Journal of Theoretical and Applied Information Technology
30th September 2014. Vol. 67 No.3
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PERFORMANCE ENHANCEMENT OF CBS ALGORITHM USING FSGP AND FEAT ALGORITHM

[1]IMAM MUKHLASH, [2]MUHAMMAD IQBAL, [3]HANIM MARIA ASTUTI, [4]SUTIKNO

[1,2]Department of Mathematics, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia
[3]Department of Information System, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia
[4]Department of Statistics, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia
E-mail: [1]imamm@matematika.its.ac.id, [2]iqbalmohammad.math@gmail.com, [3]hanim03@gmail.com, [4]tikno@statistika.its.ac.id

ABSTRACT

Classification and prediction in temporal data is one of important tasks of data mining which are used in various fields such as financial data forecasting, meteorological data prediction, geographic data processing, and biomedical data analysis. A number of methods have been developed based on learning models such as decision tree, neural network or using other data mining task such as clustering, association rule and sequence pattern mining. One of the classification algorithms based on sequential mining is Classify-by-Sequence (CBS) algorithm. CBS algorithm is a combination of sequential pattern mining with probabilistic induction. However, this algorithm has a drawback regarding its computational time which is quite inefficient. This paper proposes an improvement of CBS algorithm to solve the computational time's inefficiency. The enhancement is conducted by improving the pruning part using FGSP and FEAT algorithm. These algorithms produce classifier that will use to classify testing data. The quality of classifier is then measured using $covacc_t$ parameter which is the combination of coverage and accuracy. To achieve the purpose of this research, we use meteorological data to classify rain or dry season. The simulation result show that the computational time of modified CBS algorithm with FGSP algorithm is faster than the modification using FEAT algorithm with the same level of accuracy.

Keywords: *Temporal Data, Classify by Sequence, Pruning Strategy, FEAT, FGSP.*

1. INTRODUCTION

Data mining is one of the computer science branches which has been developed rapidly and enormously applied in various fields such as science, engineering and business. The main goal of data mining is to find important and interesting patterns in the database. Several data mining techniques have been developed such as association, clustering, sequence pattern, and classification.

Time is an important determiner in relation to data. Real data that describes the condition of some object at several times is called temporal data [1]. Temporal data mining is a branch of data mining in which the data is determined by time.

Among various problems in data mining, classification and prediction of temporal data such as financial data, meteorological, and biomedical is one of important tasks in data mining [2]. Classification is a task learning function that maps a data item into a class from known classes [3]. Once the classification model is obtained, the model is then used to predict new data. Based on this prediction result, the accuracy of the method can be determined.

Prior studies mentioned that some researchers have developed a classification method for sequential data that combines sequential pattern mining [4] and some classic classification methods such as Naive-Bayes-Classifier. However, the result obtained by this method is still lacking. Following this, another classification method called Classify-by-Sequence (CBS) that combines sequential pattern mining and probabilistic induction was then developed. This algorithm provides (1) good performance on the classification of temporal dataset, (2) classification information to user at the same time, (3) high execution efficiency using induction probability, and (4) the classification results are available. CBS algorithm consists of two stages: extraction of sequential pattern as the main features of each class, and generation of classifier through the score of features extracted in the first stage based on probabilistic induction [5].

However, this algorithm still has a drawback, i.e. the execution result for predicting temporal data is still inefficient. This drawback is occurred due to this algorithm requires some parameters and the pruning process on CSP rules still needs to be improved. Dealing with these causes is considered to improve the execution result on prediction process of temporal data.

In addition to that, research on the implementation of data mining techniques for meteorological data has been performed [6],[7],[8]. Classification on the weather is a problem that has been widely used to predict tomorrow's weather. Researches on the weather are also under development. Some of these researches use the modified Apriori algorithm as in [9], and a classification algorithm based on association rules (CBA) at [10]. Research on comparison of classification techniques using association and sequence has been performed and it showed that the sequence approach produces better results than of the approach using association in meteorological data [11].

Based on the result above, we develop an algorithm to improve the performance of CBS algorithm by obtaining sequential pattern as the main feature of each class which is called Classifiable Sequential Patterns (CSP). This research also improves the pruning process using FEAT algorithm [12][13] and FGSP algorithm [14] in order to determine a more appropriate CSP to build CSP rules. As a case study, our proposed algorithm has been applied to classify weather data which has two classes, rainy and dry seasons. Since the research on this field is still in a growing state, this paper is expected to contribute to enrich literature on the classification of temporal data.

2. CLASSIFICATION BY SEQUENCE

2.1. *CBS* Algorithm

CBS algorithm (Classify by Sequence) is a classification algorithm that merges sequential pattern mining algorithms and induction probabilistic. The sequential pattern mining algorithm is based on Apriori algorithm. CBS itself is used for classification in temporal databases.

The main benefit of this algorithm is a simple implementation. In addition, the obtained architecture patterns yields obvious classification information to users.

Consider a large database *D* that stores numerous data instances, where each instance is composed of a sequence of temporal data. Each recorded data is associated with a class label. Let D_i represents all temporal data instance belonged to class *i*. Database $D=\{D_1, D_2, \ldots ,D_N\}$ and assume that there are *N-class* to the database. For each class, *dataset* D_i consists of a number of temporal data instances $I_j = <a_{j1}, a_{j2},\ldots,a_{ji}>$ where w a_{jk}represents the value of the k-th time point of instance I_j.

```
Input  : a temporal dataset(D)
Minimum support(min_sup)
Output
:temporal_classifier(CBS_classifier)
Procedure CBS(D, min_sup)
1. $CSP\_feature\_set = \mathbf{CSP\_Miner}(D, \min\_sup)$;
2. $CBS_{classifier} =$
         $\mathbf{Classifier\_Builder}(CSP\_feature\_set)$;
3. Output $CBS\_classifier$
```

Figure 1.CBS Algorithm [5].

CBS algorithm consists of two phases: the first stage of mining classifiable or CSP sequence which is the main feature of each class, and the second stage are to generate classifier from CSP resulting from the first stage. In general, the CBS algorithm is shown in Figure 1.

2.2 CSP_Miner

First, set all the frequent items of the entire dataset as 1-frequent sequences. Then, the construct root contains itemset, where the itemset as a leaf. As with Apriori algorithm, all the candidate sequences constructed from the expansion of the tree. Expansion tree is done by adding all the elements that may be as a leaf to evoke tree by all k-frequent sequences. Furthermore, the candidate sequences of different lengths k +1, only need to specify all the parents of the leaf nodes and add an element to each of the leaf, which is a child node. CSP-Miner algorithm process is shown in Figure 2.

```
Input  : Temporal dataset (D)
         Minimum_support(min_sup)
Output : all CSP from each class
         ($\{CSP\_set_c | c \in each\_class(D)\}$)
Procedures CSP_Miner(D, min_sup)
 1   for each class $c \in each\_class(D)$ do
 2      Let $D_c$, the dataset consists of
        all data on $D$ in class $c$.
 3      $F_1 = frequent\ itemset\ D_c CSP\_set_c = F_1$
```

```
4    Let $T$ as prefix tree built by $F_1$
5    $k = 2$;
6    while $T.root$ exist
7       Generated $T'$ from extension $T$
8       for each temporal s on $D_c$ do
9          Trace $T'$ with $s$ and add a
           leaves on reached node
10      end for
11      $T = T'$ after pruning non-
        frequent on subtrees
12      $F_k = all\_sequence$ on $T$
13      $CSP\_set_c = CSP\_set_c \cup F_k$
14      $k = k + 1$
15   end while
16 end for
   Output $\{CSP\_set_c | c \in all\_class(D)\}$
```

Figure 2.CSP-Miner Algorithm[5].

2.3 Classifier Builder

Classifier builder algorithm is the first step of CSP Miner algorithm. At this stage, the classifier built using all CSP was obtained from the CSP-Miner. Before developing classifier CSP, CSP that has no classification information should be removed. After the pruning, the remaining CSP has the features of each class in the classification. Further, a class of uncategorized CSP is determined in two steps based on the information available at the CSP. Firstly, we are intuitively able to determine that the CSP will be unclassified in class A if more attributes refer to class A. Secondly, for all attributes (nodes) on each CSP that has more than one value, the support of those attributes is found in more than one class. These supports indicate the possibility of information that holds little significance for the classification process so that the attributes can be trimmed from the CSP. Selection of the CSP for each class is based on their score calculation for each class C_i using the formula in [5]. CSP and Class C_i which has the highest score indicate that CSP is a class C_i and further a classifier. The process of this algorithm is shown in Figure 3.

```
Input  : CSP set for each class
         $(\{CSP\_set_x | x \in all\_class\_of(D)\})$
Output : CBS classifier($T$)
Procedures Classifier_Builder
         $(\{CSP\_set_x | x \in all\_class\_of(D)\})$
1.  Let $CSP\_dup$ is CSP set consists $CSP_s$
    on each class
2.  for each $x \in all\_class\_of(D)$ do
3.      for each $CSP_p$ on $CSP\_set_x$ do
4.      if $(p \in CSP\_dup)$
5.              removes $p$ from $CSP\_set_x$
6.      end for
7.  end for
8.  Let T as prefix tree that only have
    a root node
9.  for each $x \in all\_class\_of(D)$ do
10.     for each $CSP_p$ on $CSP\_set_x$ do
11.         put $p$ into T and add
            a class label at end node
12.     end for
13. end for
output T
```

Figure 3 Classifier-Builder Algorithm[5].

After the classifier is obtained, we can use the classifier to predict (classify) the value of a new class instance. If there are two classes with the same score value, then the class is randomly determined.

CBS algorithm has drawbacks including the CBS algorithm still needs some improvement process parameters and the resulting pruning rules. The enhancement of this algorithm has been performed by Iqbal and Mukhlash [13] by reducing process with utilizing pruning algorithm FEAT [12].

2.4 Frequence Sequence Generator

FEAT Algorithm

FEAT algorithm is an algorithm that works on mining frequent sequence generators, where a sequence generator is defined as one of the minimal subsequences in an equivalence class. Furthermore, the generators have the same ability to describe an equivalence class as their corresponding subsequence of the same equivalence class. Let an input sequence database SDB contains unordered list of item (each item can appear multiple times in a sequence) and can be denoted by $S = e_1e_2e_3 \dots e_n$. Given a prefix of sequence S, $S_{pre} = e_1e_2e_3 \dots e_i$. Given projected sequence of S_{pre} write as $e_{i+1}e_2e_3 \dots e_n$. In addition, given a sequence $S = e_1e_2e_3 \dots e_n$ and an item e', denoted $e_1e_2e_3 \dots e_{i-1}e_{i+1} \dots e_n$ by $S^{(i)}$, $e_ie_{i+1} \dots e_j$ by $S_{(i,j)}$ and $e_1e_2e_3 \dots e_ne'$ by $\langle S, e' \rangle$. Two pruning strategy used is forward and backward prune [1]. Forward prune on subsequence S_p and its expansion $S_p^* =< S_p, e' >$ is performed if $\sup(S_p) = \sup(S_p^*)$ and for each local frequent item u of S_p^* satisfy $SDB_{<S_p,u>} = SDB_{<S_p^*,u>}$, and then S_p^* can be pruned. While the backward pruning strategy on $S_p = e_1e_2e_3 \dots e_n$ performed if there is an index i = 1,2, ..., n-1 and j = i +1, i +2, ..., n such that $SDB_{(S_p)_{(1,j)}} = SDB_{((S_p)_{(1,j)})^{(i)}}$, then Sp can be prune. Subsequently, to verify whether the sequence is a plant or not is determined by checking the value of sup (S) and sup(Si). If sup (S) = sup (S(i)) then S is a generator.

Although this approach is able to reduce the computing time, but still has some drawbacks, among others the backward pruning process is slow computing time. Based on some earlier studies, in this paper will be explained the development of algorithms to improve the performance of CBS algorithms that can reduce the computational time while maintaining accuracy. Improved performance of the computational time is done using FSGP algorithm [14] in the process of pruning candidate classifiable sequence pattern (CSP).

FSGP Algorithm

Considering the performance of FEAT algorithm, it has a lack on pruning strategy. There are two strategies: forward pruning and backward pruning. However, it has enormous time cost for pruning the non-generator sequences that should be pruned since it causes many useless the database projection operation and the comparison of the projected databases [14]. To handle this lack, we are used FSGP algorithm. FSGP (Frequent Sequential Generator Pattern) algorithm is an algorithm based on depth first search performance is forward. Strategy of safe pruning is a basic of inclusion between sequence and its subsequence [14]. Suppose given transaction database SDB with minimum support is min_sup. For safe-pruning strategy, given a sequence $\mathrm{S_n} = \mathrm{e_1 e_2 e_3 \dots e_n}, \mathrm{i} = \mathrm{n} - 1$, where a subsequence of Sn is $\mathrm{S_{n-1}} = \mathrm{e_1 e_2 e_3 \dots e_{n-2} e_n}$ where ith item can be romeved from the sequence S_n if S_n can be substituted for S_{n-1} based on transaction database SDB. Finally, perform to testing whether S_n is a generator or not. S_n is a generator if $\mathrm{sup}|_{\mathrm{SDB}}(\mathrm{S_n^i}) = \mathrm{sup}(\mathrm{S_n})$.

3. IMPROVEMENT OF CBS ALGORITHM

In this paper, we changed the first step of CBS algorithm with modified CSP_Miner1 algorithm. We modify pruning part for built the CSP. When we get the CSP set then can be applied into second step of CBS algorithm to get classified rule. It is used to classify another temporal data. The modified of CBS algorithm can be seen in Figure 4.

```
Input  : A temporal dataset($D$)
         Minimum support($\min\_sup$)
Output: temporalclassifier($CBS\_classifier$)
Procedures CBS modification($D, \min\_sup$)
1.$CSP_{feature_{set}}$ = Modified $CSP_{Miner}$
         ($D, \min\_sup$);
2.$CBS_{classifier}$ = $Classifier_{Builder}$
         ($CSP\_feature\_set$);
3.OuputCBS_classifier
```

Figure 4. Modified CBS algorithm

3.1 CBS ImprovementWith FEAT Algorithm

At first, CSP_Miner algorithm on pruning process only prune candidate CSP which have support value less than a given minimum support. The modified of CSP_Miner algorithm on pruning process using FEAT algorithm is not only prune the candidate CSP which have support value less than a given minimum support, but also checking the candidate CSP whether generator or not. If the CSP candidates are a generator then do the forward prune or backward prune part as a condition given on Lemma 1 and Lemma 2 in [5]. Modified CSP_Miner1 algorithm shown on Figure 5.

```
Input  : temporal dataset (D)
         Minimum_support(min_sup)
Output : all CSP for each class
         ({$CSP\_set_c | c \in all\_class(D)$})
Procedures Modified CSP_Miner(D,
min_sup)
1 For each class $c \in all\_class(D)$ do
2    Let $D_c$ a dataset consists of all
     data on $D$ in class c.
3    $F_1$ = frequent itemset $D_c$
4    $CSP\_set_c$ = $F_1$
5     LetT as prefix tree built by $F_1$
6     $k = 2$;
7     while T.root exist
8      Generated T' from extension T
9       for each temporal s on $D_c$ do
10           $T^s = < T, s >$
11           CanPrune←False
12           generator←True;
13           if $\sup(D_T) = \sup(D_{T^s})$then
14              canPrune←
15              ForwardPrune;
16              Generator ←False;
17           end if
18           if cannotPrune then
19              BackwardPrune;
20           end if
21           if generator then
22              $T' = T \cup T^s$
23           end if
24        end for
25        $T = T'$ after pruning non-
```

```
            frequent on subtrees
            $F_k = all\_sequence\ on\ T$
            $CSP\_set_c = CSP\_set_c \cup F_k$
            $k = k + 1$
      end while
  end for
Output $\{CSP\_set_c | c \in all\_class(D)\}$
```

Figure 5. Modified CSP_Miner1 Algorithm

3.2 CBS ImprovementWith FSGP Algorithm

The second modification of the CBS algorithm is using FSGP algorithm. Difference with FEAT algorithm, FSGP algorithm prune of the candidates CSP using Safe Pruning with condition given by Lemma 3, Lemma 4 and Theorem 2 then we checking whether generator or not as a condition given by Theorem 3 in [14].

For example, let $S = \{S_1, S_2, S_3, S_4, S_5\}$ is a transaction database, where $S_1 = \{A, A, B, C\}$, $S_2 = \{A, C, D, A, B\}$,$S_3 = \{B, C, C\}$, $S_4 = \{E, B, A, B, C\}$, $S_5 = \{C, A, D, D, A\}$ and its own classes, which is S_1 belongs to class A, S_2 belongs to class B, S_3 belongs to class A, S_4 belongs to class A, S_5 belongs to class B. Given a minimum support value is 0.8, the frequency of each itemset is $\{(A{:}2), (B{:}3), (C{:}3), (E{:}1)\}$ and the 1-frequent itemsetis $\{(B{:}3), (C{:}3)\}$. After that, the possible extended tree is $\{(BC{:}3), (CB{:}0)\}$ and the 2-frequent itemset is $\{(BC{:}3)\}$ but there is no 3-frequent itemset because its not satisfy from the minimum support value. So, we get the classifier for class A is $\{(B \rightarrow class\ A), (BC \rightarrow class\ A)\}$. Modified CSP_Miner2 algorithm shown on Figure 6.

```
Input  : a Temporal dataset (D)
         Minimum_support(min_sup)
FS = ∅, GS = ∅
Output : all CSP from each
class({$CSP\_set_c | c \in all\_class(D)$})
Prosedures CSP_Miner(D, min_sup)
  For each class $c \in each\_class(D)$ do
      Let $D_c$ a dataset consists of all
      data D in class c.
      $F_1 = frequent\ itemset\ D_c$
      $CSP\_set_c = F_1$
      Let T as prefix tree built by $F_1$
      $k = 2$;
      while T.root exist
         Generated $T'$ from extension T
         for each temporal s on $D_c$ do
            $T^s = < T, s >$
            FS=ValidFreSeqMiner
            ($D|_{T^s}$, $T^s$, min_sup, FS)
            GS=GeneratorCheck(D, FS)
            $T' = T \cup FS$
         end for
         $T = T'$ after pruning non-
         frequent on subtrees
         $F_k = all\_sequence\ on\ T$
         $CSP\_set_c = CSP\_set_c \cup F_k$
         $k = k + 1$
  end while
  end for
Output $\{CSP\_set_c | c \in each\_class(D)\}$
```

Figure 6. Modified CSP_Miner2 Algorithm

4. TESTING AND DISCUSSION

We are used climate data in Indonesia, it is located in Perak-Surabaya from 1973 until 2009, obtained from the Bureau of Meteorology, Climatology and Geophysics of Indonesia (BMKG). We defined event on dataset as an instance consists of temperature, humidity, speed of wind, and air pressure above sea level which causes rain or not in a day. Considering from the Meteorology and climatology Bureau Indonesia classify weather into 2 category class

Dry (D)	0 mm/24 hour
Rain (R)	>0 mm/24 hour

Dataset obtained from the simulation data is one month of data (31 records), one year of data (366 records), five years of data (1830 records) and the overall dataset (7900 records).

The two parameters used to measure the performance of the algorithm is the computational time and covacc. Covacc is combination of accuracy (acc) and coverage (cov) which is defined by [15]

$$\text{covacc} = \text{acc}\,(\,1 + \text{cov}\,)$$

where the accuracy is stated with
acc(B→H) =

$$\frac{\text{number object } Y_{TR} \text{for body and head rule true}}{\text{number of object } Y_{TR} \text{rule body true}}$$

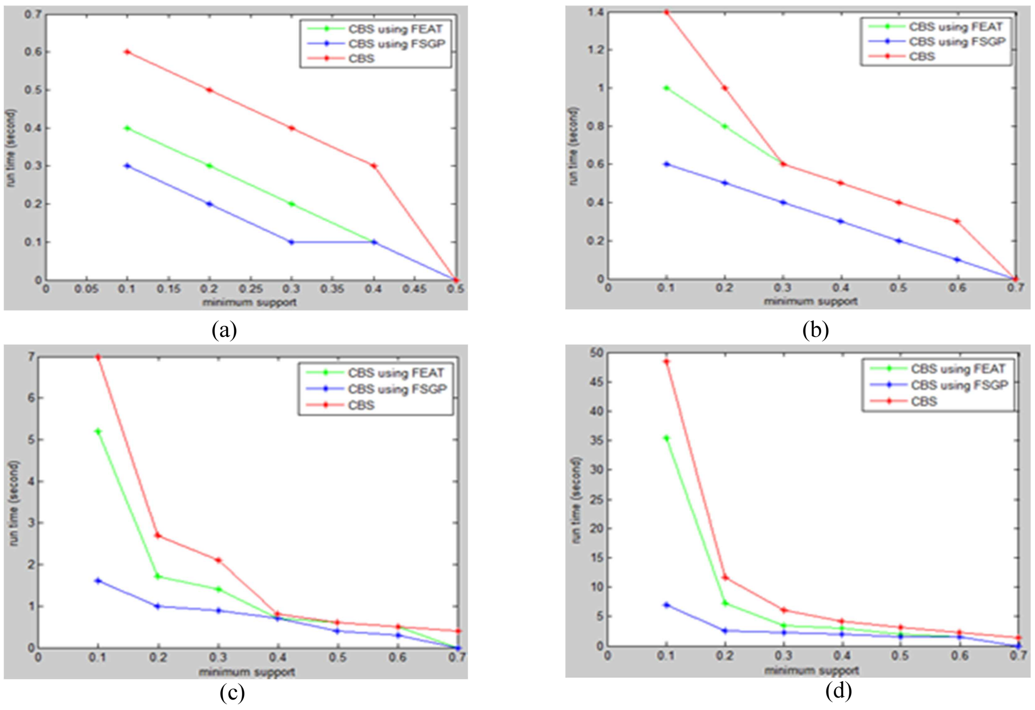

Figure 7. Time vs minimum support between modified CBS Algorithm with FEAT and FSGP algorithm for : (a) input 31, (b) input 366, (c) input 1830, and (d) input 7900

Table 1:Comparison of Accuracy, coverage,and covacc between CBS and modified CBS algorithm with FEAT and FSGP algorithm

Minimum support	Input	Accuracy (%)			*Coverage* (%)			*covacc*		
		CBS	CBS with FEAT	CBS with FSGP	CBS	CBS with FEAT	CBS with FSGP	CBS	CBS with FEAT	CBS with FSGP
0.1	31	99.65	99.65	97.65	99.41	99.41	94.63	1.99	1.99	1.90
0.2		80.2	80.2	51.50	98.85	98.85	5.73	1.59	1.59	0.54
0.3		78.25	78.25	92.71	88.52	88.52	98.92	1.47	1.47	1.84
0.4		91.85	91.85	92.71	99.8	99.8	98.92	1.84	1.84	1.84
0.5		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.1	366	77.64	77.64	99.29	99.72	99.72	91.20	1.55	1.55	1.90
0.2		77.61	77.61	78.36	99.56	99.56	95.19	1.55	1.55	1.53
0.3		77.71	77.71	77.71	100.0	100.00	100.00	1.55	1.55	1.55
0.4		77.59	77.59	77.59	99.48	99.48	98.48	1.55	1.55	1.55
0.5		77.47	77.47	77.47	98.9	98.9	98.90	1.54	1.54	1.54
0.6		77.47	77.47	77.47	98.9	98.9	98.90	1.54	1.54	1.54
0.7		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.1	1830	78.53	78.53	99.28	99.96	99.96	91.53	1.57	1.57	1.90
0.2		78.54	78.54	78.55	100.0	100.00	99.96	1.57	1.57	1.57
0.3		78.54	78.54	78.54	100.0	100.00	100.00	1.57	1.57	1.57
0.4		78.41	78.41	78.42	99.47	99.47	99.47	1.56	1.56	1.56
0.5		78.28	78.28	78.28	98.78	98.78	98.78	1.56	1.56	1.56
0.6		78.28	78.28	78.28	98.78	98.78	98.78	1.56	1.56	1.56
0.7		77.88	0.00	0.00	96.91	0.00	0.00	1.53	0.00	0.00
0.1	7900	80.37	80.37	98.19	99.34	99.34	89.07	1.60	1.60	1.87
0.2		80.51	80.51	80.51	100.0	100.00	100.00	1.61	1.61	1.61
0.3		80.51	80.51	80.51	100.0	100.00	100.00	1.61	1.61	1.61
0.4		80.51	80.51	80.51	100.0	100.00	100.00	1.61	1.61	1.61
0.5		80.51	80.51	80.51	100.0	100.00	100.00	1.61	1.61	1.61
0.6		80.51	80.51	80.51	100.0	100.00	100.00	1.61	1.61	1.61
0.7		79.92	0.00	0.00	96.57	0.00	0.00	1.57	0.00	0.00

$$\text{cov}(B \rightarrow H) = \frac{\text{number object } Y_{TR} \text{ for body rule true}}{\text{number object } Y_{TR}}$$

Simulation results of the comparison of computation time between the modified CBS algorithm for four records data set can be seen in Figure 7. While the comparison results of calculation of the covacc parameter for all algorithms shown in Table 1.

From the Figure 7, we see that the computing time of modified CBS algorithm using FSGP algorithm is faster than CBS algorithm using FEAT algorithm or CBS itself. It means that performance of the CBS algorithm using FSGP algorithm is more efficient than two other algorithms. From the Table 1, it's shown that the comparison covacc parameter between CBS algorithm and CBS algorithm using FEAT algorithm is similar. At the minimum support is 0.1, the parameter covacc of CBS algorithm using FSGP algorithm is higher than the other algorithm. When the minimum support is more than 0.1, the parameter covacc of CBS algorithm using FSGP algorithm as good as the other algorithms.

It happened because there is a process to check whether a sequence is a generator or not in the FEAT and FSGP algorithms. Furthermore, the computation time of FSGP algorithm is faster than FEAT algorithm because the FSGP algorithm only perform forward pruning without doing backward pruning process is just as found in FEAT algorithm.

However, experimental results also show that on the high support values (≥ 0.7), improvement of the CBS algorithm with FSGP algorithm and FEAT algorithm not able to produce a classifier. This is caused by pruning process of frequent item in both algorithms. In this experiment, when the minimum support is 0.5 for input 31 and the minimum is 0.7 support for input 366, 1830, 7990 after pruning is done by looking at the value of each support item exceeds the minimum support value is given then be checked against the value of the support items with extension support items obtained are different. It means that the item is not generating or infrequent item. Consequently, a classifier can't be built because there is no 1-frequent items are formed. To overcome this problem, the expanded tree can be developed so that it is capable to handling multiple values of support.

5. CONCLUSION

This paper discusses the enhancement of CBS algorithm using FSGP and FEAT algorithm. Improvement process is done on the CBS's pruning step. Furthermore, these algorithms are tested to solve the classification of the meteorological data. Simulation results show that the performance of the FSGP-based algorithm is more efficient than CBS algorithm and its improvement using FEAT algorithm. From the parameter covacc, CBS algorithm using FSGP algorithm is as good as the two other algorithms when the minimum support more than 0.1, but its higher than the two others algorithms when minimum support is 0.1.

For future work, the research will be focused on how algorithms that have been developed are used to deal with multiple support values and multi-class classification problems.

ACKNOWLEDGEMENTS
We would like to thank the Higher Education Directorate of the Education Ministry of Indonesia for supporting this research through the "Penelitian Unggulan Perguruan Tinggi" Research Grant 2013-2014.

REFERENCES:

[1] Hsu, W., Lee, M. L., Wang, J. (2008), *Temporal and Spatio-Temporal Data Mining*, IGI Global, Hershey and London.

[2] Mitsa, C., (2010), *Temporal Data Mining*, A Chapman & Hall/CRC., New York.

[3] Tan, N.P. Steinbach, M., Kumar, V., (2006), *Introduction to Data Mining*, Pearson Addison Weasly., New York.

[4] Zhou, C., Cule, B., Goethals, B. "Itemset Based Sequence Classification", *ECML PKDD 2013, Part 1, LN AI 8188, pp 353-368*

[5] Tseng, V. S. and Lee, C. H. (2009). "Effective *Temporal* Data Classification by Integrating Sequential Pattern Mining and Probabilistic Induction". *Journal of Expert System with Application*, Vol. 36, Issue 5, July 2009, pp 9524–9532

[6] Chen, Y. L., Wu, S. Y., Wang, Y. C. (2011). "Discovering multi label temporal patterns in sequence databases", *Journal of Information Sciences, 181, pp 398 – 418*

[7] Kohail, N.S., El Halees, M.A.(2011). "Implementation of Data Mining Techniques for Meteorological Data Analysis", *International Journal of Information and Communication Technology Research*, Vol. 1, (3).

[8] Mathur, Harsh (2013). "Sequential Mining of Multimedia Images by using SPADE Algorithm", *International Journal of Computer Science and Information Technologies, Vol. 4 (6), 791 – 795*

[9] S. Nandagopal, S.Karthik, V.P. Arunachalam (2010). *"Mining of Meteorological Data Using Modified Apriori Algorithm", European Journal of Scientific Research,* Vol.47 No.2, pp 295-308.

[10] Putri, K.P., Iqbal, M., Mukhlash, I.(2013). *"*Application Classification Based Association Rules Algorithm for Meteorological Data*", National Seminar on Mathematics, Universitas Negeri Yogyakarta, Indonesia.*

[11] Iqbal, M., Mukhlash, I., Astuti,H. M., (2013). "The Comparison of CBA Algorithm and CBS Algorithm for the Classification of Meteorological Data ". *2nd Information System International Conference, Sepuluh Nopember Institute of Technology, Indonesia.*

[12] Gao, C., Wang, J., He, Y., Zhou, L. (2008). "Efficient Mining of Frequent Sequence Generators", *17th International World Wide Web Conference, University of Beihang, Beijing.*

[13] Iqbal, M., Mukhlash, I. (2012). "Enhancement of Pruning Efficiency on CBS Algorithm Using FEAT Algorithm". *16th Mathematical National Conference, University of Padjajaran, Indonesia.*

[14] Yi, S.,Zao, T., Ma, S., Che, Z. (2011). "An Effective Algorithm for Mining Sequential Generators", *Procedia Engineering,* Vol. 15, pp 3653–3657

[15] N., Boutsinas, B., Giannikos, I. (2009). "A Method For Improving The Accuracy of Data Mining Classification Algorithm". *Computer and Operation Research*, Vol. 36 (10), pp 2829–2839