A method of rule induction for predicting and describing future alarms in a telecommunication network

Published version at: http://dx.doi.org/10.1007/978-3-319-47175-4_23

Publisher: Springer International Publishing

A Method of Rule Induction for Predicting and Describing Future Alarms in a Telecommunication Network

Chris Wrench, Frederic Stahl, Thien Le, Giuseppe Di Fatta, Vidhyalakshmi Karthikeyan, Detlef Nauck

Abstract In order to gain insights into events and issues that may cause alarms in parts of IP networks, intelligent methods that capture and express causal relationships are needed. Methods that are predictive and descriptive are rare and those that do predict are often limited to using a single feature from a vast data set. This paper follows the progression of a Rule Induction Algorithm that produces rules with strong causal links that are both descriptive and predict events ahead of time. The algorithm is based on an information theoretic approach to extract rules comprising of a conjunction of network events that are significant prior to network alarms. An empirical evaluation of the algorithm is provided.

1 Introduction

The reliance on Telecommunications in our personal and business lives makes alarm prediction in this domain an extremely important field of research with the potential of saving businesses a great deal of money; personal users a large amount of inconvenience; and, in crisis situations, may save lives [5]. The goal of this work is to produce a Rule Induction Algorithm to produce human readable rules that predict the occurrence of alarms in a telecommunication network ahead of time. These are advantageous as the prediction is accompanied with rationale of why the prediction was made which may give valuable insight to domain experts. The domain experts are then better equipped to correct the issue as well as mitigate it in the future. Fur-

Chris Wrench, Frederic Stahl, Thien Le and Giuseppe Di Fatta
Department of Computer Science, University of Reading, PO Box 225, Whiteknights, Reading, RG6 6AY, UK, e-mail: C.Wrench@pgr.reading.ac.uk, F.T.Stahl@reading.ac.uk, G.DiFatta@reading.ac.uk

Vidhyalakshmi Karthikeyan and Detlef Nauck
BT Research and Innovation, Adastral Park, IP5 3RE, UK e-mail: Vidhyalakshmi.Karthikeyan, Detlef.Nauck@bt.com

thermore, it is desirable to produce these in such a way as to utilise as much context information from the data rich events that make up the data set. The paper focusses on a data set provided by BT of events gathered from a national Telecommunication network over a period of two months. Events are often generated in a non-uniform, bursty manner but contain a large number of attributes describing the cause of AND conditions that led to the target events being generated [20]. There are algorithms that are able to predict the occurrence of an event with some accuracy [7] but they are limited to using a market basket approach which discards the majority of data. The terms *alarm* and *events* are used throughout this paper, the data set that forms the focus of this research consists of alarm data, these are a special form of events as studied in event mining. Whilst an event can be anything that has happened, an alarm further indicates that something has happened that was *unexpected* [14].

The paper is laid out with a summary of existing Rule Induction techniques and other works focussed on Telecommunication data in Section 2, Section 3 outlines the pre-processing steps taken to refine the data set with a description of how the temporal (timestamps) and geolocation attributes were exploited in Subsection 3.2. Section 4 details the progression towards the current state of the Rule Induction Algorithm and Section 5 contains the results. Finally Section 6 describes some ongoing work and Section 7 contains a conclusion.

2 Related Work

2.1 Event Mining in Telecommunication Networks

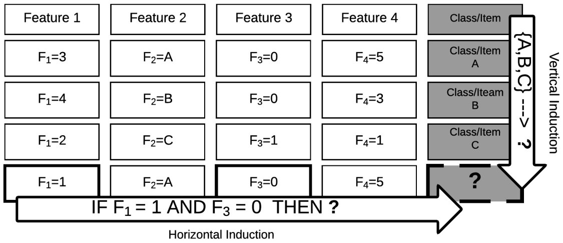
Telecommunications were amongst the first to use Data Mining and Data Stream Mining for applications that, as well as Fault Prediction, include Marketing (detecting likely adopters of new services or value customers) and Fraud Detection (identifying unscrupulous accounts to prevent losses to both the company and victimised customers), as categorised by [25]. The following is a collection of algorithms involved in fault detection in both telecommunication networks and networks in general.

In [13] Ant Colony Optimisation is used to produce time based rules. The approach yields a high accuracy, however, it is limited to predicting the class value of an instant rather than predicting a future alarm. Decision Trees (a variant of C4.5) are used in [12] for alarm detection using log data by presenting event types (i.e. one key attribute of the event) as an item set. The Decision Tree is passed an item set that consists of the previous observations and the current observation. The rules produced are then filtered to retain only those whose terms obey temporal ordering (the conditional is made of those items that appeared before the consequent).

TP Mining from [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. The authors of [24] produced a genetic algorithm named *Timeweaver* that specialises in predicting rare events from a telecommunications alarm data set.

Several algorithms are based on, or incorporate, Association Rule Mining and the Apriori [1] approach to make their predictions. The authors of [19] focus their work around alarm prediction in the Pakistan Telecom (PTLC) network. They approach the problem with an adaptation of Association Rules (along with Decision Trees and Neural Networks). It allows the prediction of alarms based on a sequence, much like the target of this work. The rules are non-descriptive, however, and the restriction to producing rules by device type is a narrower problem than the one we are presented with. The authors of [14] investigate an enhancement to alarm correlation. The algorithm TASA (Telecommunication Alarm Sequence Analyser) uses sliding windows to find both Association Rules and Episodic rules (frequently occurring sequences of event types that occur in a time interval). The rules produced are validated by domain experts.

The above algorithms can collectively predict an instance's target class using its own attributes or predict the value of a future instance by focussing on one target feature from the preceding instances and grouping these into an item set, see Figure 1. The approaches described above are effective for prediction along one axis, either vertically or horizontally, but not both. Utilising both methods could discover alternative rule sets and make more interesting causal relations, which is the focus of the work presented in this paper.



Feature 1	Feature 2	Feature 3	Feature 4	Class/Item
F_1=3	F_2=A	F_3=0	F_4=5	Class/Item A
F_1=4	F_2=B	F_3=0	F_4=3	Class/Iteam B
F_1=2	F_2=C	F_3=1	F_4=1	Class/Item C
F_1=1	F_2=A	F_3=0	F_4=5	?

Fig. 1: Data utilisation in Prediction and Description, existing algorithms predict a value (or event) along just one dimension.

2.2 Rule Induction

The purpose of Rule Induction is to generate a series of human readable rules that classify an instance and provide an explanation behind the classification:

$$\text{IF } A \text{ AND } B \text{ THEN } C_i$$

Meaning that if an instance matches the conditions on the Left Hand Side (LHS) then there is a high probability that this instance belongs to the class on the Right Hand Side (RHS). Rule Induction algorithms generate rules either as bottom-up or top-down[8, 9].

- **Bottom-up** - (starting with a highly specialised rule which matches a single instance, and generalising by successively removing the least valuable attributes
- **Top-down** - the data set is split using the value of one attribute and reduced further with the addition of each rule term (referred to as specialising the rule).

Rules can be generated through the application of pre-existing forms of machine learning: Decision Trees [21, 15], Covering algorithms [3, 4, 18], and the extraction of rules from black box algorithms such as Neural [23] or Bayesian networks [10]. The goal is always to produce a rationale along with a classification that is easily understood by a human reader. The more general a rule is (the less specific) the greater coverage it will have and the rules produced will have a greater resilience to noise. The goal of this work is to produce expressive rules, such as those described above, but with the additional ability to predict, ahead of time, events occurring in the near future.

3 Data Preparation

3.1 Pre-Processing

Feature Selection was conducted manually using input from domain experts. Strings, particularly those relating to location, were cleaned of white spaces and trailing characters. These could then be matched using Levenshtein distance [11] to reduce the number of unique string values in the dataset. Numerical attributes (excepting timestamps) were binned and instances with a high number of missing values were removed. If the majority of an attribute's values were missing it was replaced with a boolean attribute to indicate if the field is populated or not.

One of the key features in the data set is Event Name which contains over 180 distinct string values. To simplify the data set it was decided to reduce these. This was done by dividing the data set into 2000 time windows, making each window 2547s long, and counting the occurrences of each value in every window. This effectively creates 180 time series, one for the occurrence of every event type. These time series are then comparable through Dynamic Time Warping (DTW) [27] producing a distance matrix. The final step is to perform hierarchical clustering using these distances producing a dendrogram that clearly indicates the groups that occur alongside each other. Many of the events in these groups have a high string similarity suggesting they are semantically similar. The process of aggregating event counts across time windows has also demonstrated that there are events that occur regularly across all time windows, indicating that these are largely noise and will need to be addressed.

3.2 Co-Occurrence detection

Included in the data set are several features that relate to the temporal or physical location of the event origin that can be used for Co-Occurrence detection. Co-Occurrence is used to increase the likelihood that two events are causally linked by identifying events that occur within physical and temporal proximity to each other. The utilisation of both kinds of proximity are described in Subsections 3.2.1 and 3.2.2.

3.2.1 Geo-Location Clustering

The national data set was divided into subsets based on the event's accompanying northing and easting values. This was done by first normalising these values, to account for the disproportion in the UK's easting and northing, then breaking the data set into time windows and running each window through a clustering algorithm. Figure 2 is an example of one of these sliding windows plotted over a map of the UK [16]. After trials with other clustering algorithms, DBScan [2] was chosen as the clustering algorithm as it produces a dynamic number of clusters based on density. The clusters produced were also amongst the most logically defined out of the other trials. The resultant clusters were then plotted making it possible to manually identify the approximate centres of activity in the UK. Using this information, 5 centroids were selected and each data point assigned a cluster based on its nearest centroid, measured using Euclidean distance, resulting in 5 data sets. Experiments from here are limited to one of these clusters. Figure 3 depicts the chosen centroids for the overall geo-location clustering of the data along with the relative size of each cluster.

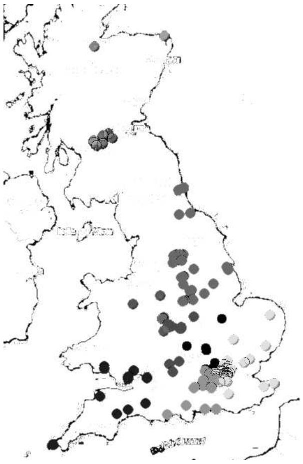


Fig. 2: Clustering of geo-location events map

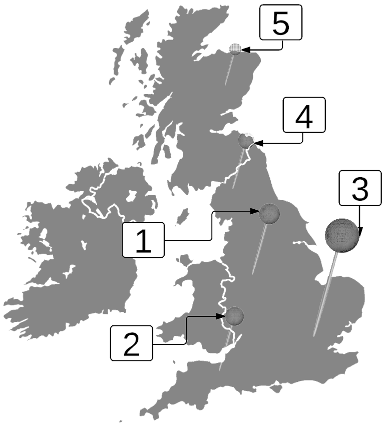
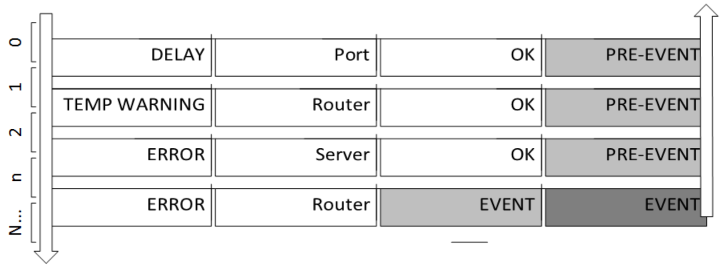


Fig. 3: Cluster centroids and relative sizes

3.2.2 Pre-Event Assignment for Prediction

Section 4 details one part of the contribution of this paper, specifically the modifications made to two algorithms that, as they stand, induce rules along the horizontal axis (predicting a class value within an instance). The goal of this work is to make human readable rules using as much context information as possible that predict an event in the near future. To make these rules behave in this way an additional binary class label was introduced to the dataset, referred to here as a *Pre-Event*. These indicate the presence of an event of interest within a given time window. For the experiments in Section 5 this window is between 10 and 15 minutes before the event occurs, this interval was settled upon through personal correspondence with BT. The data set is parsed in reverse order (starting with the most recent) and events that occur within the window are marked as positive, see Figure 4. A problem with this method is that it skews class bias, this will be addressed in future work.



DELAY	Port	OK	PRE-EVENT
TEMP WARNING	Router	OK	PRE-EVENT
ERROR	Server	OK	PRE-EVENT
ERROR	Router	EVENT	EVENT

Fig. 4: Creating a *Pre-Event* class used for prediction

4 Alarm Prediction and Description

This Section outlines the modifications and optimisation of a rule induction algorithm used on the dataset from two ITRULE based approaches to a covering approach. The first algorithm used is a basic implementation of ITRULE with minor alterations to combat over specialisation, used here as a baseline for comparison. The second algorithm is an adaptation of ITRULE first introduced in a previous work [26] featuring a pruning method to correct partial rule dominance, developed originally for streaming data. Following this, a covering method was developed based on the PRISM method but using J-measure to induce rules and later incorporating pruning based on confidence. Performance increases in some areas are accompanied by each new algorithm.

ITRULE was initially chosen for adaptation because of it's strong statistical foundation, faculty to produce expressive rules, and it's resistance to over-specialisation through the property Jmax, explained later in this section. ITRULE was developed by Goodman and Smyth [22] and produces generalised rules from batch data consisting of many nominal attributes. It evaluates every combination of possible rule terms using the resultant rule's theoretical information content, known as it's J-measure, see Equation 1. This is calculated from the product of the LHS's probability (*p(Y)*) and the cross entropy of the rule (*j(X:Y=y)*) see Equation 2. A very useful property of ITRULE is that the maximum information content of a given rule (it's maximum J-measure) is bounded by the property Jmax, see Equation 3. This enables ITRULE to stop specialising a rule as it nears Jmax as no further gain in J-measure can be obtained.

It uses Beam Search to keep the search space to a manageable size, only selecting the top N rules from each iteration to expand upon. The rule is then specialised by appending further rule terms to the LHS of the rule. When every combination of the next phase of rules have been produced, the top N are again selected and the process continues until Jmax is reached.

$$J(X:Y=y) = p(Y).j(X:Y=y) \tag{1}$$

$$j(X:Y=y) = p(x|y).\log(\frac{p(x|y)}{p(x)}) + 1 - p(x|y).\log(\frac{p(x|y)}{1-p(x)}) \tag{2}$$

$$J_{max} = p(y).\max\{p(x|y).\log(\frac{1}{p(x)}), 1 - p(x|y).\log(\frac{1}{1-p(x)})\} \tag{3}$$

Beam Search is a greedy algorithm and frequently produces rule sets with little variation in the rule terms used, a problem known as partial rule dominance. This is particularly evident in cases where there is a large disparity between the assigned worth, or *goodness*, of attributes. In this case there is a great disparity between the number of distinct values belonging to an attribute, even after pre-processing. As the J-measure is a product of the probability of an instance matching the LHS of a rule, those features with fewer distinct values tend to have a much higher J-measure than those with many values. If not accounted for, these feature values will repeatedly be selected for further specialisation to the extent that they dominate the beam width. This in turn leads to rules which share very similar conditionals and cover a very narrow range of features, often repeating similar tests [9]. For example, in the below, feature values A_1 and B_1 have a high probability of both occurring and being selected in the top N rules. This leads to very similar rules with low coverage:

IF A_1 AND B_1 AND C_1 THEN X
IF A_1 AND B_1 AND C_2 THEN X
IF A_1 AND B_1 AND C_3 THEN X

This rule set has a large amount of redundant information and could instead be presented as a more general rule leaving space in the beam for more interesting rules:

IF A_1 AND B_1 THEN X

This can be addressed either as rules are induced or afterwards (pre-pruning or post-pruning).

To mitigate against partial rule dominance a form of pre-pruning was incorporated. When selecting the candidates for the next iteration of rule specialisation, the rules are ranked according to their respective J-measures and the top N are added into the beam for specialisation. When a rule is added a distribution of the features within the beam is updated. Here a rule is only added if this rule's individual member features do not surpass a threshold proportion of the beam (set to 40% for these experiments). Otherwise it is not included and the next best rule term is tested in the same way. It is possible to exhaust the list of candidate rule terms with this method if the beam size is sufficiently large or the candidate list is sufficiently small. This being the case, the list (minus the rules already selected) is iterated again until a sufficient number of rules have been selected. This approach leads to a more varied rule set though it has the disadvantage of potentially losing a rule with high a J-measure in preference for a new minority rule term with a lower J-measure. The rule set stands a greater chance of producing rules that are of interest to the domain experts.

An alternative to using the algorithms original Beam Search is to adopt the Separate and Conquer method used in covering algorithms such as AQ [18] and PRISM [3]. PRISM was developed as a means to overcome the repeated sub-tree problem inherent with decision trees whereby parts of the tree are repeated leading to the same tests being carried out multiple times. It does this by selecting the rule term so that the conditional probability of covering the target class is maximised before separating out the instances it covers from the rest of the data set. From this new, smaller data set another rule term is selected (the rule is specialised) and the data set is further reduced, those not falling under the longer rule are returned to the original dataset. This continues until all instances in the set have the same class value and the rule is finalised. The process repeats until all instances are covered by a rule.

A new algorithm was developed as a hybrid between PRISM and ITRULE by replacing the beam search approach with a covering technique which removes any partial rule dominance issues.

PRISM uses conditional probability, designed to maximise coverage, to induce rules. Here, in place of this, the J-measure is used, inducing rules based on their information content, and Jmax is retained as a stopping criteria to avoid over-fitting. The resultant algorithm is referred to here as a *Prism-ITRULE Hybrid*. A pruning step based on confidence was later incorporated after rule induction, the algorithm is described in Algorithm 1.

5 Results

This Section details results from the experiments performed with the algorithms outlined in Section 4. The following tests were conducted using three different alarm types as their Pre-Event target class (i.e. for each data set one of three different alarm types was used for Pre-Event generation). These data sets were split into 3 smaller data sets of 30,000 instances based on the time of occurrence and divided randomly into test and training sets. The training and test sets consist of 10,000 and 20,000 instances respectively.

Algorithm 1 Prism-ITRULE Hybrid - where J is calculable using Equations 1 and 2 and J_{max} is calculable using Equation 3

```
Dataset $D$ with Target Classes $C_n$ and Attributes $A_n$
for Every Class $C_i$ in $D$ do
    while $D$ contains classes other than $C_i$ do
        for Every Attribute $A_i$ in $D$ do
            for Every Attribute Value $A_{iv}$ in $A_i$ do
                Generate Rule $R_n$ with Rule Term $A_{iv}$
                Calculate $j$
                $J_{max} \leftarrow J_{max} * ThresholdT$
                add $R_n$ to Candidate Rule Set
            end for
        end for
        Select $R_n$ where $j(R_n)$ is maximised
        Remove Instances not covered by $R_n$
        if $j(R_n) > J_{max}(R_n)$ then
            Rule Complete
            Break
        end if
    end while
    for all Rule Terms $RT_i$ : in Rule $R$ do
        if Confidence( $RT_i$) <Confidence($RT_{i-1}$) then
            Remove $RT_i$ from $R$
        end if
    end for
    $C_i \leftarrow$ Remove Instances Covered by $R_n$ from $D$
end for
```

Table 1 shows the percentage of instances marked as positive (the target attribute value) in each data set. 3 values of the feature Event Name were selected as target classes (from which Pre-Events are made), these will be referred to as events A, B and C. Target class A is of particular importance to BT (determined from personal communication with BT) whilst B and C were selected for their contrasting distributions.

Table 1: Class proportion of target events

Data Set	Class Breakdown	Percentage Target Class
A	A_i	7.41
	A_{ii}	6.92
	A_{iii}	4.24
B	B_i	69.55
	B_{ii}	35.50
	B_{iii}	66.16
C	C_i	0.68
	C_{ii}	0.73
	C_{iii}	0.51

Several tests were carried out using the algorithms described in Section 4 on a sample of the data. Tables 2 and 3 detail some of the recorded metrics from these experiments, including Abstain Rate and Tentative Accuracy. A brief description of these is provided below:

- Accuracy % (Acc) - the percentage of instances that were correctly classified from the whole data set.
- Tentative Accuracy % (Tent-Acc) - the percentage of instances that were correctly classified from those covered by rule set.
- Abstain Rate % (Abs Rate) - the percentage of instances in the data set that were not covered by a rule and no attempt at classifying was made.
- Time (s)- the time taken for training on the data set taken from the mean of three runs.

Tentative Accuracy is included here as this problem does not require total classification of all instances, more important is the correct classification of the rule firing. The data set now uses a binary class (an event is marked as a Pre-Event or not) and rules are produced for both class types, as such the accuracies apply, as well, to both classes.

5.1 ITRULE

The first tests were run on the original version of ITRULE and serve as a baseline for comparison, see Table 2. For these experiments a beam width of 45 was used and the value of Jmax was set to 80% of its full value to combat over fitting. These values were found to yield the highest accuracies in previous experiments. Table 2 also contains the results of experiments using Partial Rule Dominance Pruning. In terms of accuracy and tentative accuracy the results are very similar, the accuracy for predicting A events are higher due to a much reduced abstain rate. The trade-off in the variance in execution time is higher in all cases, this is due to a large amount of additional tests needed to populate the beam size in such a heavily skewed data set. Even with the inclusion of Partial Rule Dominance Pruning the overall accuracy is low and the abstain rate is high.

5.2 Prism-ITRULE Hybrid

In Table 3 are the results of experiments done with two implementations of a Prism-ITRULE Hybrid, one with and one without confidence pruning. Again with a Jmax threshold set to 80% of its full value. There are many evident benefits over beam search as the abstain rate is far lower, boosting the accuracy in turn. The tentative accuracy has increased too even though it is independent of the latter two metrics.

Table 2: Results for ITRULE and ITRULE with Partial Rule Dominance Pruning

	Basic ITRULE				Partial Rule Dominance Pruning			
Data Set	Acc%	Tent-Acc%	Abs Rate %	Time(s)	Acc %	Tent-Acc%	Abs Rate%	Time(s)
A_i	23.03	93.75	75.44	68.41	36.63	92.87	60.57	41.83
A_{ii}	35.92	92.94	61.35	31.35	49.07	95.93	48.85	20.18
A_{iii}	33.73	93.47	63.92	41.89	30.51	93.49	67.37	40.54
B_i	99.16	99.31	0.15	7.91	76.83	99.14	22.51	7.75
B_{ii}	99.49	99.49	0.00	8.38	92.56	99.47	6.96	6.69
B_{iii}	95.13	99.26	4.17	6.71	43.54	99.16	56.09	12.20
C_i	0.92	45.75	98.00	11.72	5.43	68.52	92.08	65.48
C_{ii}	0.37	18.25	98.00	30.82	0.42	18.66	97.75	19.42
C_{iii}	1.20	59.01	97.98	0.99	0.78	37.05	97.90	26.13

It does, however, suffer on the smaller target class set, B. It can be seen that incorporating the pruning step yields an increase in overall accuracy. The execution time has increased as expected but, as pruning only takes place on the selected candidate rule, the percentage increase is minimal.

Table 3: Results for the Prism-ITRULE Hybrid with an without Pruning using Confidence

	Prism-ITRULE Hybrid				Prism-ITRULE Hybrid with Pruning			
DataSet	Acc%	Tent-Acc%	Abs Rate %	Time(s)	Acc%	Tent-Acc%	Abs Rate %	Time (s)
A_i	94.29	94.29	0.00	79.23	94.20	94.20	0.00	78.75
A_{ii}	95.91	95.91	0.00	77.95	95.90	95.90	0.00	80.56
A_{iii}	94.23	94.23	0.00	76.07	94.23	94.23	0.00	78.51
B_i	99.31	99.31	0.00	82.13	99.31	99.31	0.00	84.68
B_{ii}	98.50	99.43	0.94	78.03	99.46	99.46	0.00	79.42
B_{iii}	99.22	99.24	0.02	74.32	99.29	99.29	0.00	76.99
C_i	17.68	78.85	89.87	75.12	8.70	70.36	98.55	78.91
C_{ii}	40.40	87.46	53.81	74.50	42.14	88.98	52.65	77.58
C_{iii}	17.68	78.85	77.59	74.89	8.88	86.17	89.70	75.70

5.3 Comparison of Results

Table 4 contains the confidence, support, J-measure, Jmax and J-distance of 3 runs from each iteration of the algorithm for comparison. J-distance is the distance between the final J-measure of a rule and the Jmax, a low distance indicates that the rules were nearly optimised when they reach their finished form. Confidence is only used by the algorithm for pruning in the final stage of the Prism-ITRULE hybrid and support is recorded but not used at all in producing rules. Each result is the average of the top 10 rules produced by the algorithm when ranked by their J-measures.

It can be seen that there has been a steady rise in the support of the rules over each iteration. There is a drop in the confidence between the switch from Beam

Search to Separate and Conquer, this is misleading however as the high confidence of the original Beam Search approach is due to the over general form of the rules, leading to the very low support values. The beam search approach produces rules with a higher average J-measures and Jmax which is likely due to the less restricted data set used at each phase of rule induction.

Introducing confidence pruning into the algorithm increases the distance between the rule's J-measures and Jmaxs as would be expected. The J-measure of a rule can increase or decrease with the addition of any term (with the exception that it is bounded by 0 and Jmax). In one instance the J-measure has increased after applying pruning from 0.27 to 0.34, demonstrating that pruning to increase confidence is not necessarily at the cost of the other metrics used.

Table 4: Support, Confidence and J-measure values for each algorithm run on the A data sets

	Set	Supp.	Conf.	J-measure	Jmax	Jdist
Prism-ITRULE Hybrid with Confidence Pruning	A_i	0.38	0.68	0.38	0.51	0.13
	A_{ii}	0.29	0.59	0.34	0.47	0.13
	A_{iii}	0.32	0.59	0.33	0.47	0.14
	Avg	0.33	0.62	0.35	0.48	0.13
Prism-ITRULE Hybrid	A_i	0.29	0.68	0.30	0.35	0.06
	A_{ii}	0.22	0.54	0.27	0.30	0.03
	A_{iii}	0.25	0.59	0.28	0.39	0.11
	Avg	0.25	0.60	0.28	0.35	0.07
ITRULE with Partial Rule Dominance Pruning	A_i	0.18	0.49	0.33	0.51	0.18
	A_{ii}	0.35	0.98	0.74	1.08	0.34
	A_{iii}	0.36	0.96	0.66	1.02	0.36
	Avg	0.30	0.81	0.58	0.87	0.29
Basic ITRULE	A_i	0.15	0.94	0.46	0.56	0.10
	A_{ii}	0.15	0.95	0.51	0.61	0.10
	A_{iii}	0.17	0.94	0.50	0.62	0.12
	Avg	0.16	0.94	0.49	0.60	0.11

Particularly in terms of accuracy, the algorithm developed in this work has seen dramatic improvement whilst its other recorded metrics have remained steady. It follows that the resultant algorithm is effective at predicting and describing alarms ahead of time in line with this paper's goals.

6 Ongoing Work

In Section 3.2.2 it was mentioned that the class bias of the data set has been largely skewed through the introduction of Pre-Events. A further area of research is to be conducted on producing a Rule Induction algorithm that works with multi-label data (i.e. where an instance can belong to more than one target classes at the same time) using the generated Pre-Events [28, 17]. This would directly address the class bias problem and lead to more expressive rules being produced. As this work represents an extension to algorithms that, in their base form, are limited to one class attribute, this further extension has not been included here. There is also an additional data set containing the underlying IP network performance data from which the events are generated, it is hoped that by incorporating this additional data more interesting rule sets can be produced.

7 Conclusion

This paper outlines the progression of a Rule Induction Algorithm to produce rules that predict target events in the IP network. It does this by utilising the information available in the events rather than discarding this in favour of an Association Rule Mining Approach. A great deal of pre-processing has gone into the data set to clean the data and exploit the geolocation and temporal properties of the data set to increase the probability of producing rules that have strong causal links.

The ITRULE algorithm initially chosen for the project has been altered significantly due to the issues with using Beam Search on such a highly disproportionate distribution of feature values. Partial rule dominance led to producing a rule set with little variation in the rules and a tendency to favour numerical attributes over nominal ones. This was addressed with Partial Rule Dominance Pruning, however, the abstain rates are still high and accuracies were still low. Replacing Beam Search with a Separate and Conquer technique produced a large increase in accuracy which was higher still with the inclusion of confidence pruning though there are still issues with producing accurate rules from the dataset with the most sparse target class.

References

1. Rakesh Srikan Agrawal and Ramakrishnan. Fast algorithms for mining association rules. In *Proc. of 20th Intl. Conf. on VLDB*, pages 487–499, 1994.
2. Feng Cao, Martin Ester, W Qian, and A Zhou. Density-based Clustering over an Evolving Data Stream with Noise. *SDM*, 6:328–339, 2006.
3. Jadzia Cendrowska. PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27:349–370, 1987.
4. Nauman Chaudhry. Stream Data Management. In Nauman Chaudhry, Kevin Shaw, and Mahdi Abdelguerfi, editors, *Database*, chapter Introducti, pages 1–11. Springer US, 2006.
5. CPNI / Centre for the Protection of National Infrastructure. Telecommunications Resilience Good Practice Guide Version 4. Technical Report March, Centre for the Protection of National Infrastructure, 2006.
6. Ann Devitt, Joseph Duffin, and Robert Moloney. Topographical proximity for mining network alarm data. *Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data - MineNet '05*, page 179, 2005.
7. Lajos Jen Fülöp, Gabriella Tóth, Róbert Rácz, János Pánczél, Tamás Gergely, Árpád Beszédes, and Lóránt Farkas. Survey on Complex Event Processing and Predictive Analytics. In *Proceedings of the Fifth Balkan Conference in Informatics*, pages 26–31, 2010.

8. J. Fürnkranz, D. Gamberger, and N. Lavrač. *Foundations of Rule Learning*. Cognitive Technologies. Springer Berlin Heidelberg, 2012.
9. Johannes Fürnkranz. A pathology of bottom-up hill-climbing in inductive rule learning. *Algorithmic Learning Theory*, 2533(Section 2):263–277, 2002.
10. V. Gopalakrishnan, J. L. Lustgarten, S. Visweswaran, and G. F. Cooper. Bayesian rule learning for biomedical data mining. *Bioinformatics*, 26(5):668–675, 2010.
11. D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. 1997.
12. K. Karimi and H.J. Hamilton. TimeSleuth: a tool for discovering causal and temporal rules. *14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings.*, pages 375–380, 2002.
13. Imran Khan, Joshua Z Huang, and Nguyen Thanh Tung. 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.
14. Mika Klemettinen, Mannila Heikki, and Hannu Toivonen. Rule Discovery in Telecommunication Alarm Data. *Journal of Network and Systems Management*, 7(4), 1999.
15. WJ Leech. A rule-based process control method with feedback. *ISA transactions*, 26:73–78, 1986.
16. Map Data @2016 Geo-Basis-DE/BKG and Google. Google Maps, 2016.
17. Eneldo Loza Menc and Frederik Janssen. Towards Multilabel Rule Learning. 2008.
18. Ryszard S Michalski. On the quasi-minimal solution of the general covering problem. 1969.
19. Nacem Iqbal Mohammad Jaudet. Neural networks for fault-prediction in a telecommunications network. *8th International Multitopic Conference, 2004. Proceedings of INMIC 2004.*, pages 315–320, 2004.
20. Louis Perrochon, Walter Mann, Stephane Kasriel, and David C. Luckham. Event Mining with Event Processing Networks. *Methodologies for Knowledge Discovery and Data Mining. Third Pacific-Asia Conference, PAKDD-99 Beijing, China, April 2628, 1999 Proceedings*, pages 474–478, 1999.
21. J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, 1993.
22. Padhraic Smyth and Rodney M. Goodman. An Information Theoretic Approach to Rule Induction from Databases, 1992.
23. Brian J. Taylor and Marjorie A. Darrah. Rule extraction as a formal method for the verification and validation of neural networks. *Proceedings of the International Joint Conference on Neural Networks*, 5:2915–2920, 2005.
24. Gary Weiss and Haym Hirsh. Learning to predict rare events in event sequences. *Kdd-98*, pages 359–363, 1998.
25. Gary M Weiss. Data Mining in the Telecommunications Industry. *Data Mining and Knowledge Discovery Handbook*, pages 1189–1201, 2005.
26. Chris Wrench, Frederic Stahl, Giuseppe Di Fatta, Vidhyalakshmi Karthikeyan, and Detlef Nauck. *Research and Development in Intelligent Systems XXXII: Incorporating Applications and Innovations in Intelligent Systems XXIII*, chapter Towards Expressive Rule Induction on IP Network Event Streams, pages 191–196. Springer International Publishing, Cham, 2015.
27. Christos Yi, Byoung-Kee and Jagadish, HV and Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 201—-208. IEE, 1998.
28. Min-ling Zhang and Zhi-hua Zhou. A Review on Multi-Label Learning Algorithms. 26(8):1819–1837, 2014.