A rule induction approach to forecasting critical alarms in a telecommunication network

A Rule Induction Approach to Forecasting Critical Alarms in a Telecommunication Network

Chris Wrench*, Frederic Stahl*†, Giuseppe Di Fatta*, Vidhyalakshmi Karthikeyan‡, Detlef Nauck‡
*Department of Computer Science, University of Reading, PO BOX 225, Whiteknights, Reading, RG6 6AY
C.Wrench@pgr.reading.ac.uk
F.T.Stahl@reading.ac.uk
G.DiFatta@reading.ac.uk
†German Research Center for Artificial Intelligence GmbH (DFKI)
DFKI Laboratory Niedersachsen, Marine Perception
Marie-Curie-Strae 1, 26129 Oldenburg, Germany
‡BT Research and Innovation, Adastral Park, IP5 3RE, UK
Vidhyalakshmi.Karthikeyan@bt.com
Detlef.Nauck@bt.com

***Abstract*—This paper proposes a white box method of predicting critical alarms so they can be mitigated and understood by engineers. Forecasting these alarms will avoid outages and maintain the agreed service level which is beneficial to both the provider of telecommunication services and the consumers. The paper evaluates several item set mining approaches on a set of alarms of the British Telecom (BT) national telecommunication network and proposes a novel transformation of the data to enable the discovery of patterns undetectable by current item set mining approaches. The result is a method for rule induction that predicts alarms with high precision using a wide range of features.**

I. Introduction

The goal of this work is to develop a method that allows the expressive forecasting of events in a telecommunication network. Telecommunications pay a vital role in our day to day lives and faults can be costly to both the provider and the consumer [28]. Amongst other reasons, service outages impact customer satisfaction and increase churn rates [9], fines are issued if promised service levels are not sustained and a level of service is required for national security [5]. It is important, therefore, to keep outages to a minimum.

Predicting the most severe events ahead of time would aid in both of these aspects. Additional benefits include:

- Given sufficient warning there could be an automated or manual intervention to avoid an outage
- A prediction would give an engineer a head start in remedying the issue.
- It would lighten the workload of both network monitors and network engineers.
- There are large potential savings from a reduction in the work load of engineers, call centers and through the reduction of churn rates [30].

In pursuit of this goal a range of techniques from the literature were applied with little success. An alternative system is needed to address these shortcomings. The developed system must forecast events ahead of time using human readable rules.

The contributions from this paper are a method of transforming algorithms designed for classification into forecasters and an empirical evaluation of new and existing algorithms when applied to the BT data set. This is a collection of alarms generated over a period of 2 months that represent warnings and faults on devices of varying severity. The data has several features that describe both the nature of the alarm and the device to which it pertains. Loosely speaking, the outcome of the research is a rule based method to forecast network alarms with a high precision.

Section II contains a description of the telecommunication domain and other approaches from the literature that have been developed for similar problems. This includes a number of approaches utilising different forms of item set mining as this is prolific throughout the field. Section III describes and evaluates some experiments using existing item set mining approaches on the BT data set. The problems with these approaches lead to the development of a new method that is laid out in Section IV and evaluated in Section V. Finally some discussion, conclusions and further work can be found in Section VI.

II. Rule Induction and Telecommunication Networks

In this section a brief description of telecommunication networks is provided as well as some of the existing works on forecasting events within the domain. The purpose of this review is to identify methods that are applicable to the BT data set.

A. Telecommunication Networks

The core network of the UK's telecommunication network is a collection of IP devices that are responsible for routing traffic from the exchange up until the network edge. Core nodes are placed in full mesh linking together a number of sub-networks. To limit the amount of routing needed to deliver these packets across the country, devices are grouped into SVLANs designated for carrying a subset of packets labelled with the corresponding SVLAN tag. If devices fail then network protocols govern the re-routing of traffic to avoid outages, placing greater strain on other parts of the network. This can result in increased packet loss, increased latencies and potentially further device failures. Though most outages occur outside the core network, in the 'last mile' between the cabinet and the home, the core network must be resilient to prevent more larger scale issues [13].

A level of service must be maintained for customer satisfaction. Falling below this service level can result in an increased customer churn rate and action from the regulatory body the Office of Communications (Ofcom). Because of their importance there have been a number of approaches focused on increasing the resilience of telecommunication networks. A selection of these are described in Section II-C. These are grouped into those that produce descriptions of the alarms and those that produce predictions.

B. Alarm Description in Telecommunication Networks

For engineers to understand and act on a forecasted alarm it is important to produce these forecasts in a descriptive way. Rule Induction is an inherently expressive modelling approach where models take the form of trees or rule sets. Rule sets require no additional interpreting or transformation to make them human readable. The trained model is a set of human readable rules whose Left Hand Side (LHS) are matched in turn against an instance to assign a label. The following is an overview of a number of Rule Induction applications with specific focus on telecommunications.

The authors of [11] applied ITRULE to generate rules for an expert system in order to automate network management and the networks response to alarms. [22] presents a very general method of detecting a root cause using topology data by finding the most likely failed component between pairs of links. The output is a hypothesised failed link in what is termed a *silent failure* or *black hole*, a similar problem is described in [18] and combated using a modified Bayesian Network.

TASA [14] produces episodic rules from alarm data with the goal to provide new insight into alarm relations, the method of producing these rules is similar to Apriori and Association Rules Mining (ARM). The LHS of these rules are *alarm predicates* defined as any 'expression that can be evaluated from a single occurrence of an alarm', not simply alarm types but any combination of an alarm's features. These predicates have either a strong or weak ordering within their set and are given to the system by the user along with the desired window width, experimentally found to be between 5 seconds and 10 minutes. The method depends on finding frequent patterns with sufficient support before enriching them with additional alarm features.

Two final approaches that focus on describing relations are mentioned here. TP Mining [6] searches for repeated event patterns within a time window and promotes those with a high Topographical Proximity (TP), a metric derived from the relative position of a source device to other devices. As geographical data is provided for a number of devices in the BT data set this method of rule promotion could be beneficial, the suitability of the approach will depend on presence of patterns with sufficient support in the event data. The authors of [19] use Ant Colony Optimisation to produce time based rules, these are rules created by traversing a matrix of all feature values starting with a time to reach a target class, producing and evaluating IF-THEN rules with the nodes selected. Node selection is biased towards the arbitrary ordering of the features within the graph and a high dimensionality would make the method very computationally expensive. Further more there is an assumption that the time of alarm generation is always a defining feature which may not hold.

C. Alarm Prediction in Telecommunication Networks

In this section an number of existing methods of predicting alarms in a telecommunication network are evaluated. Frequent Pattern Mining, which appeared throughout the literature in Section II-B features prominently again. In light of this, this section is split between Apriori based and alternative Frequent Pattern Mining approaches.

1) Apriori Based Approaches: The authors of [16][24] focus their work on alarm prediction in the Pakistan Telecom network. They approach the problem with Decision Trees, an adaptation of Association Rules (termed temporal rules), and Neural Networks. To predict the chosen network events (in this case limited to three types), Apriori is used to identify patterns leading up to the event in a restricted time window. The data is separated according to device type and a large number of alarms deemed non-critical are filtered out before processing. The rules produced are non-descriptive and the restriction to producing rules by device type is a more narrow problem than the one we are presented with.

The authors of [29] produced a genetic algorithm named Timeweaver that specialises in predicting rare events from a telecommunications alarm data set. It follows a two step pattern very similar to Apriori and details a bespoke language enabling patterns to be produced from ordered events, unordered events and wild-cards.

The authors of [20] investigated an enhancement to the algorithm TASA, mentioned above, using sliding windows to find both Association Rules and Episodic Rules.

Episodes are frequently occurring sequences of event types that exist within a window that occur in a time interval. TASA produces human readable rules along with a confidence value. Expert domain knowledge is then required to analyse the great quantity of rules produced by this system before it is applied to a live system. The system has been evaluated by these same domain experts and, amongst the Episodic Rules produced, several unknown patterns have been reported.

2) Alternative Frequent Pattern Approaches: As well as the Apriori methods above there are a number of alternative approaches. The authors of [33][21] use a Markov based codebook approach as an alternative to Association Rule Mining Approaches. It is used to manually identifying problem alarms (alarms of interest) and labelling the succeeding alarms as potential symptoms. This method is concerned with root cause analysis rather than prediction but is included here as it discovers patterns within noisy alarm data. Correlation graphs are formed from the ordering of the symptoms, with some aggregation for repeated sub-sequences, these are then vectorised and presented as a codebook. Hamming distance is used to detect re-occurrences of each code in the network with some resilience to noise. The authors claim a significant speed increase over rule based systems though no empirical evidence of this is presented. eXpose [17] learns dependency rules using the J-measure to help diagnose faults. The J-measure

is a measure of the theoretical information content of an *IF-THEN* rule. exPose uses the information within packet traces as its basis for rule learning, it gathers these within time windows of 1 second, network packets being far busier than the telecommunication data set which consists of many sparse alarm channels. Relations between packets sent through the network are assumed to be unidirectional (as communications are nearly always two way) and so the unweighted J-measure is used . The J-measure is further modified to remove the negative component so as to only score positive relations between events (i.e dismissing the relation between event A happening and event B **not** happening). The rule set is also pruned to keep rules where the following conditions hold:

- The rule has sufficient support
- The LHS and Right Hand Side (RHS) occur with similar frequency

As this work focused on packet traces the inter-arrival times are in the order of nanoseconds whereas the BT data set on the scale of minutes, nevertheless this may be an effective approach.

In [7] a kernel based approach is used. A Support Vector Machine combined with Singular Value Decomposition (SVD) [10] is used to predict alarms over an IP network . Each window of events is converted into a representative discretised vector and the collection of windows form an event-by-window matrix where each column represents a window. SVD is applied and the first k columns of the resulting υ matrix become a new data set to train the SVM with a Radial Basis Kernel (RBF). An online version is proposed using an incremental version of SVD. The optimum window size for the data is investigated through mapping error rates to window sizes from 5 to 100 minutes. The error rate falls and plateaus at the 35 minute mark. This work does produce accurate predictions for a specific alarm type, additional SVD-SVMs must be included for other target classes. There is no rationale behind the rules produced but a further rule extraction phase could produce these.

From the above it can be seen that Rule Induction and Frequent Pattern Mining are common approaches to rule description and prediction. The goal of this work is to produce descriptive predictions of alarms and so Frequent Pattern Mining must be explored. In the next section the BT data set undergoes some pre-processing before the application of 3 established Frequent Pattern Mining algorithms are applied to the data. The goal is to establish if these methods can extract recurring patterns of alarms that can be utilised for prediction.

III. Analysis of Forecasting events with Sequence Mining

The application of a large number of predictive techniques detailed above incorporate or extend Frequent Pattern Mining approaches. Three examples of established Frequent Pattern Mining algorithms were applied to the BT data set. Frequent Patterns are a starting point for discovering sequences with strict ordering constraints. For these experiments each alarm, which follows a hierarchical naming convention, is described by it's over arching alarm type. A successful experiment will result in a number of frequent patterns of varying lengths that can be used to predict alarms on a test set.

To reduce the size of the data set, alarms are first compartmentalised into clusters using the density based clustering algorithm DBSCAN [8] to determine the number and location of centroids. This produces 5 clusters based around population centres that capture both the dense urban and sparse rural network devices at the centroids and edges respectively. It also provides a level of homogeneity across the devices that may be beneficial for rule extraction.

The data has also undergone some preprocessing to remove low variance features and merging repeated events as per [32]. Two clusters were chosen for the experiments, one for testing and one for validation. These clusters were selected as they were of similar size and had populations nearest to the mean cluster size. This is useful as a system produced and verified using these two clusters will be more relevant to the larger and smaller clusters.

Three Frequent Pattern Mining algorithms were applied to the BT data set: Apriori [1], ECLAT [34], and FP-growth [12]. Apriori is one the earliest algorithms designed for detecting small, high confidence item sets and using the Apriori principle to manage the search space. FP-growth uses a depth first approach to generate longer patterns with a lower confidence. These longer patterns are more suited to this task as the transaction sets will take the form of long sequences of events. ECLAT uses a depth first approach with a vertical item set format that can offer some performance gains. Each algorithm retains item sets with sufficient support, a support threshold can be adjusted to increase or decrease the number of patterns produced.

The event data first needs to be compartmentalised into transactions as seen in [17]. To do this the event series was divided using a threshold applied to the inter-event arrival times. This threshold was taken from a sample of SVLANs using their inter-event arrival times. The thresholds were set at each 10^{th} percentile from a sample of SVLANs to give a broad coverage.

The distributions of the transactions under each threshold were examined in order to find the optimal value. The number of items in a set decreases with the value of the threshold. Table I contains aggregate statistics of these transaction sets as the cut off inter-event arrival time between events increases.

The maximum length of the transaction sets increases exponentially as the boundary increases whilst the number of distinct event types increase in a more linear fashion, the mean length being between 1 and 2 across each boundary condition. Table II shows length of the item set produced using Apriori. It demonstrates that the transaction sets produced are predominantly singularities with very high support. Alarms with very high support are likely to be noise events. The support threshold is set conservatively to 10, given the size of the data set this would ideally be larger but the lack of item sets produced makes this impractical.

TABLE I
THE AVERAGE LENGTHS OF THE TRANSACTION SETS BASED ON INTER EVENT ARRIVAL TIMES (TIME BOUNDARIES)

percentile	boundary	avg length	max length	max set length
10	60.0	1.049	44	4
20	240.0	1.085	104	8
30	327.0	1.137	113	8
40	614.0	1.164	121	12
50	1271.0	1.210	121	15
60	3009.0	1.287	171	20
70	5126.0	1.389	188	25
80	10822.0	1.564	299	26
90	21726.5	1.797	483	30

TABLE II
MEAN LENGTH AND SUPPORT OF FREQUENT PATTERNS GENERATED THROUGH APRIORI THRESHOLD (MEASURED IN SECONDS) FOR APRIORI

	avg confidence	avg length	avg support	boundaries
1	0.0	1.0	28293.00	181.0
2	0.0	1.0	28095.50	298.0
3	0.0	1.0	27637.25	689.0
4	0.0	1.0	25507.00	3377.0

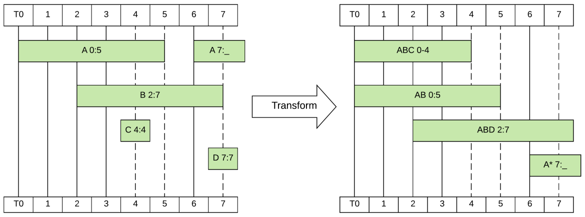

Fig. 1. Transformation of bursty event data to burst format. Multiple smaller events are absorbed into longer complex events with little information loss

Table III shows the output of the FP-Growth and Eclat algorithms when trained over the transaction sets. As the results are identical they are listed only once. Only one pair of transactions occurred with any frequency and each boundary condition produced rules with insufficient confidence to serve as predictors.

A possible reason for the failure to detect frequent patterns is the distribution of alarms. The alarm data is sparse and unevenly spread, a behaviour describable as *bursty*. The probability of a network alarm in a window is relatively small but rises sharply given a network alarm in the preceding time window. The authors of [23] proposed an interval transformation for bursty data. The transformation reduces the number of repeated events by replacing each burst with a set of the events along with the burst's start and end time. These events can then be treated new transaction sets or, by using the start and end times, as complex events. These new complex events can be merged into new transaction sets using the same approach used to create transaction sets detailed above.

TABLE III
MEAN LENGTH AND SUPPORT OF FREQUENT PATTERNS GENERATED THROUGH FP-GROWTH AND ECLAT THRESHOLD (MEASURED IN SECONDS)

	avg confidence	avg length	avg support	boundaries
1	0.0	1.0	28293.00	181.0
2	0.0	1.0	28095.50	298.0
3	0.0	1.0	27637.25	689.0
4	0.0	1.0	25507.00	3377.0

Figure 1 depicts a set of alarms over a short window before and after the transformation. Each alarm type is labelled (A,B,C,D) along with it's starting time-stamp and end time-stamp, for example alarm A starts at 0 and ends at 5. Units of time in this example are seconds but the granularity of the implementation depends on the data. Complex events are created from all events that are live when an event terminates. Events C and D both have a duration of 0 and so their creation time is the same as their termination time. The following is a description of the process followed in this diagram to transform the bursty events into complex events. From left to right across the time window:

1) As event *A* overlaps with events *B* and *C* these are merged into two different complex events at the point of *C*'s termination. This creates event *ABC* which runs from *A*'s starting point at 0 to 4.
2) Similarly, the event *AB* is created from the termination of event *A* at time 5.
3) The termination of *D* and *B*, both at time 7, results in event *ABD*.
4) Event A is still open at the end of the example, it may go on to create further complex events. The start time will be 7 but the end time cannot be determined. The symbols '*' and '_' represent these two values.

The algorithm from [23] is laid out in Algorithm 1, the effect on the alarm data from the BT data set is displayed in a plot of events sampled from three devices on the same SVLAN in Figure 2. Table IV contains the results of the same sample data using the burst transformation.

As the transformation is dependent on overlapping events there is no need to set a boundary threshold to compartmentalise the data. The maximum length of a rule is still 2 but the proportion of length 2 rules is higher, there are also a great number of patterns with sufficient support to produce rules. The lengths of these rules are still too short to be useful in serving as a filter to resolve pre-event clashes.

Algorithm 1 The algorithm transcribed from [23] to convert bursty sequences into a complex form

```
transaction = $\emptyset$
result = $\emptyset$
adding_phase = TRUE
**for** opening $To$ and closing times $Tc$ of all events in sequence $S$ **do**
    items = $S(To)$
    **if** items $\neq \emptyset$ **then**
        trans = trans $\cup$ items
        adding_phase = TRUE
    **end if**
    items = $S(Tc)$
    **if** items $\neq \emptyset$ **then**
        **if** adding_phase = TRUE **then**
            result = result $\cup$ trans
            adding_phase = FALSE
        **end if**
        trans = trans - items
    **end if**
**end for**
```

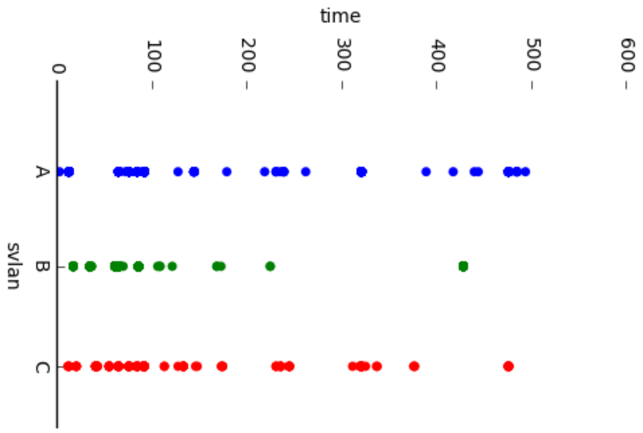

Fig. 2. Examples of bursty, congested and sparse event streams by devices in a share SVLAN, each series is displayed with filtering and without

TABLE IV
MEAN LENGTH AND SUPPORT OF FREQUENT PATTERNS BY FREQUENT PATTERN ALGORITHMS ON THE BURST TRANSFORMED DATA

	algorithm	avg_confidence	avg_length	avg_support
1	Apriori	0.308694	1.533333	31472.933333
2	FP-Growth	0.308694	1.533333	31472.933333
3	Eclat	0.308694	1.533333	31472.933333

Using the same process as before, the data set is split into transactions based on the inter-event arrival times, this time using the complex burst representation. From Tables V and VI it can be seen that the new transaction sets share the same issues as the original. Singularities and repeated events make up the bulk of the transaction sets and there are no sets with a confidence or support above the set threshold. An improvement to the results can be made if a minimum transaction set of 2 is imposed on the data set as seen in Tables VII and VIII but there is no improvement in average confidence.

TABLE V
THE AVERAGE LENGTHS OF THE BURSTY EVENT BASED TRANSACTION SETS BASED ON INTER EVENT ARRIVAL TIMES (TIME BOUNDARIES)

avg length	avg set length	boundary	max set	max set length
1.049146	1.001207	60.0	4	44
1.085810	1.004728	240.0	8	104
1.137938	1.006343	327.0	8	113
1.164693	1.009908	614.0	12	121
1.210368	1.016034	1271.0	15	121
1.287583	1.028528	3009.0	20	171
1.389394	1.047441	5126.0	25	188
1.564482	1.078456	10822.0	26	299
1.797264	1.116214	21726.5	30	483

TABLE VI
CONFIDENCE AND SUPPORT OF FREQUENT SETS PRODUCED BY FP-GROWTH USING THE BURSTY EVENT BASED TRANSACTION SETS

avg confidence	avg length	avg support	boundaries
0.0	0.0	0.0	60.0
0.0	0.0	0.0	240.0
0.0	0.0	0.0	327.0
0.0	0.0	0.0	614.0
0.0	0.0	0.0	1271.0
0.0	0.0	0.0	3009.0
0.0	0.0	0.0	5126.0
0.0	0.0	0.0	10822.0
0.0	0.0	0.0	21726.5

TABLE VII
CONFIDENCE AND SUPPORT OF FREQUENT ITEM SETS PRODUCED BY FP-GROWTH USING THE BURSTY EVENT BASED TRANSACTION SETS WHERE TRANSACTIONS OF CARDINALITY ONE OR LESS ARE REMOVED

avg confidence	avg length	avg support	boundaries
0.0	0.0	0.0	60.0
0.0	0.0	0.0	240.0
0.0	1.0	716.0	327.0
0.0	1.0	735.0	614.0
0.0	0.0	0.0	1271.0
0.0	0.0	0.0	3009.0
0.0	1.0	962.0	5126.0
0.0	1.0	1360.0	10822.0
0.0	1.0	1061.5	21726.5

TABLE VIII
CONFIDENCE AND SUPPORT OF THE SETS PRODUCED BY FP-GROWTH USING THE BURSTY EVENT BASED TRANSACTION SETS WHERE TRANSACTIONS OF CARDINALITY ONE OR LESS ARE REMOVED USING APRIOI, FP-GROWTH AND ECLAT

Algorithm	Avg confidence	Avg length	Avg support
Apriori	0.0	1.0	735.0
FP-Growth	0.0	1.0	735.0
Eclat	0.0	1.0	735.0

This section has demonstrated that the application of Frequent Pattern Mining on the BT data set has been unsuccessful.

This was potentially due to the bursty nature of the alarms creating an environment non-conducive to Frequent Pattern Mining. To investigate this a transformation designed for bursty network data was applied to the alarm data with only a small increase in the number of transaction items but with a very small cardinality. In the next section a new approach to extracting patterns from this data is described.

IV. Pre-Event Marking and Prediction

In Section II-C a number of approaches to alarm prediction were examined. It was apparent that they depend on the extraction of Frequent Patterns using a key feature or item type. Frequent Pattern Mining was applied to the focus data set of this paper and was unable to extract use-able patterns for alarm prediction. In this section an alternative approach hybridising predictive and descriptive approaches is examined. The approach has number of benefits:

- It allows a variety of expressive approaches to make predictions
- It allows forecasting using a wide range of features rather than the single values used in Frequent Pattern Mining
- It can address class imbalance problems if they are present in the data A disadvantage of the approach is the adoption of a large amount of noise into the target class but some work has gone to partially address this.

A. *Pre-Event Marking*

Figure 3 depicts Rule Induction across two axes, forecasting on the vertical axis and classification on the horizontal. Each are limited entirely to their own axis and so a forecaster cannot utilise additional features in it's predictions, this can be a disadvantage if the concept is not entirely contained within one feature. Pre-event Marking is a technique designed to allow an algorithm on the horizontal axis to be trained on a target class that represents the vertical.

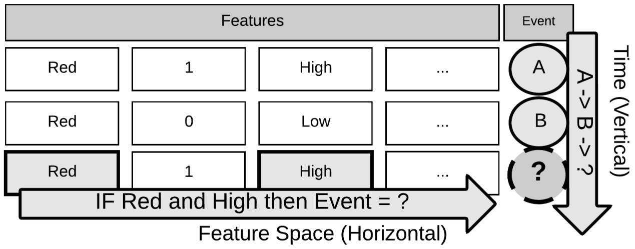

Fig. 3. The axis used for prediction by forecasting and descriptive approaches

This is a very simple transformation of the target class from a descriptor of the instance to an indicator of a future event, in this case a particular alarm type.

A time window of length w is set, terminating at an event of interest, A, and starting at time $A - w$. Events within this time window are marked as pre-events, retaining all their features. This transformation allows a Rule Induction algorithm to produce rules that forecast an event rather than simply describing the containing event. Some methods are required to refine the transformation. There are issues with heavily altering the class balance of the data set and the optimal window with which to produce these rules must be determined. In this section experiments will be conducted over a range of windows and finally an attempt will be made to forecast the expected time of an event.

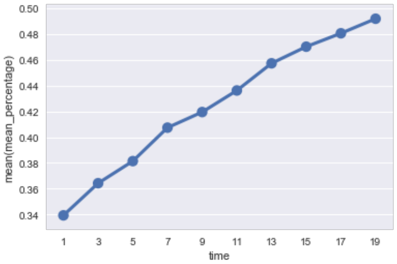

Fig. 4. Proportion of events marked as pre-events against interval size (filtered)

Figure 4 depicts the proportion of class labels as the pre-event window increases from 1 to 19 minutes in the data set. As the time window increases to 19 minutes the proportion of target events approaches 50%. A benefit of this is that algorithms that suffer from class imbalance may perform better, however, this transformation is very likely to introduce a substantial amount of noise.

B. *Rule Induction of Pre-Events*

Pre-event marking is more respectful of the original distribution with smaller window sizes, though the window must be kept large in order for any mitigating action to be taken to prevent the alarm. To attempt to marginalise the amount of noise included in the process at filtering stage is included. One of these techniques is to use an SVM [27] as a filtering black box model to transform the class labels ahead of Rule Induction, the authors of SVM_DT [3] specifically use a decision tree as the second stage.

An SVM was not considered in the early stages of this work as it is a black box system and as such violates the key requirement of producing human readable rules. It also requires all the features to be numerical, as the BT data set's predominant feature type is categorical this will require transforming the data set. Approaches exist to extract human readable rules from SVMs [15], [2].

Transforming categorical attributes to numerical attributes is most simply done using hot-point encoding, or creating dummy vectors. This process converts each distinct feature value into a new boolean feature indicating it's presence or not for that observation. This technique has a number of disadvantages, foremost is the large increase in dimensionality which increases training time and can impact the performance of a model. It is likely that a large number of the additional features created through this process will add no value to the data, an additional phase of feature selection could be employed to lower the training time of the SVM.

Figure 5 demonstrates the work-flow of this Two Stage system. The SVM is trained on the data containing clashes. A second set of training data is then passed through the SVM where pre-events are relabelled based on the SVM's classification. These are then passed to the rule induction algorithm for training. As a final stage a whole data set is passed through the system so the performance of the induced rules can be evaluated.

The effect of SVM filtering on the population of pre-events is shown in Figure 6. The number of events decreases as expected by as much as a 31%. The number of distinct events drops quite heavily, this loss may be a limiting factor in the approach as any concepts linked to the lost event types will not be captured.

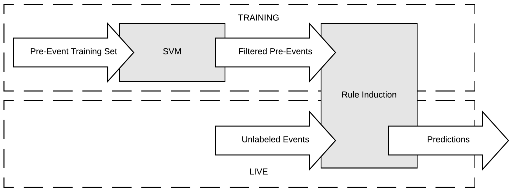

Fig. 5. SVM resolves clashes training the rule induction model. The model is then able to predict pre-events with a greater precision.

In a live system the set up is substantially simplified as after training both algorithms the SVM is no longer required and the rule induction algorithm can classify events without filtering, though the SVM would be required again to retrain the algorithm if, through changes to the network or consumer behaviour, the learnt concepts become invalid.

A number of rule induction algorithms were run on the transformed BT data to predict the pre-event class. These include Prism [4]; PrismTCS [26], an variant of PRISM designed to produce rules for a minority class; and a Decision Tree [25]. As ITRULE and the J-Measure make frequent appearances in Section II a variety of algorithms from that family are trialled. These are ITRULE; ITRULE PRD [31], a variant of ITRULE actively combats Partial Rule Dominance (PRD), a form of over-fitting; a variant using simulated annealing to handle over fitting; and a variant based on eXpose [17] from the literature, referred to as ITRULE Positive.

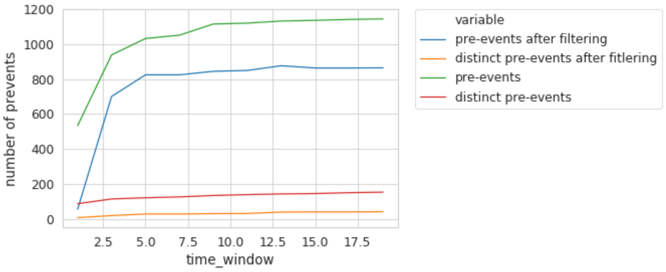

Fig. 6. The effect of the SVM filter on the pre-event population across all time windows.

Figure 7 shows the precisions of all the rule induction algorithms across the time windows as the output of the Two Stage system. The decision tree outperforms the other methods of rule induction. As for the ITRULE variants: ITRULE PRD has over-fit on the majority class while ITRULE Annealing and ITRULE Positive both report precisions lower than before filtering was applied. It is likely that the concepts learnt before filtering have been filtered out by the SVM. The Two Stage SVM filtering with a Decision Tree is by far most effective of those explored. In the following section the SVM filtering algorithm will be explored further.

To optimise the system the same experiments were run with different kernels. Figures 7 and 8 depict the precision and recall for each kernel across different time windows. A summary of the performance of each is provided in Table IX. The kernel choice has little difference on the precision with the exception of the sigmoid kernel under which the Decision Tree algorithm does not perform well. The kernels do, however, have a large effect on the recall of the system though this is secondary to the evaluation under precision. There is a large variation in results for the polynomial and RBF kernel. This suggests that the rules responsible for the high recall are in close competition with other rules.

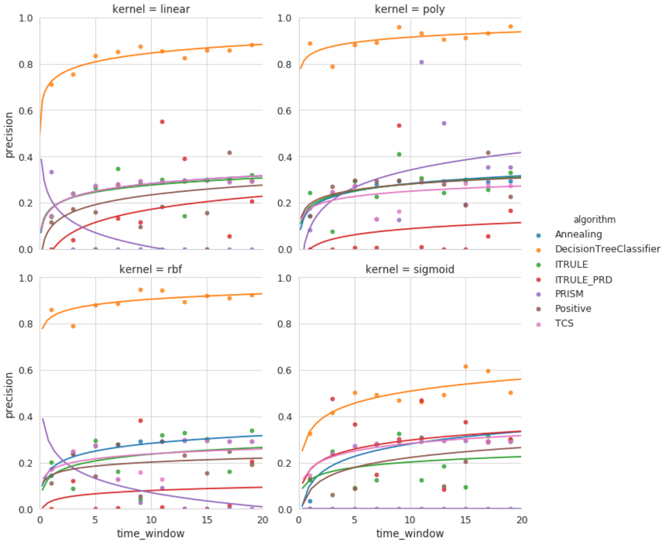

Fig. 7. Mean Precision of the Two Stage classifier under different kernels

The results suggest that the Decision Tree offers the highest precision results when testing as a prediction problem. In the next section the evaluation will be extended to examine predictive performance.

V. Evaluation

In this section the system developed above is tested over an event stream replicating it's use in a live system. The trained model is passed one instance at a time to classify and with each classification an accompanying ground truth is produced.

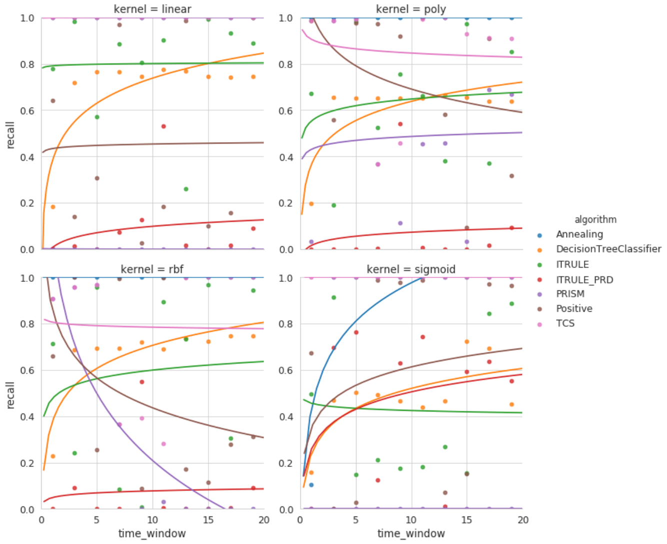

Fig. 8. Mean Recall of the Two Stage classifier under different kernels

TABLE IX
MEAN RESULTS OVER ALL TIME WINDOWS FOR A TWO STAGE CLASSIFIER VARYING KERNELS AND RULE INDUCTION MODELS

Algorithm	Kernel	Accuracy	Precision	Recall
ITRULE Annealing	linear	0.269803	0.269785	1.000000
ITRULE Annealing	poly	0.269777	0.269777	1.000000
ITRULE Annealing	rbf	0.269777	0.269777	1.000000
ITRULE Annealing	sigmoid	0.300986	0.258905	0.910406
Decision Tree	linear	0.888923	0.831425	0.695305
Decision Tree	poly	0.881917	0.905826	0.604988
Decision Tree	rbf	0.895751	0.895853	0.666622
Decision Tree	sigmoid	0.733485	0.487562	0.486057
ITRULE	linear	0.359560	0.264398	0.799775
ITRULE	poly	0.444512	0.269358	0.636508
ITRULE	rbf	0.443955	0.224744	0.584396
ITRULE	sigmoid	0.402580	0.193069	0.428411
ITRULE_PRD	linear	0.664264	0.149453	0.087450
ITRULE_PRD	poly	0.643854	0.077689	0.066094
ITRULE_PRD	rbf	0.658599	0.073700	0.074288
ITRULE_PRD	sigmoid	0.606272	0.281463	0.475431
Prism	linear	0.730197	0.033333	0.000176
Prism	poly	0.529995	0.311696	0.479527
Prism	rbf	0.522585	0.094302	0.323219
Prism	sigmoid	0.730223	0.000000	0.000000
ITRULE Positive	linear	0.457562	0.217441	0.451218
ITRULE Positive	poly	0.416464	0.270959	0.730776
ITRULE Positive	rbf	0.377795	0.195153	0.487249
ITRULE Positive	sigmoid	0.436899	0.204826	0.581314
Prism TCS	linear	0.270182	0.269818	0.999824
Prism TCS	poly	0.258447	0.240875	0.853467
Prism TCS	rbf	0.271624	0.229169	0.786998
Prism TCS	sigmoid	0.269777	0.269777	1.000000

This ground-truth represents the presence of a critical alarm in the subsequent time window on the same SVLAN. The output is a set of predictions and a set of ground-truths with which to verify the predictions.

These tests were conducted with data from a medium sized cluster and further verified with a model trained and tested on a second cluster of similar size, as per Section III. The results are displayed in Figure 9, generally the precision, recall and accuracy are high. The recall for predicting events under 5 minutes is low but in contrast the precision for these events is very high indicating that only a few predictions were made but these predictions were accurate.

The drop in recall for the 5 minutes window may indicate that the concepts required to predict these alarms were not available so close to the alarms generation.

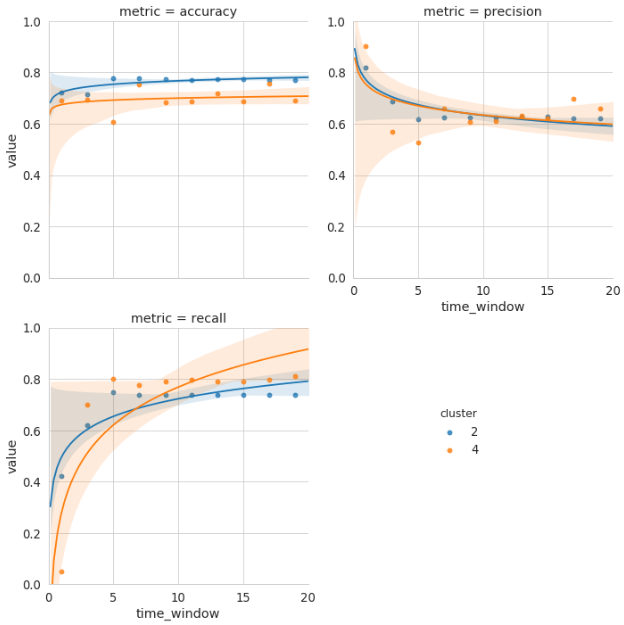

Fig. 9. Precision, recall and accuracy of the rules produced by the Two Stage system

Training a model on the pre-event data ensures that the model is kept general across SVLANs as the events seen are not SVLAN specific. As SVLANs are likely to contain attributes peculiar to themselves this is a strength of the system. An alternative would be to produce a model for each SVLAN which would be computationally inefficient and could to lead to over fitting. A problem with this method of evaluation is that not all SVLANs will produce target alarms. Under such a circumstance any positive prediction will immediately create a precision of 0.0. A potential improvement to the system would be to monitor the number of times the rules fire. This could be examined and a confidence level for the alarm prediction produced.

Table X contains the results from a further set of tests that include the number of alarms covered by each time a rule has fired. Of note here is the high number of True Negatives (TN) and the equivalent True Negative Rate (TNR). As described above, the event stream is bursty with down events appearing in clusters rather than appearing in isolation and so the detection of one critical alarm is highly likely to encompass several more alarms of the same type. It is important to establish that the system does not fire randomly and skew the results by covering one or more of these clusters, the high TNR rates indicates that this not the case.

TABLE X
MEAN VOLUME OF ALARMS CORRECTLY AND INCORRECTLY PREDICTED ALONG WITH TRUE POSITIVE AND FALSE POSITIVE RATES FOR THREE FOLDS ACROSS CLUSTER 1

window length	fold	TP	FP	TN	TNR	TPR	FPR
5	1	552.28	181.34	564.56	66.74	56.26	33.26
5	2	424.12	185.80	520.46	57.32	59.66	42.68
5	3	523.94	188.86	553.98	62.22	60.05	37.78
11	1	574.50	188.04	557.86	66.68	57.32	33.32
11	2	434.82	193.06	513.20	58.58	58.81	41.42
11	3	540.64	197.50	545.34	62.12	60.56	37.88
15	1	574.50	188.04	557.86	66.68	57.32	33.32
15	2	434.82	193.06	513.20	58.58	58.81	41.42
15	3	540.64	197.50	545.34	62.12	60.56	37.88

VI. Conclusion

This paper has put forward a method of alarm prediction that can successfully forecast down events in the BT telecommunication network in an expressive way using a novel transformation proposed here and a SVM and Decision Tree combination [3]. The method was developed in response to the poor performance of traditional item set mining approaches on this data set which has been attributed to both the bursty nature of the data and the underlying concepts existing outside of the class label. The transformation is able to discover these patterns as it is not constrained to just one *item label* and is able to make forecasts based on all the features including previous class labels. Several methods of item set mining were trialled along with a method designed for such data and these were unable to produce usable patterns from the data.

Also trailed were a number of Rule Induction classifiers, chosen as they are able to produce classifications in human readable rules without any intermediate stages. They are also able to abstain from classification which is a useful property when not all values of the target class are of interest. These were outperformed by a Decision Tree, still a white box model that textual rules can easily be extracted from. The method proposed adapts a lateral classifier into a predictor using pre-event marking. It produces high precisions and a high TNR rate which are both key in an alarm prediction system to keep false alarms to a minimum and retain user trust. It has been evaluated in both a classification scenario to establish the presence of predictable patterns in relation to the class label and finally in a streaming scenario to establish its performance as a forecaster.

The system could be further tuned using confidence levels to decrease the false positive rate, it is also lacking an estimated time for the event to arrive. As it stands a window size must be selected for training and the event can occur anywhere within that time. Whilst the combination of window sizes can be used to produce a more precise prediction time this is not explicitly included within the rules.

References

[1] Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. ACM SIGMOD Record **22**(2), 207–216 (1993). DOI 10.1145/170036.170072

[2] Barakat, N., Diederich, J.: Rule Extraction from Support Vector Machines. Computational Intelligence **80**(1), 59–62 (2008). DOI 10.1007/978-3-540-75390-2. URL http://www.springerlink.com/index/10.1007/978-3-540-75390-2

[3] Barakat, N.H., Bradley, A.P.: Rule extraction from support vector machines: A sequential covering approach. IEEE Transactions on Knowledge and Data Engineering **19**(6), 729–741 (2007). DOI 10.1109/TKDE.2007.190610

[4] Cendrowska, J.: PRISM: An algorithm for inducing modular rules. International Journal of Man-Machine Studies **27**, 349–370 (1987). DOI 10.1016/S0020-7373(87)80003-2

[5] of Communications), U.O.: Communications Act 2003 (2003). URL http://www.legislation.gov.uk/ukpga/2003/21/contents

[6] Devitt, A., Duffin, J., Moloney, R.: Topographical proximity for mining network alarm data. Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data - MineNet '05 p. 179 (2005). DOI 10.1145/1080173.1080179. URL http://portal.acm.org/citation.cfm?doid=1080173.1080179

[7] Domeniconi, C., Perng, C., Vilalta, R., Ma, S.: A classification approach for prediction of target events in temporal sequences. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 125–137. Springer (2002)

[8] Ester, M., Kriegel, H.P., Sander, J., Xiaowei, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD **96**(34), 226–231 (1996)

[9] Fornell, C., Johnson, M.D., Anderson, E.W., Bryant, B.E.: American Customer Satisfaction Index. Choice Reviews Online **46**(03), 46–1254–46–1254 (2013). DOI 10.5860/choice.46-1254

[10] Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. Numerische Mathematik **14**(5), 403–420 (1970). DOI 10.1007/BF02163027. URL https://doi.org/10.1007/BF02163027

[11] Goodman, R.M., Ambrose, B., Latin, H., Ulmer, C.T.: Noaa: An Expert System Managing The Telephone Network. Springer (1995)

[12] Han, J., Pei, J., Yin, Y.: Han-frequent-patterns-without-candidate-generation.pdf. ACM SIGMOD Record **29**(2), 1–12 (2000). DOI 10.1109/FSKD.2007.402

[13] Harriss, L.: Telecommunications Infrastructure: Cabling, Ducts and Poles. POST Notes: Houses of Parliament Office of Science and Technology (24) (2017)

[14] Hätönen, K., Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H.: Rule Discovery in Alarm Databases. Tech. Rep. C-1996-7, University of Helsinki, Helsinki (1996). URL https://www.cs.helsinki.fi/TR/C-1996/7/

[15] Huysmans, J., Baesens, B., Vanthienen, J.: Using Rule Extraction to Improve the Comprehensibility of Predictive Models. SSRN Electronic Journal pp. 1–55 (2006). DOI 10.2139/ssrn.961358. URL https://lirias.kuleuven.be/bitstream/123456789/121060/1/KBI_0612.pdf http://www.ssrn.com/abstract=961358

[16] Jaudet, M., Hussain, A., Sharif, K.: Temporal Classification for Fault-prediction in a real-world Telecommunications Network pp. 209–214 (2005)

[17] Kandula, S., Chandra, R., Katabi, D.: What's going on?: learning communication rules in edge networks. In: Proceedings of the ACM SIGCOMM 2008 conference on Data communication, pp. 87–98. ACM (2008). DOI 10.1145/1402958.1402970

[18] Kandula, S., Katabi, D., Vasseur, J.p.: Shrink: A tool for failure diagnosis in IP networks. Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data pp. 173–178 (2005). DOI 10.1145/1080173.1080178. URL http://portal.acm.org/citation.cfm?id=1080178

[19] Khan, I., Huang, J.Z., Tung, N.T.: Learning Time-based Rules for Prediction of Alarms from Telecom Alarm Data Using Ant Colony Optimization. International Journal of Computer and Information Technology (ISSN: 2279 0764) **03**(01), 139–147 (2014)

[20] Klemettinen:, M.: A Knowledge Discovery Methodology for Telecommunication Network Alarm Databases. October (1999)

[21] Kliger, S., Yemini, S., Yemini, Y.: A coding approach to event correlation. ... Network Management IV pp. 1–12 (1995). DOI doi:10.1007/978-0-387-34890-2_24. URL http://link.springer.com/chapter/10.1007/978-0-387-34890-2_24

[22] Kompella, R.R., Yates, J., Greenberg, A., Snoeren, A.C.: Detection and localization of network black holes. In: Proceedings - IEEE INFOCOM, pp. 2180–2188 (2007). DOI 10.1109/INFCOM.2007.252

[23] Lachmann, A., Riedewald, M., Zhang, Z., Wang, S., Lachmann, A., Riedewald, M.: Finding relevant patterns in

bursty sequences. Proceedings of the VLDB Endowment **1**(1), 78–89 (2008). DOI 10.1145/1453856.1453870. URL http://portal.acm.org/citation.cfm?id=1453856.1453870

[24] Mohammad Jaudet, N.I.: Neural networks for fault-prediction in a telecommunications network. 8th International Multitopic Conference, 2004. Proceedings of INMIC 2004. pp. 315–320 (2004). DOI 10.1109/INMIC.2004.1492896. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1492896

[25] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)

[26] Stahl, F., Bramer, M.: Random prism: A noise-tolerant alternative to random forests. Expert Systems **31**(5), 411–420 (2014). DOI 10.1111/exsy.12032

[27] Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer New York, New York, NY (2000). DOI 10.1007/978-1-4757-3264-1. URL http://link.springer.com/10.1007/978-1-4757-3264-1

[28] Weiss, G.M.: Data Mining in the Telecommunications Industry. Data Mining and Knowledge Discovery Handbook pp. 1189–1201 (2005). URL http://dx.doi.org/10.1007/0-387-25465-X_56

[29] Weiss, S.M., Indurkhya, N.: Predictive data mining: a practical guide. Morgan Kaufmann (1998)

[30] Wrench, C.: personal communication

[31] Wrench, C., Stahl, F., Fatta, G.D., Karthikeyan, V., Nauck, D.: Towards expressive rule induction on ip network event streams. In: AI-2015 Thirty-fifth SGAI International Conference on Artificial Intelligence (2015). URL http://centaur.reading.ac.uk/56395/

[32] Wrench, C., Stahl, F., Le, T., Di Fatta, G., Karthikeyan, V., Nauck, D.: A Method of Rule Induction for Predicting and Describing Future Alarms in a Telecommunication Network. In: Research and Development in Intelligent Systems XXXIII, pp. 309–323. Springer International Publishing, Cham (2016). DOI 10.1007/978-3-319-47175-4_23. URL http://link.springer.com/10.1007/978-3-319-47175-4_23

[33] Yemini, S.A., Kliger, S., Mozes, E., Yemini, Y., Ohsie, D.: High Speed and Robust Event Correlation. IEEE Communications Magazine **34**(5), 82–90 (1996)

[34] Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. KDD **97**, 283–286 (1997)