion of the overall network. With today's advanced alarm management systems, there is a significant increase in the challenge of managing alarms using manual or rules-based techniques, especially when there is a need for highly experienced individuals to be deployed. As network size and complexity expand dramatically, it is becoming imperative to use data-driven approaches to recognize semantically comparable alarms produced by heterogeneous network components from several suppliers. The approach presented in this project utilizes Alert Management Systems to create a troubleshooting prediction model powered by data. A time and feature engineering function extraction are implemented in order to increase performance. Data collected from the primary Italian telecom operator has supported the model development and validation. Trouble Ticket prediction reduced the number of false positives that were seen in the experimental data.

Keywords: Ticket Trouble Prediction, Alarming System, Telecommunication.

1. INTRODUCTION

To make things faster and cheaper, the system has been developed to decrease the time and money spent on alert handling. The method produced good results when critical warnings were identified that required the opening of a TT This will help telecoms providers to improve their receptivity in the management of a large number of alerts. However, with a one-hour judgement delay, About 28 percent of the True- Positive alerts missed models. The missed warnings are, however, halved by the same time delay as the NOC experts. Also, supervised techniques may require a significant number of training examples to adapt to new technologies with differing features.

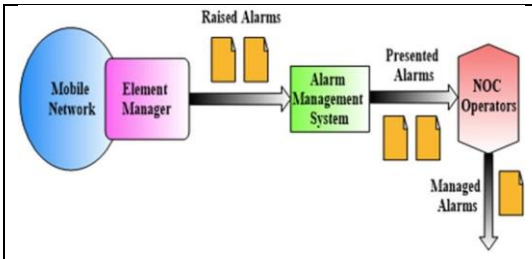


Fig. 1: (a) Current

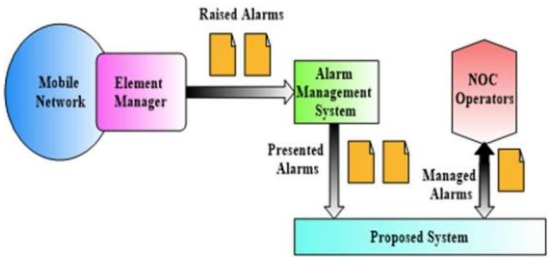


Fig. 1: (b) Proposed

Finally, inspite the low computing expense, the suggested feature extraction approach employing TWFE achieved a perceptible improvement in alert categorization. As a result, we believe that this method may be used in Data stream-based Industry 4.0 applications.

NOC operators will be better able to manage a high number of alarm streams because to the improved handling of alarms provided by automated TTP. This means that the proposed method is beneficial to improve service providers' efficiency and competitiveness by decreasing the time and money spent on alert handling. Finally, while we've done a lot here, here are the biggest achievements:

- Before the TT Triaging procedure is finished by NOC operators, the opening decision algorithm is utilized to reduce false-positive alarm floods caused by a number of AMS problems. Mobile telecommunications networks.
- Integrating time-windowing and feature engineering techniques with feature augmentation and extraction methods in previously handled linked alarm streams.

Decision delay-time varia.tions are used to simulate TTP behavior. Better prediction of which alarm will go off sooner in the life cycle reduces the time it takes to fix the defects, which, in turn, minimizes the costs associated with service disruptions. When that is finished, the rest of the project will be completed.

2. BACKGROUND

This part addresses basic topics such as AMS operating principles, the associated work in the handling of fault alarms and the technologies utilized in our research.

2.1 System of AMS

The Alarm management system stores, characterizes, filters, root cause This includes not only finding the errors that have occurred, but also analyzing how many alerts, messages, and event logs are gathered across many networks. The schematic in Figure 1a depicts the overall design of an AMS. The manager of a supplier's networks gets events occurring in a heterogeneous network, which have been flagged by other suppliers, and passes them through to AMS, where they are treated by means of a rule- based argument. The processed alarms, alarms, provide findings that are presented to NOC operators, where genuine alarms are kept under control via TT opening. The goal of an automated raised alarm flood control system is to identify floods at a lower-level and deal with them by employing domain information based on expert-defined Three resources are essential: rules, an ontology, and topological databases.

Although statistical and machine learning-based models are frequently mentioned in the literature as methods for managing big data [6], [11], [12], [14]–[16], [18], [19], AMS solutions utilise rule- based algorithms instead for most of their essential functions [2]. Incorporating domain expertise is an effective way to enhance rule- based algorithms and make choices more accurate. However, algorithms based on rules do not manage unanticipated circumstances and huge heterogeneous systems [11], [12]. As a result, most flood filtering systems available do not manage false positives adequately. [2], [12]. Thousands of notifications, thus a day still burden NOC operators with judgements to reject or silence False-positive alarms before correct TT is assigned.

TABLE I: False-positive alarm filtration Related works

Solution	Purpose	ML Model	Features	T W	Target Alarms	Domain
[6], [12]	ALCR	YES	AOS	YES	Raise d	Telecom
[18], [19]	ALCR	YES	AOS, AA	YES	Raise d	Telecom
[13], [17]	ALCR	YES	AOS	YES	Raise d	Telecom
[15]	ALCR	NO	AOS	YES	Raise d	Telecom
[11]	ALCR	NO	AOS	YES	Presented	Telecom
[16]	ALCR	YES	AA	NO	Presented	Telecom
[14]	ALCR	YES	AA	NO	Presented	Telecom
[28]	TTP	YES	TD	NO	Ticketed	Telecom
[5]	TTP	YES	AA	NO	Presented	Telecom
Proposed	TTP	YES	AA	YES	Presented	Telecom

See Table I for some of the critical information on how to deal with the FPW flood using automated approaches. Considering the primary jobs in autonomous alarm systems, the applications may be broadly divided into two groups: those that require extensive computational capabilities and those that don't.

- Redundant alerts are being consolidated with Alarm Correlation (ALCR), which was designed to do just that. Typically, it is used to reduce relevant and bogus warnings (such as manager nodes generating them) that are distributed by other nodes.
- Analysis for Alarm Root Cause (ALRC) is used to determine the source of an alarm defect or root cause.
- TT Prediction (TTP) classifies the alarm presented in order to establish whether or not the TT will be opened. When it was suggested in [5], that TTP be used to suppress false alarm warnings, it was suggested in other articles that TTP be utilized to avoid repetitive ticketing for IT customer premises equipment alerts. How to Silence False Positive Alarms: Here are recommended based on their TT description TTPs Even so, this might aid technicians who resolve alarms and, therefore, the NOC operators, as the biggest problem when it comes to alarms flooding is earlier on in the ticketing process.
- The TT classification (TTC), depending on necessary problem resolution, priority or associated technical teams, assigns the suitable TT category to the True positive alarms.
- TT Resolution Recommendation (TTRR) proposes techniques for ticketing alarms.

2.2 LGBM

LGBM has been employed to construct the proposed TT prediction models. This technology, called LightGBM, uses a hybrid method that combines decision tree-based machine learning with distributed computing. Although new developments in the area of profound learning techniques have been up-to- date with vision, language and speech, the GBDT is now popular for large-scale production models Due to their speed, interpretability, lesser demand for computer resources and precision in the training for mixed feature types. In an ensemble of GBDT models, decision-making booms are trained sequentially by adjustment of the negative steps in each model (also called residual errors). XGBoost, LightGBM, and CatBoost are among the current state-of-the-art implementations of GBDTs. To achieve a balance between the need to reduce the quantity of data samples and maintaining the level of precision, lightGBM utilizes flatbed learning. By surpassing other boosting methods, LightGBM proved its speed accuracy and memory consumption in diverse information sets.

2.3 Hyper Parameter Tuning

Hyper parameters are required before training. It is a tedious and critical process to tune hyper parameters, as algorithms' success is dependent on these settings. Several approaches for automating the tuning process have therefore been developed and used. However, Bayesian optimizations approaches have been found to be more rapid and surpass human specialists in choosing hyper parameters on particular data sets. Therefore, for hyper parameters modification, we chose to use an efficient technique based on Bayesian Optimism, i.e. Hyper opt]. The data set utilized in this investigation is described. The information was collected from the TIM, Italy's leading telecommunications provider, data collector and correlation engine. This data set focuses on alarm generation on the various mobile networks, including 2G, 3G, and LTE access networks. This dataset contains Alarm Stream events about 7.5 million, which are recorded for every five minutes between 2018 and 2019. The number of distinct alerts that may be composed is around 900,000. The remainder of the events occur during real-time scenes with references to historically recorded alarms, or when there is an alert characteristic or Trouble Ticket change. Table II includes a description of the most important features of each alarm record. NOC operators are provided more information about how the alert is handled by the NOC for alarms handled by TT. These include the acknowledgement status, ticket status, the closing time and closing remark. Based on the alarm severity attribute, two primary groups are mentioned, i.e., critical and major (they each account for about 48 percent of the presented alarms Major alarms are intended to generate Trouble Tickets after the alarm has been activated. Severity and the time at which the first alarm occurs provide helpful insight when making decisions on which alert to focus on in the triage process [10]. In addition, only approximately 10 per cent of the presented alerts with a trouble ticket were managed by NOC operators in the data provided. Operators are considerably inconvenient in the face of the AMS alarm overload to find adequate alerts manually.

3. METHODOLOGY

This is a first look at an ML-based TT prediction system.

Attribute	Description

Feature-1	Alarm type
Feature-2	AMS is filled inCorrelation indicator
Feature-3	Element Manager
Feature-4	Time in which the alarm is first observedby the AMS
Feature-5	Time in which the alarm is last observed by the AMS
Feature-6	Severity level of thealarm
Feature-7	Alarm root causestatus
Feature-8	Site name
Feature-9	Site category
Feature-10	Description of apparatus function
Feature-11	Type of the network
Feature-12	Fault source
Feature-13	Possible probable causes which generated the alarm
Feature-14	Operation view flag
Feature-15	Number of attachedchildren alarms
Feature-16	Number of repeatedoccurrences of the alarm
Feature-17	Minutes between the first and lastoccurrence of thealarm
Target	TT class label from TTidentifier

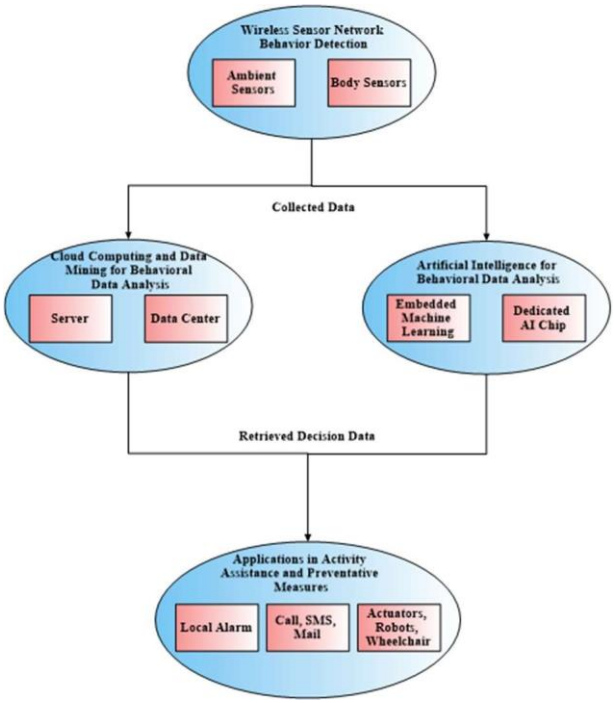


Fig no 2: The proposed AI-based TT prediction structure

The major modules of the system suggested are shown in the architecture diagram of Figure 2. To carry out the preparation of data sets in the pre-processing block, you use the approach described here. The inputs for this block are alerts provided by the Alarm management system.

Then the processed alarms are passed to the extraction block, where the features are enlarged to facilitate the model learning process. The resulting features are split into a training and test set. The training set is used to fit and adjust the ML models by means of the training package. In the TT, these models help determine the likely results of test-driven prediction and performance evaluation of novel alarms.

3.1 Pre-Processing

This section discusses techniques for data cleansing, function selection and encoding on alarm workouts and assay sets. Feature selection: In the process, the field specialists began to advise us to select the most significant characteristics. The first 43 characteristics gave us concern and TT qualities (Operators not available in real time during alarm intervention). This is further clarified by the analytical methods utilized, such as Chi-Square test and Pearson correlation and Association Mining (Cramer's V and Theil's U) for numerical characteristics, as well as Association Mining (Cramer's V and Theil's U) for classifying features. The methodologies of data analyses have helped to quantify the correlation/association of feature selection procedures and predictability of the target variable characteristics, eventually lowering the total of the important attributes to 18 (17 from alarm and 1 from labelling TTs) listed in Table II.

By far the most often utilised categorical feature encoding method, however, is Integer and One- hot encoding. The weaknesses in this system become more problematic when the feature set has both a large number of category variables and when the categories cannot be ordered. A tree constructed on one hot features, in particular for high-cardinality categorical features, is imbalanced and must grow extremely deep to attain excellent accuracy [36]. We were unable to detect any increased classification accuracy by utilizing advanced word insertion approaches such as Word2Vec. To accommodate categorical response processing, LightGBM integrates a built-in mechanism for classifying categories on the basis of at each split, the training aim.

Tokenization: Some alarm features contain descriptive strings created by the computer. Algorithms for text mining are often used to extract keywords while suppressing foreign phrases to categories such textual descriptions. Tokenization is generally used in text normalization to increase extraction of keyword while preserving semantic system [14], [21]. In some of our information characteristics, information like IP address, node identity, date time, serial number, site name, vendor name, and numbers are referenced. normalization of text tokens. The block inputs are obtained via AMS alerts. The model learning procedure becomes easier once the characteristics have been enhanced. Anonymized tokens like IP address, serial number, node ID, vendor, date and time, and location were utilized to clean up messy information by simply interchanging them with text strings like serialno, ipaddress, nodeid, vendor, datetime, number, and location. classifying: The class target labelling was created utilizing the TT prediction model training technique based upon the TT alert data-set variables. Each alert contains one or more events in real-time that are associated with TT, who is assigned, or who is in the current stream (in some situations). Since the TT state of the last instance of the alerts is used for tagging, the TT status of the alerts is retained for later usage. Every previous alert that occurred is then apportioned to the labels. This technique enables the TT

prediction models to be trained in their early stage, leveraging the features of the alarms. The target variable's class labels are listed below:

- Positive class, TT alarm(s): whether TT is issued or opened. No TT alarm: if the alarm is not sent using TT. This section discusses techniques for data cleansing, function selection and encoding on alarm workouts and assay sets. Feature selection: The procedure for determining the most important aspects began by consulting domain experts, who supplied us with the initial 43 attributes from whom 33 and 10 are alert numbers and TT attributes (which operators do not get during real-time alarm intervention). The resultant group was nonetheless further investigated by analysis of data methods like numerical connection of Pearson and categorical characteristics using Cramer V and Theil's U. The methodologies of data analyses have helped to quantify the correlation/association of feature selection procedures and predictability of the target variable characteristics, eventually lowering the total of the important attributes to 18 (17 from alarm and 1 from labelling TTs) listed in Table II.
- Categorical function encoding: although categorical encoding techniques are among the most prevalent, integrate and one-hot encoding are opting-out because of semantine change, functional explosion, workout duration and accuracy limitations. Categorical fields that have large numbers and/or quantities in their set, and have not been strictly ordinal, may have faults that are much more significant. In the event of high cardinality, a tree with a shallow root system to be precise, it has to develop exceedingly deep. Although word integration, such as Word2Vec, adds to the processing costs, categorizations of our experiments have not benefited. We utilized LightGBM's built-in categorical handling method in this work, which classifies the categories Based on each split training objective.
- Tokenization: Certain alarms include descriptive text created by the machine. Text mining algorithms are commonly used to extract keywords while removing foreign terms for categorization. One technique for improving keyword extraction while maintaining semantic structures is tokenization, which is frequently employed in text normalization [14], [21]. In our collection, some of the textual alert characteristics relate to devices, networks, locations, vendors, and dates and times. To clean up such data, keywords such as IP address, serial number, node ID, vendor, date and time, and so on were replaced with textual tokens that are just identifiers.

LabelingTT-based background approach to The TT fields for the TT prediction model training targets in the alert data set were used. All warnings involve a sequence of distinct occurrences that occur in real time, during which TT is either issued or withdrawn. Thus, to indicate that the warnings have reappeared, the TT statuses of the latest alert instance are utilized. The notifications are allocated their labels in reverse chronological order to all prior alerts. This method of tagging TT prediction models trains the models on the alarm features during the early phases of alarm development. Below you will see a list of class labels for the target variable.

Positive class, TT alarm(s): whether TT is issued or opened. No TT alarm: if the alarm is not sent using TT.

3.2 Feature Extraction

we included both a TTP model and feature set increase, with the former based on prediction functions for features of the alerts and the latter based on engineering characteristics from past historical alarms, which were detailed in Table III. The fundamental notion is that the same root causes are most often triggered by fault warnings coming from the same source in a short time. We thus provide a functional extraction technique, which uses this notion and increases TTP models with enhanced feature sets gained from prior alerts performance. A thorough explanation of the approach to the proposed TTP system is shown in Figure 3. When taking on a project, two important phases that are generally included are feature-engineering (FE) and time-windowing (TW) which are sometimes referred to as TWFE (see Algorithm 1). Algorithm 1 first samples the historical alarm streams that originated from the same source as the target alert (lines 2-7 in). Target alarm A collects all prior alarms, including all the alarms that happened inside the TW range (see line 5). From the hAlarms, alerts relating to Nh, rAlarms produced from the same S, the origin of A, are then filtered (see line 7). rAlarmsFE (see lines 8-9) the engineering characteristics of A are obtained from the feature filtering procedure of the alarms. In order to combine experts' topic expertise, computational difficulties, and model accuracy, the time- window size (TW = 15 days) and number of historic alerts previously managed were learnt by heuristics (Nh = 15).

While the fact that the sequences of rAlarms together create pseudo time series data are certainly useful, the importance of such historical alarms should not be confused with the traditional meanings of events. FE serves as a bridge between rAlarms and a new, smaller set of features called rAlarmsFE. These additional facts are that rAlarms with correlación-based function selection are only employed with the most relevant categorial characteristics, namely sF, and importance analysis for the feature (see Table III). The one-hot coding feature expansion in function engineering provides additional consideration when determining sF for low cardinality characteristics. . (see line 7). Features 5, 6, and 7 of the TWFE, with the exception of Feature 5, which correlates to rD, may be utilised to Find relative time time duration to target alert A for the occurrence of rAlarms. Furthermore, including features from rAlarms' TT properties, which were previously handled linked alarms, lets us incorporate TT properties from rAlarms (TWFE Feature4 and -6). Furthermore, Figure 4 depicts how the TWFE operates, as well as the key responsibilities described below:

a) Before utilizing TWFE, properties retrieved from target alarm A are prepared . From this point on, these features will be known as without TW.

b) This retrieves history alarms, as well as historical alarms, such as alerts that happened inside the time frame TW, before the alarm A occurrence time.

c) The source of the fault alert is discovered in location attributes and network element identification of the target alarm.

This operation filters historical alarms (also known as rAlarms), which have been made from the same source as the target alarm, so that the timeline contains just the same source history alarm.

TABLE II: In related history alarms select the features

Selected Features	Description
TWFE Feature-1	Correlation indicator filled by the AMS
TWFE Feature-2	Severity level of the alarm
TWFE Feature-3	Operation view flag
TWFE Feature-4	Alarm acknowledgment status
TWFE Feature-5	Relative time of occurrence from the current alarm
TWFE Feature-6	Trouble Ticket opening execution status

For each value of categorization, it provides the counts of catV warnings (block 2 in Figure 4, see line 16 in Algorithm 1). features are then merged using a weighting process to create features for every catV aluminum (See Figure 4 for Line 18 in Algorithm 1, Block 3 and 4). The weighing function is placed on line 12 in the rD of rAlarms. The extracted characteristically characteristics, Cat V aluminum, are combined to create the developed characteristics for one specific characteristic F, rAlarmsFEF (see line 19 in Algorithm 1). This stage culminates with the ultimate construction of the Engineering FE,Alarms FE, as a result of the full-extraction of allattributes from the sF and their integration into asingle dataset (see line 20 in Algorithm 1). The characteristics of A, which have no relationship to TW, are also appended to the alarms retrievedby FE to produce the features with TWFE. best described by the word "fsc" and delivered in (1):i=Nh

$$fsc(v, occScores, rD) = X\ occScores(vi,v) \times fw(rDi) \quad (1)\ i=1$$

The occScores(vi,v) is an alarm v in rAlarms (2), whereas fw is an alarm function (3). For example, the RDi is the relative time for each relative alarm, which is simply called rAlarms. In this equation, K and B are constant normalizing factors, whereas bias is a constant-valued source of non-uniformity. When the target alarm sounds, the most appropriate historical alarm sounds, too. [6]. Setting the constant k and b controls the weighting sensitivity, and thus influences the uniqueness of the feature score. In our experiments, our values were arrived at by employing heuristics. An alternative approach to time-series prediction (i.e., using temporal significance weighting) has been proposed. According to the model, good predictions of past occurrences should be given more weight.

3.3 Model Training

This algorithm, which predicts whether a specific alert will be managed via TT or not, is called the Trouble Ticket prediction system.

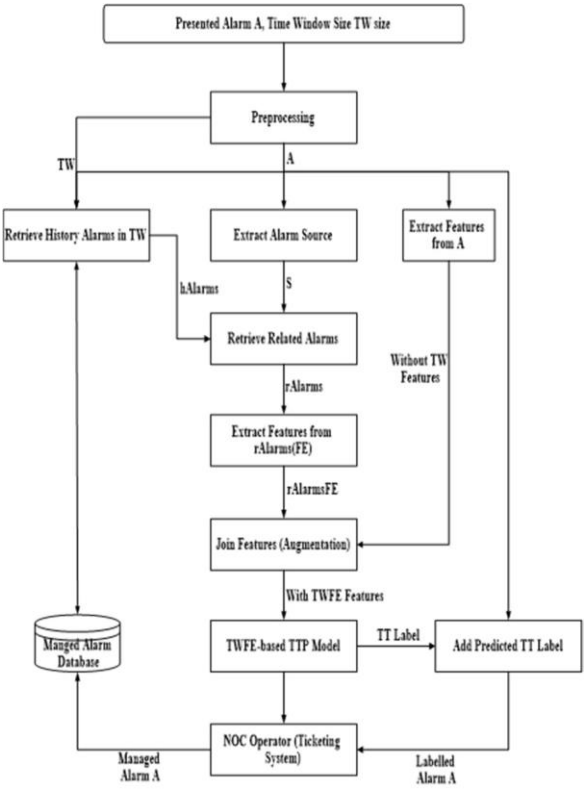


Fig. 3: Flowchart of time-windowing feature- engineering

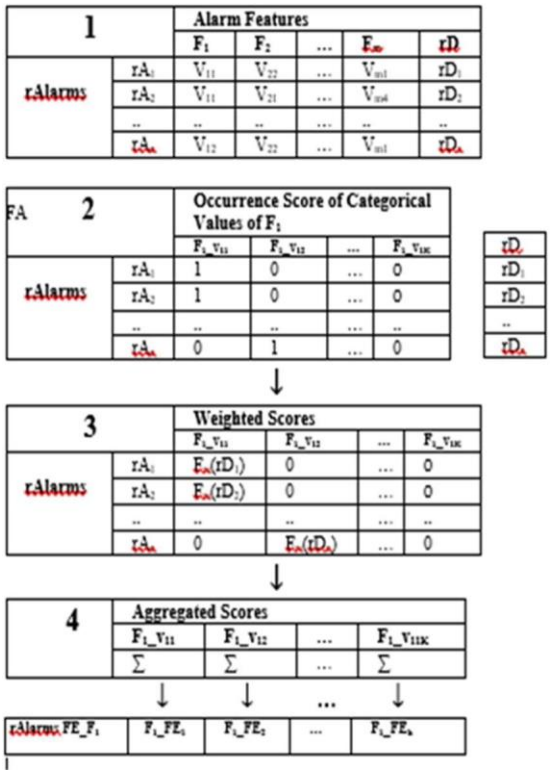


Fig. 4: Feature extracted using feature- engineering

It is carried out via training ML models for alert categorization. Therefore, the alarm classification models trained in categorization of the TT and TT alerts under a range of

situations are utilised to show the effectiveness of the approaches presented. In addition, the data set contains every five minutes information about each alert, such as the current status of the alarm. Since several models have been built and tested to provide TT prediction based on delays models that categories alarms that come within 60 minutes of the first arrival, different models will be employed, and training and testing will be conducted on a different set of samples.

TABLE III: labels distribution Target

Class Label	All (%)	Training (%)	Test (%)
Alarms with TT	9.30	8.60	11.14
Alarms without TT	90.70	91.40	88.86

For the training and preparation of test sets, the time window structure for functions and separate time series without shuffling is adopted. The initial four months of training were used out of the six-month alarm set (approximately 70%) for the suggested templates, and the remaining two month (approximately 30%) were spent on training.

Tuning hyperparameters: Tuning hyperparameters typically becomes permissive when a large number of parameters or candidate values are available even with an effective guided search procedure. Therefore, it is necessary to focus on a limited set of factors which have a considerable effect on model performance to get high-performance models. We have selected a number of key parameters, established their ideal values and evaluated the training set model to obtain the cross-validation score. The criterion for picking control parameters based on their capacity to enhance speed, accuracy and specificity, to choose the most accurate, quick and not overfit controls. Most parameter tuning searches use a classifier (estimator), a parameter space, a sampling strategy, and a cross-validation scheme. The following parameters are now a part of LightGBM: boosting, metric, leaf data minimum, leaf data maximum, depth learning rate, feature sampling fraction, bagging fraction, boost rounding count. Another issue with our training data is the fact that the majority of it is made up of negative examples. This bias causes our classifier to place more emphasis on the negative examples in the data. To reduce the bias of our training data, the AUC (Recipient Operating Characteristic Curve) is used as a cost-effective function done using other cost-effective functions. All of our tests were conducted using the identical hyperparameters settings, which were derived via the technique outlined above. Feature significance analysis: helping to determine the overall feature importance when developing machine learning models makes significant contributions to the overall model interpretability, which helps in facilitating model adoption into real-world systems. The LightGBM is based on a split (how often the feature in a model is utilised) and a gain-based (weight of the feature) method (total gains of splits which use the feature). The use of such measurements may nevertheless be problematic for all purposes, as the measurements are created exclusively on the basis of the trainings and do not show how significant the features of unseen data are. Due to the aforesaid problem, it was suggested in literature that the test set

performance of permutation-based features is high. In addition, the use of the Permanent Character Importance form to determine how a model has changed its sensitivity to random permutation in function values is another way to assess the feature relevance of trained models (PPI). An intuitive model-agnostic approach. The random permutation of the functionality in the trial set shows the actual potential of the PFI. The prediction and post-processing of the learning model is put into this block to generate fresh test data predictions. The findings are retained and sent to the post-processing scripts, which they use to conduct evaluations and to show them properly. The anticipated TT class labels and confidence ratings are delivered following decision-making time and alert intensity.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Here, the results of the experiments conducted with the recommended techniques are described. Compare the binary classification of alarms for troubleshooting alarms by determining how well each system classifies alars in respect of how many troubleshooting tickets are predicted without Time Window (without W) in a certain time window and how well each system classifies alarms using Time Functional Engineering Window (with TWFE). Unless otherwise stated, the reported findings are accessible on the test set. The previous section describes the use of GBDT with AUC as a cost function and early-stop to minimise overfitting, all used in binary LightGBM classification training. Measuring and assessing performance of classifiers using such measures as Precision, Recall, F1 score (F1), AUC, Precision- Confusion Matrix, Recall curve, and False Positive Rate is common practice. In addition, we've also utilized Average Precision (AP) to gauge the effectiveness of different machine learning methods. where R I and P I are weighted averages of the accuracy and reminder values attained by each threshold, with the change of recall as weight from the preceding threshold.

Precision and Recall at ith Threshold. Moreover, two types of alert severity, critical and major, have distinct sensitivities to the time of the ticket choice. In these severity situations, the debate also involves the performance of the models. Finally, comparing the effectiveness of different machine learning (ML) tools' trained models on two alternative algorithms is provided as a test to measure the competence of the suggested TWFE and TTP model. Trouble Ticket Alarm Classification Trained classifiers using the real time alarm stream and various delay durations were evaluated. In addition, the classifiers trained for test setup with a delay of 60 minutes were chosen to explain further the efficiency advantages of the proposed feature engineering technique.

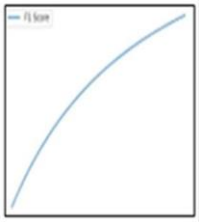
1) classifiers based on Time-delay: The explanation describes how to create period- delay-based classifiers, which use a five-minute inter-classification delay time to measure the influence of ticketing delay after the alarms first occurrence. The Figures 5-6 demonstrate the delay time-based classifier performance. Ten sets of repeated trials with random seeds have been performed for each experiment. The validation- set (10 percent) was left unchanged for each trial during training, and on both TW and TWFE. 95% Confidence Interval plot.

The performance of classification rises during the delay period for both feature sets. The greatest percentage increase was the delay in predicting the TT alerts which took place between 10 and 20 minutes. After you set the alarm, the first half-hour will be spent dealing

with the most important alarms. Our feature important assessment also revealed that the relevance of Feature-17 (i.e. Feature-17) in the TT prediction grows as the time increases to achieve the objective. This recommended TTP strategy's biggest success is that the False Positive Alarms (which account for 93.6% of alerts) have been substantially decreased and the True Positive Alarms (which account for 80% of alarms) have been increased by a factor of two. The performance of the classifier improved further with the addition of the with TWFE approach, as compared to the without TWFE technique. However, trained models with the Extra Hop plugin yielded scores that increased to [0.600, 0.663] with a standard deviation of [0.002, 0.004] (see Figure 5). the absolute change in F1 varies between [0.055, 0.075] with a standard deviation of [0.005, 0.012] (see Figure 6). By using the TWFE technique, the mean AUC increased from 0.902 to 0.930 with a standard deviation of 0.002. The significant finding is that the Critical Alarms' F1 reaches [0.647, 0.727] when TWFE is in use. In comparison, the gain over TW is [0.051, 0.083]. The testing data show that alarm-based feature extraction was able to boost performance while expecting a choice to open TT when using the time displays.

2) Nominal classifiers: The part above demonstrated the predictive strength of the alarm in TT prediction categorization improves as the alerts continue to stay longer While withTWFE's performance rises over withoutTW remains consistent. Alert time is one of these characteristics, since TT alerts are typically shorter than TT alerts. After an hour, only about 40 percent of the tick alarms have their TT. The most serious system alerts also highlight issues that must be addressed within hours. The models trained for the 'single test' results are consequently classified as the nominal classificators for the upcoming performance comparison at 60 minutes late. (Figure 7 - 8). In the accuracy, reminder and F1 scores, TWFE surpassed TW without TW with an increase of 0.08, 0.04 and 0.07, while a fake positive rate dropped from 9.18% to 6.37%. It is important to analyse the performance of the recommended TTP models on the balance between the ability to detect true-positive alerts and to eliminate false- positive alerts at different thresholds. The models were therefore evaluated in a number other ways, including the area below the curve (AUC) and the recall curve. The model has achieved a metric value of 0.945 with a gain of 0.021 with the usage of training weights for TWFE. If the objective F1 is achievable, the Precision recall curve is usually used to create a T-threshold to attain and recall the required precision. The precision reminder curve in figure 7 is approximated by the F1 contour lines and the true precision reminder curve. T annotations are presented on the reminder curve, whereas F1 annotations are displayed on the reminder curve. The curves shown in Figure 7b with TWFE even increased the performance at T 6 = 0.5. The T-values that yield maxF1 are close to T = 0,5 with roughly equal numbers are illustrated by the F1 recall curve and F1 contour. The results of this experiment show that the models are correctly calibrated and demonstrate the assertion. Figure 7 and Figure 8 illustrate that precise recall leverages are derived as assessed by changes in With TWFE accuracy by reducing incorrect positive predictions (reduce by about 9 percent). Simultaneously, the accuracy of the negative class rises by 3%. Figure 9 displays the average PFI scores for the top 10 important characteristics after ten PFI trials per test set feature. While there are good characteristics of the likelihood and alert length of Feature 13 (the likely causes), the

qualities of the past alarms preceded by TWFE have put the features in the top 10. This demonstrates that the speed increase is due to the extraction of new relevant predictability functionality which has previously been covered by historical alarm features while the TWFE feature was also added.



	Precision	Recall	F1-Score	Difference
0	0.8	0.1	0.17778	0.07778
1	0.8	0.2	0.32000	0.12000
2	0.8	0.3	0.43636	0.13636
3	0.8	0.4	0.53333	0.13333
4	0.8	0.5	0.61538	0.11538
5	0.8	0.6	0.68571	0.08571
6	0.8	0.7	0.74687	0.04667
7	0.8	0.8	0.80000	0.00000
8	0.8	0.9	0.84706	-0.06294
9	0.8	1.0	0.83289	-0.11111

Fig no :6 ΔF1 score of the time-delay

We observed that in the concluding part, the PFI score values are related with the functional analysis provided. A study of a pair of associations using a U-test and a Pearson correlation investigation. Feature-13 has the strongest link with the target variable when the nominal and continuous correlation is considered. You can easily identify the key features with approaches like these before a model is training, but to comprehend the model properly once it's learned, you have to find the salient features. In our method the feature analysis is paired with PFI to assist us create a better model step by step (reverse).

Comparison with Benchmark Models performances: We used additional ML techniques to do TTP modelling for the suggested model. The selection of tools is based on their strong practicality in managing classes imbalances, various categorical characteristics, huge data sets, and high cardinality. Categorical feature inputs are available only in a handful of the programmers, such as LightGBM and CatBoost. This has been done to make more ML instruments available, such as, XGBoost, Random Forest and Deep Neural Network (DNN).

5. CONCLUSION

This study utilises machine learning and feature increase to develop an issue of mobile network alerts using a problem ticket prediction system. The suggested supply system has been established to help service providers minimise the time and expenditures involved with alert management. This study utilises machine learning and feature increase to develop an issue of mobile network alerts using a problem ticket prediction system. The suggested supply system has been established to help service providers minimise the time and expenditures involved with alert management. There were positive results from the new methodology that is able to classify the alerts that need the TT to be opened. Because of this, the proposed models will aid telecommunications companies in managing huge numbers of warnings. While the models did catch over three-quarters of the True-Positive alerts within one hour, the models missed more than 28% of the True-1-hour decision delay positive alarms Regardless, compared to NOC professionals, who were also performing the same task, missed alarms were cut in half. This method also reduces the necessity to retrain the models of the TTPs as new network technologies are introduced, as they provide considerably different characteristics. The recommended TWFE functional extraction technique was successful in improving the

classification of the alerts. The usage of technology in stream-based data applications may be found in the Industry 4.0 data stream. 1-hour decision delay positive alarms Regardless, compared to NOC professionals, who were also performing the same task, missed alarms were cut in half. This method also reduces the necessity to retrain the models of the TTPs as new network technologies are introduced, as they provide considerably different characteristics. The recommended TWFE functional extraction technique was successful in improving the classification of the alerts. In the Industry 4.0 data stream, the use of technology may also be seen in stream-based data applications.

REFERENCES

1. Metaheuristics in Telecommunication Systems: Network Design, Routing, and Allocation Problems: Stephanie Alvarez Fernandez;Angel A. Juan;Jésica de Armas Adrián;Daniel Guerreiro e Silva;Daniel Riera Terrén, IEEE Systems Journal_2018.
2. Dependability Integration in Cloud- hosted Telecommunication Services: Wiem Abderrahim;Zied Choukair, IEEE Transactions on Dependable and Secure Computing_2019.
3. Supporting Telecommunication Alarm ManagementSystemWith Trouble Ticket Prediction: Mulugeta Weldezgina Asres;Million Abayneh Mengistu;Pino Castrogiovanni;Lorenzo Bottaccioli;Enrico Macii;Edoardo Patti;Andrea Acquaviva, IEEE Transactions on Industrial Informatics_2021.
4. Lightning Propagation Analysis on Telecommunication Towers Above the Perfect Ground Using Full-Wave Time Domain PEEC Method: Hongcai Chen;Yang Zhang;Yaping Du;Qingsha S. Cheng, IEEE Transactions on Electromagnetic Compatibility_2019.
5. Lightning Transient Analysis of Telecommunication System With a Tubular Tower: Hongcai Chen;Yang Zhang;Yaping Du;Qingsha S. Cheng, IEEE Access_2018.
6. Reconstruction of Optical Pulse Intensity and Phase Based on SPM Spectra Measurements in Microstructured Tellurite Fiber in Telecommunication Range: Elena A. Anashkina;Maxim Y. Koptev;Alexey V. Andrianov;Vitaly V. Dorofeev;Surinder Singh;Gerd Leuchs;Arkady V. Kim, Journal of Lightwave Technology¬¬_2019.
7. eFRADIR: An Enhanced FRAmework for DIsaster Resilience: Alija Pašić;Rita Girão-Silva;Ferenc Mogyorósi;Balázs Vass;Teresa Gomes;Péter Babarczi;Péter Revisnyei;János Tapolcai;Jacek Rak, IEEE Access_2021.
8. A Seq2Seq Learning Approach for Link Quality Estimation Based on System Metrics in WSNs: Weiwei Miao;Zhengyang Ding;Hao Tang;Zeng Zeng;Mingxuan Zhang;Shaqian Zhang, IEEE Access_2021.
9. Content Centric Cross-Layer Scheduling for Industrial IoT Applications Using 6TiSCH: Yichao Jin;Usman Raza;Adnan Aijaz;Mahesh Sooriyabandara;Sedat Gormus, IEEE Access_2018.
10. An Adaptive Bidirectional Multibeam High-Altitude Platforms Aeronautical Telecommunication Network Using Dual Concentric Conical Arrays: Yasser Albagory, IEEE Access_2021.

11. Switch and Inverter Based Hybrid Precoding Algorithm for mmWave Massive MIMO System: Analysis on Sum-Rate and Energy-Efficiency: Mengqian Tian;Jianing Zhang;Yu Zhao;Lianjun Yuan;Jie Yang;Guan Gui, IEEE Access_2019.
12. Side Channel Attack-Aware Resource Allocation for URLLC and eMBB Slices in 5G RAN: Yajie Li;Yongli Zhao;Jun Li;Jiawei Zhang;Xiaosong Yu;Jie Zhang, IEEE Access_2020.
13. IT and Multi-layer Online Resource Allocation and Offline Planning in Metropolitan Networks: Miquel Garrich;José-Luis Romero- Gázquez;Francisco-Javier Moreno-Muro;Manuel Hernández-Bastida;María- Victoria Bueno Delgado;Anderson Bravalheri;Navdeep Uniyal;Abubakar Siddique Muqaddas;Reza Nejabati;Ramon Casellas;Óscar González de Dios;Pablo Pavón Mariño, Journal of Lightwave Technology_2020.
14. An Analysis of Social Networks Based on Tera-Scale Telecommunication Datasets: Hidayet Aksu;IBRAHIM KORPEOGLU;Özgür Ulusoy, IEEE Transactions on Emerging Topics in Computing_2019.
15. Online scheduling protocol design for energy-efficient TWDM-OLT: Sourav Dutta;Dibbendu Roy;Chayan Bhar;Goutam Das, IEEE/OSA Journal of Optical Communications and Networking_2018.
16. Deep Learning for Selection Between RF and VLC Bands in Device-to-Device Communication: Mehyar Najla;Pavel Mach;Zdenek Becvar, IEEE Wireless Communications Letters_2020.
17. Evaluating Reliability/Survivability of Capacitated Wireless Networks: Ozgur Kabadurmus;Alice E. Smith, IEEE Transactions on Reliability_2018.
18. Tidal-traffic-aware routing and spectrum allocation in elastic optical networks: Boyuan Yan;Yongli Zhao;Xiaosong Yu;Wei Wang;Yu Wu;Ying Wang;Jie Zhang, IEEE/OSA Journal of Optical Communications and Networking_2018.
19. Refractive Index and Temperature Sensing Based on an Optoelectronic Oscillator Incorporating a Fabry–Perot Fiber Bragg Grating: Yuguang Yang;Muguang Wang;Ya Shen;Yu Tang;Jing Zhang;Yue Wu;Shiying Xiao;Jingxuan Liu;Buzheng Wei;Qi Ding;Shuisheng Jian, IEEE Photonics Journal_2018.
20. Overview of development and regulatory aspects of high altitude platform system: Dong Zhou;Sheng Gao;Ruiqi Liu;Feifei Gao;Mohsen Guizani, Intelligent and Converged Networks_2020.
21. Estimation of Lightning Incidence to Telecommunication Towers: Pantelis N. Mikropoulos;Thomas E. Tsovilis, IEEE Transactions on Electromagnetic Compatibility_2019.
22. An Accurate Texture Complexity Estimation for Quality-Enhanced and Secure Image Steganography: Ayesha Saeed;Fawad;Muhammad Jamil Khan;Humayun Shahid;Syeda Iffat Naqvi;Muhammad Ali Riaz;Mansoor Shaukat Khan;Yasar Amin, IEEE Access_2020.
23. Adaptive Protection of Scientific Backbone Networks Using Machine Learning: Ferenc Mogyorósi;Alija Pašić;Richard Cziva;Péter Revisnyei;Zsolt Kenesi;János Tapolcai, IEEE Transactions on Network and Service Management_2021.

24. DRL-Based Energy-Efficient Resource Allocation Frameworks for Uplink NOMA Systems: Xiaoming Wang;Yuhan Zhang;Ruijuan Shen;Youyun Xu;Fu-Chun Zheng, IEEE Internet of Things Journal_2020.
25. Priyadharshini, D., Gopinath, R., and Poornappriya.T.S. (2020). A fuzzy MCDM approach for measuring the business impact of employee selection, International Journal of Management (IJM), 11(7), 1769-1775.
26. Poornappriya,T.S., & Gopinath, R. (2020). Rice Plant Disease Identification using Artificial Intelligence Approaches, International Journal of Electrical Engineering and Technology, 11(10), 392-402.
27. Gopinath, R., and Poornappriya.T.S. (2020). An Analysis of Human Resource Development Practices in Small Scales Startups, International Journal of Advanced Research in Engineering and Technology (IJARET), 11(11), 2475-2483.
28. Poornappriya,T.S., & Gopinath, R. (2020). Application of Machine Learning Techniques for Improving Learning Disabilities, International Journal of Electrical Engineering and Technology (IJEET), 11(10), 403-411.
29. Gopinath, R. (2016). Industrial Relations impact with Job Satisfaction using SEM Model with special reference to BSNL Employees in three different SSAs, Indian Journal of Research, 5(7), 94-97.
30. Gopinath, R. (2016). Is the Employee Health and Safety Related to Job Satisfaction? An Inquiry into BSNL Employees with Special Reference in three different SSAs Using Modeling, IOSR Journal of Business and Management (IOSR-JBM), 18(7-IV), 135-139.
31. Gopinath, R. (2016). A Study on Appraisal and Reward in BSNL with special reference to Job Satisfaction in three different SSAs using Modeling, Indian Journal of Applied Research, 6(7), 275-278.
32. Gopinath, R. (2016). Is Promotion and Transfer helps to Employee's Job Satisfaction? An Empirical Study at BSNL with special reference in three different SSAs using modelling, Asian Journal of Management Research, 6(4), 277-285.
33. Gopinath, R. (2016). Impact of HRD to Job Satisfaction with Special References to BSNL Employees in Three Different SSAs using SEM Model, International Journal of Management (IJM), 7(5), 1-9.
34. Gopinath, R. (2016). A Study on Recruitment and Selection in BSNL with special reference to Job Satisfaction in three different SSAs using SEM Modeling, International Journal of Scientific Research, 5(7), 71-74.
35. Gopinath, R. (2016). HRD Factor Managing People Influence to Job Satisfaction With Special Reference to BSNL Employees in Three Different SSAs Using Modeling, Global Journal of Research Analysis, 5(7), 323-326.