Predicting Flood in Perlis Using Ant Colony Optimization

Syaidatul Nadia Sabri and Rizauddin Saian
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia
E-mail: nadyasabry@gmail.com and rizauddin@perlis.uitm.edu.my

Abstract. Flood forecasting is widely being studied in order to reduce the effect of flood such as loss of property, loss of life and contamination of water supply. Usually flood occurs due to continuous heavy rainfall. This study used a variant of Ant Colony Optimization (ACO) algorithm named the Ant-Miner to develop the classification prediction model to predict flood. However, since Ant-Miner only accept discrete data, while rainfall data is a time series data, a pre-processing steps is needed to discretize the rainfall data initially. This study used a technique called the Symbolic Aggregate Approximation (SAX) to convert the rainfall time series data into discrete data. As an addition, Simple K-Means algorithm was used to cluster the data produced by SAX. The findings show that the predictive accuracy of the classification prediction model is more than 80%.

1. Introduction
Flood has terrible impacts on people as it disrupts their daily activities [1]. Flood also gives impacts to the country. In 2010, Perlis had experienced the worst flood whereby the flood had effected 47,000 hectares of land in Perlis. It damaged property belonging 63,000 people and led to monetary loss of nearly RM200 million. The cause of the flood, among others, was due to the existing river was not able to cope with the overflow of water from Timah Tasoh Dam in Perlis.

Analysis of rainfall is an important task nowadays. Many researches [2, 3, 4, 5, 6] had been carried out to predict the occurrence of flood due to heavy rainfall. A good classification prediction model is needed to develop a good flood early warning system. This is to ensure that the government is ready to handle the situation and as a result be able to reduce the effect of flood disaster.

This study developed a classification prediction model using rainfall data from three rainfall gauging stations in Perlis from year 1993 to 2014. The gauging stations are Kaki Bukit, Tasoh dan Padang Besar. These three stations are located nearby the Timah Tasoh reservoir. Timah Tasoh reservoir is the largest multipurpose reservoir in Perlis which serves as a flood mitigation and water supply.

The research phases is divided into two main phases, data pre-processing and model development as depicted by figure 1.

2. Data pre-processing
Rainfall data is a time series data. The data pre-processing phase is a phase to prepare the data so that it can be used in the classification algorithm chosen for this study. The main output of
the pre-processing phase is a discrete rainfall data.

The data pre-processing phase consists of five steps: data selection, data normalization, dimension reduction, data discretization and data clustering.

The first step in the data pre-processing step is data selection. This step will select the rainfall data from three target months with the highest average amount of rainfall. Table 1 depicted the average rainfall distribution for each month from year 1993 until 2014. Three months that received high average amount of rainfall are September, October and November. Hence, this study selected rainfall data on September, October and November.

| Month | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|
| Average | 37.1 | 56.8 | 138.3 | 161.8 | 140.5 | 148.6 | 160.7 | 188.6 | 215.4 | 265.4 | 212.3 | 140.1 |

The second step is data normalization. This step prepares data input on a suitable format before the implementation of data representation. The normalization reduced the gap between the data. The rainfall data is normalized at range of 0 and 1.

The third step is dimension reduction. This step reduced the data dimensionality. This study used Piecewise Aggregate Approximation (PAA) [7] to convert the original rainfall data (day, rainfall value) to new segments (week, average rainfall). The data size of each of the segments is seven days. The average rainfall for seven days is the value for each of the segments. The rainfall data dimension is 91 days (September to November). Each segment size is equal to 7 days. Besides, each of the segments will represent weekly average rainfall. PAA approximates the rainfall data of length \( n \) into vector \( \mathbf{\bar{C}} = (\bar{c}_1, \ldots, \bar{c}_w) \) of any arbitrary length \( w \leq n \) where each of \( \bar{c}_i \) is calculated as in (1).
\[ \bar{c}_i = \frac{w}{n} \sum_{j=\frac{i}{w}(i-1)+1}^{\frac{i}{w}n} c_j \]  \hspace{1cm} (1)

where

- \( \bar{c}_i \) is PAA data values
- \( n \) is original data length (days)
- \( w \) is the number of segments (weeks).

The forth step is data discretization. The discretization step converts PAA values into Symbolic Aggregate Approximation (SAX) symbol using Gaussian distribution table [8] depicted by table 2. This study divided the rainfall distribution into three classes \((\alpha = 3)\): light, moderate and heavy) as shown in table 3. According to table 2, when \( \alpha = 3 \), \( \beta_1 = -0.43 \) and \( \beta_2 = 0.43 \). Table 4 displays the rainfall data in SAX alphabet symbol format. Finally, from the SAX alphabet symbol, the data is transformed into SAX integer symbol format as for ‘a’ to 1, ‘b’ to 2 and ‘c’ to 3 as shown in table 5.

| Table 2. Gaussian distribution table [8]. |
|--------------------------------------------|
| \( \alpha \) | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| \( \beta_1 \) | -0.43 | -0.67 | -0.84 | -0.97 | -1.07 | -1.15 | -1.22 | 1.28 |
| \( \beta_2 \) | 0.43 | 0 | -0.25 | -0.43 | -0.57 | -0.67 | -0.76 | -0.84 |
| \( \beta_3 \) | 0.67 | 0.25 | 0 | -0.32 | -0.43 | -0.52 |
| \( \beta_4 \) | 0.84 | 0.43 | 0.18 | 0 | -0.14 | -0.25 |
| \( \beta_5 \) | 0.97 | 0.57 | 0.32 | 0.14 | 0 |
| \( \beta_6 \) | 1.07 | 0.67 | 0.43 | 0.25 |
| \( \beta_7 \) | 1.15 | 0.76 | 0.52 |
| \( \beta_8 \) | 1.22 | 0.84 |
| \( \beta_9 \) | 1.28 |

| Table 3. Symbolic representation of rainfall rate for alphabet of size 3. |
|---------------------------------------------------------------|
| Rainfall Distribution Class | Alphabet Symbol | Value |
|----------------------------|-----------------|-------|
| light                     | a               | \( \beta \leq -0.43 \) |
| moderate                  | b               | \(-0.43 \leq \beta \leq 0.43 \) |
| heavy                     | c               | \( \beta \geq 0.43 \) |

The final step in the data pre-processing phase is data clustering. The discrete data is clustered into three classes: normal, alert and danger. This study used Simple K-Means with Euclidian distance [9] to perform the task using Weka software [10]. Table 6 depicted the rainfall data after the data clustering step.
Table 4. Rainfall distribution in SAX alphabet symbol.

| week1 | week2 | week3 | ... | week10 | week11 | week12 | week13 |
|-------|-------|-------|-----|--------|--------|--------|--------|
| b     | b     | b     | ... | b      | c      | b      | b      |
| c     | b     | b     | ... | a      | c      | a      | b      |
| c     | b     | c     | ... | a      | b      | a      | a      |
| b     | b     | b     | ... | b      | a      | b      | b      |
| b     | b     | a     | ... | a      | a      | a      | b      |
| ...   | ...   | ...   | ... | ...    | ...    | ...    | ...    |
| b     | b     | a     | ... | b      | b      | c      | a      |
| b     | b     | b     | ... | b      | b      | b      | b      |
| c     | b     | b     | ... | b      | b      | b      | b      |
| c     | b     | b     | ... | b      | b      | b      | b      |

Table 5. Rainfall distribution in SAX integer symbol.

| week1 | week2 | week3 | ... | week10 | week11 | week12 | week13 |
|-------|-------|-------|-----|--------|--------|--------|--------|
| 2     | 2     | 2     | ... | 2      | 3      | 2      | 2      |
| 3     | 2     | 2     | ... | 1      | 3      | 1      | 2      |
| 3     | 2     | 3     | ... | 1      | 2      | 1      | 1      |
| 2     | 2     | 2     | ... | 2      | 1      | 2      | 2      |
| 2     | 2     | 1     | ... | 1      | 1      | 1      | 2      |
| ...   | ...   | ...   | ... | ...    | ...    | ...    | ...    |
| 2     | 2     | 1     | ... | 2      | 2      | 3      | 1      |
| 2     | 2     | 2     | ... | 2      | 2      | 2      | 2      |
| 3     | 2     | 2     | ... | 2      | 2      | 2      | 2      |
| 3     | 2     | 2     | ... | 2      | 2      | 2      | 2      |

Table 6. Rainfall data after data clustering step.

| week1 | week2 | week3 | ... | week10 | week11 | week12 | week13 | Class   |
|-------|-------|-------|-----|--------|--------|--------|--------|---------|
| 2     | 2     | 2     | ... | 2      | 3      | 2      | 2      | danger  |
| 3     | 2     | 2     | ... | 1      | 3      | 1      | 2      | normal  |
| 3     | 2     | 3     | ... | 1      | 2      | 1      | 1      | danger  |
| 2     | 2     | 2     | ... | 2      | 1      | 2      | 2      | normal  |
| 2     | 2     | 1     | ... | 1      | 1      | 1      | 2      | alert   |
| ...   | ...   | ...   | ... | ...    | ...    | ...    | ...    |         |
| 2     | 2     | 1     | ... | 2      | 2      | 3      | 1      | danger  |
| 2     | 2     | 2     | ... | 2      | 2      | 2      | 2      | danger  |
| 3     | 2     | 2     | ... | 2      | 2      | 2      | 2      | danger  |
| 3     | 2     | 2     | ... | 2      | 2      | 2      | 2      | danger  |

3. Model development
The classification prediction model was developed by training the discrete rainfall data to construct classification rules using rule induction technique. Rule induction is a technique to
extract set of rules from a data set [11, 12]. This study used a variant of Ant Colony Optimization (ACO) algorithm that was proposed by Parpinelli [13, 14, 15] named the Ant-Miner to perform the rule induction.

ACO Metaheuristic is a branch of artificial intelligence technique called swarm intelligence, introduced in the early 1990s [16, 17]. It is a probabilistic technique for solving combinatorial optimizations problem, which was inspired by the behaviour of cooperating ants in finding path from its nest to the food source [18]. The purpose of ACO metaheuristic is to find the best solution using a set of artificial ants that communicates indirectly using an item called the pheromone. The amount of pheromone on one of the trails used by the majority ants will increase as time passed by and will decrease on the less used trails. As a consequent, since ants tend to follow the pheromones, all or most of the ants will finally converge to the best trail, with the high density of pheromones, which happened to be the shortest trail from the nest to the food source. However, there are a small number of ants still using the longer branch. This is the effect of “path exploration”, where the density of the pheromone will not bias some of the ants.

Parpinelli proposed Ant-Miner algorithm in 2002 [13, 14, 15]. The Sequential Covering algorithm is the base for Ant-Miner algorithm, that extract rules directly from data. Sequential Covering algorithm [12] discovers rules in greedy fashion based on a certain evaluation measure. In other words, this algorithm select terms using some heuristic.

Ant-Miner uses a heuristic measure as evaluation measure to fill in the antecedent part of the rule, by selecting the best term to be included into the partial rule. The heuristic measure (3) is the normalization of entropy measures between terms (2). The algorithm selects one best rule from a set of discovered rules, based on a quality measure using some fitness function.

\[ H(W|A_i = V_{ij}) = -\sum_{w=1}^{k} (P(w|A_i = V_{ij}) \cdot \log_2 P(w|A_i = V_{ij})) \] (2)

\[ \eta_{ij} = \frac{\log_2 (k) - H(W|A_i = V_{ij})}{\sum_i x_i \cdot \sum_j \log_2 (k) - H(W|A_i = V_{ij})} \] (3)

where

- \( W \) is the class attribute
- \( a \) is the total number of attributes
- \( x_i \) is set to one if the attribute \( A_i \) was not yet used by the current ant, zero otherwise
- \( k \) is the number of classes

Ant-Miner differs from other Sequential Covering algorithms implementation because this algorithm also depends on a value called the pheromone, which contributes to the behaviour of exploration of the algorithm. Hence, Ant-Miner uses a probability (4) that is proportional to the product of heuristic value and pheromone level for that term, to add terms to a rule. Dorigo in his book, called this transition rule, random proportional transition rule [18].

\[ P_{ij}(t) = \frac{\tau_{ij} \cdot \eta_{ij}}{\sum_i \sum_j \tau_{ij} \cdot \eta_{ij}}, \forall i \in I \] (4)

where

- \( \tau_{ij} \) is the amount of pheromone at time \( t \)
- \( I \) is the set of attributes that are not yet used by the ant.
Ant-Miner initialized equally the pheromone level, $\tau_{ij}$ using (5).

$$\tau_{ij}(t = 0) = \frac{1}{\sum_{i=1}^{a} b_i}$$

(5)

where

- $a$ is the total number of attributes
- $b_i$ is the number of values in the domain of attribute $i$.

Ant-Miner updated the pheromone level after each ant colony has selected the best rule from a set of rules constructed by many ants in a colony using (6).

$$\tau_{ij}(t + 1) = \tau_{ij}(t) + \tau_{ij}(t)Q, \forall i, j$$

(6)

where $Q$ is the quality of a rule.

For the next ant colony, terms in the rule antecedent that have been selected by the previous ant colony will have a higher level of pheromone and will probably are more favoured than other terms.

Ant-Miner selects the best rule after each ant colony has created a set of rules, based on rule’s quality. This algorithm measures rule’s quality using a fitness function that depends on the product of sensitivity and specificity, which were adapted from the field of information retrieval (IR). The quality of rules, $Q$ is calculated using (7).

$$Q = \frac{TP}{TP + FN} \times \frac{TN}{FP + TN}$$

(7)

where

- $TP$ is the number of cases covered by the rule and having the same class as that being predicted by the rule
- $FP$ is the number of cases covered by the rule and having a different class from that being predicted by the rule
- $FN$ is the number of cases that are not covered by the rule while the class predicted by the rule
- $TN$ is the number of cases that are not covered by the rule and having a different class predicted by the rule.

4. Findings and discussion

This study tested the rainfall data with different values of number of ants which are 15, 20, 25, 30, 40 and 45. In addition, the minimum cases per rule is fixed to 5, maximum uncovered cases is fixed to 10, rule for convergence is fixed to 10 and number of iterations is fixed to 10000.

Figure 2 depicted the predictive accuracy for different values of number of ants. When the number of ants is equal to 30, Ant-Miner produced a classification prediction model with the highest accuracy which is 84.29%. Meanwhile, the lowest accuracy reading is when the number of ants is equal to 25. The higher the accuracy, the better the classification prediction model is.

The classification prediction model produced by Ant-Miner has a higher predictive accuracy as compared to J48, an implementation of C4.5 algorithm [19] in Weka. The predictive accuracy for the classification prediction model produced by J48, using the same 22 years rainfall data, produced a much lower value which is 77.27% only.

Experiment was also done for the number of rules. The fewer the number of rules, the better the prediction model is. Figure 3 depicted the results from the experiments done to test the
number of rules. It is found that when the number of ants is low, less than 35, the number of rules is small, but, increasing as the number of ants increased.

As for the number of terms, the results varies. However, the number of terms has a low value when the number of ants are 20 and 35 as depicted by figure 4.

5. Conclusion and future work
In this study the classification prediction model has been established for the rainfall recorded at three gauging stations in the north of Perlis. The findings show that predictive accuracy of the classification prediction model is promising at the value of more than 80%. As an addition, the predictive accuracy of the classification prediction model is better as compared to J48.

The classification prediction model established in this study can be used in predicting the occurrences of flood in Perlis. Hence, predicting can be used by the Perlis government to prepare the state as well as the people in order to reduce the effect of flood disaster.

One direction for future research is as follow. Since the class for rules constructed Ant-Miner will only be determined after the creation of each rule, the selected terms are not focused relevance. In other words, this algorithm will select a term that may be high in relevancy with the current set of previously selected terms, but not for the later assigned class. Therefore, like other ACO implementations for other fields, Ant-Miner might face the problem of stagnation, where optimized rule cannot be found, and thus make the program to run forever. So, it would be interesting to improve the terms selection strategy to reduce the stagnation problem.

Acknowledgments
This study was funded by Long Term Research Grant Scheme (LRGS/b-u/2012/UUM/Teknologi Komunikasi dan Informasi) and DKCP (600-UiTPs/PJIM&A/ST/DKCP (02/2012)).
authors would like to thank Department of Irrigation and Drainage Malaysia for supplying of required data.

References

[1] Gasim M B, Toriman M E, Abdullahi M G et al. 2014 International Journal of Interdisciplinary Research and Innovations 2 59–65

[2] Mokhtar S A, Ishak W H W and Norwawi N M 2014 Modelling of reservoir water release decision using neural network and temporal pattern of reservoir water level 2014 5th International Conference on Intelligent Systems, Modelling and Simulation (IEEE) pp 127–130

[3] Ashaary N A, Wan-Ishak W and Ku-Mahamud K R 2015 Forecasting model for the change of reservoir water level stage based on temporal pattern of reservoir water level Proceedings of the 5th International Conference on Computing & Informatics (ICOCI) pp 692–697

[4] Othman M, Saian R, Nazri M N, Kamal N S A and Liyana N 2013 Fuzzy forecasting for water level of flood warning system in perlis International Symposium on Mathematical Sciences and computing Research 2013 (iSMSC 2013) pp 104–109

[5] Muhamad N S and Din A M 2015 Exponential smoothing techniques on time series river water level data Proceedings of the 5th International Conference on Computing & Informatics (ICOI) pp 64–649

[6] Othman M, Azahari S N F and Massuut N A A 2016 Fuzzy spatial forecasting model of rainfall distribution for flood early warning Regional Conference on Science, Technology and Social Sciences (RCSTSS 2014) (Springer) pp 251–263

[7] Keogh E, Chakrabarti K, Pazzani M and Mehrotra S 2001 ACM SIGMOD Record 30 151–162

[8] Lin J, Keogh E, Lonardi S and Chiu B 2003 A symbolic representation of time series, with implications for streaming algorithms Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (ACM) pp 2–11

[9] Hall M, Witten I and Frank E 2011 Data Mining: Practical Machine Learning Tools and Techniques (Kaufmann, Burlington)

[10] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P and Witten I H 2009 ACM SIGKDD Explorations Newsletter 11 10–18

[11] Han J, Pei J and Kamber M 2011 Data mining: Concepts and Techniques (Elsevier)

[12] Tan P N, Steinbach M and Kumar V 2006 Introduction to Data Mining (Pearson Addison Wesley Boston)

[13] Parpinelli R S, Lopes H S and Freitas A A 2002 An ant colony algorithm for classification rule discovery Data Mining: A Heuristic Approach pp 191–208

[14] Parpinelli R S, Lopes H S and Freitas A A 2002 IEEE transactions on evolutionary computation 6 321–332

[15] Parpinelli R S, Lopes H S and Freitas A A 2001 An ant colony based system for data mining: Applications to medical data Proceedings of the genetic and evolutionary computation conference (GECCO-2001) pp 791–797

[16] Dorigo M, Maniezzo V and Coloni A 1991 Positive feedback as a search strategy Tech. Rep. 91-016 Dipartimento di Elettronica, Politecnico di Milano Milan, Italy

[17] Dorigo M, Maniezzo V and Coloni A 1996 Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 26 29–41

[18] Dorigo M and Stützle T 2004 Ant Colony Optimization (the MIT Press) ISBN 0-262-04219-3

[19] Quinlan J R 2014 C4.5: Programs for Machine Learning (Elsevier)