RULE EVOLUTION APPROACH FOR MINING MULTIVARIATE TIME SERIES DATA

Viet-An Nguyen and Vivekanand Gopalkrishnan
School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore

Keywords: Time Series, Data Mining, Rule Evolution.

Abstract: In the last few decades, due to the massive explosion of data over the world, time series data mining has attracted numerous researches and applications. Most of these works only focus on time series of a single attribute or univariate time series data. However, in many real world problems, data arrive as one or many series of multiple attributes, i.e., they are multivariate time series data. Because they contain multiple attributes, multivariate time series data promise to provide more intrinsic correlations among them, from which more important information can be gathered and extracted. In this paper, we present a novel approach to model and predict the behaviors of multivariate time series data based on a rule evolution methodology. Our approach is divided into distinct steps, each of which can be accomplished by several machine learning techniques. This makes our system highly flexible, configurable and extendable. Experiments are also conducted on real S&P 500 stock data to examine the effectiveness and reliability of our approach. Empirical results demonstrate that our system has substantial estimation reliability and prediction accuracy.

1 INTRODUCTION

Due to the rapid advances in database system technologies as well as the World Wide Web, the growth of data captured and stored all over the world has increased massively in the last few decades. Along with this explosion, data in various complex forms have also been created and used, requiring more efficient and sophisticated data mining techniques. Among these, time series data mining has been an attractive research topic recently (Last et al., 2001), (Keogh and Kasetty, 2002), (Last et al., 2004).

Mining time series data is the process of discovering and extracting knowledge from data, taken at equally spaced time intervals in a variety of domains including financial markets, gene expressions, natural phenomena, scientific and engineering experiments and medical treatments. In (Keogh and Kasetty, 2002), the time series data mining tasks are mainly divided into four categories: indexing (finding the best matching time series in database), clustering (grouping time series in database), classification (predicting the class of an unlabeled time series) and segmentation (constructing a model to approximate a time series). Most of these approaches focus only on the time series of a single attribute or *univariate time series* data. However, in many practical applications, the time series data captured have very high dimensionality which are characterized by a large number of attributes. Such data are called *multivariate time series* data, and they occur in various fields including:

- Medical observations: electrocardiograms (ECG) and electroencephalograms (EEG) of patients are usually recorded and used for diagnosis. Each ECG or EEG might be considered a time series which contains important information about the patient's health (Wei et al., 2005).
- Economic and financial data: each company in the financial market is represented by multiple time series over time. Modeling and classifying these time series can help investors gain more insight knowledge about the intrinsic health of each stock in the market (Wah and Qian, 2002).
- Action sequences or gestures: sequences of movements and gestures of hand and body can be used to denote a sign in Auslan (Australian Sign Language) [1]. Each sequence in a time series can be used to train a model to recognize every sign in the language (Kadous, 1999).

Univariate time series data normally store the captured values of an attribute over some period of time.

[1]http://archive.ics.uci.edu/ml/datasets/

Nguyen V. and Gopalkrishnan V. (2008).
RULE EVOLUTION APPROACH FOR MINING MULTIVARIATE TIME SERIES DATA.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - AIDSS*, pages 19-26
DOI: 10.5220/0001688500190026
Copyright © SciTePress

Modeling these data can help us understand the characteristics of the particular attribute as well as forecast the future attribute values. However, in many cases such as those mentioned above, single attribute time series might not contain enough information to describe complex systems. Contrarily, multivariate time series data contain the values of multiple attributes changing over time. Based on these kind of data, insightful correlations among different attributes can be identified and captured to build more robust and reliable models.

In spite of these advantages of multivariate over univariate time series data, some problems make modeling and forecasting multivariate time series data challenging and difficult. Due to the intricate correlations among multiple attributes changing over time, these problems usually require complicated models which might lead to low accuracy and efficiency. In addition, the high dimensionality makes the intrinsic characteristics of the data too complex for the built models to capture.

In this paper, we present a novel approach to model and predict the behaviors of multiple attributes time series databases. In our approach, from each data set at each time point, a set of classification rules is generated by supervised machine learning techniques and optimized by pruning using some Interestingness Measure. The problem of predicting the target attribute based on historical data is then transformed to the problem of predicting the classification rules. Top-*k* evolutionary trends containing *k* pairs of rules which represent the *k* most important and significant changes in the data over time are extracted from data sets at consecutive time points. The series of top-*k* evolutionary trends are then used to build models to predict those trends at the current time point. Each step in our proposed approach can be accomplished by various learning techniques which makes the framework highly flexible and configurable.

In Section 2 we present some preliminary information and method to transform our original problem to a new equivalent problem. Section 3 outlines our approach in detail, including a preliminary step (data preprocessing) and three main steps (rule pruning, top-*k* evolutionary trends mining and top-*k* evolutionary trends predicting). Experimental results with analyses are shown in Section 4 to show the effectiveness and reliability of our proposed framework. Finally, Section 5 summarizes our contributions and presents some directions for future work.

2 BACKGROUND

2.1 Time Series Database

In our problem, a time series database DB contains an ordered sequence of data sets that arrives in timely order.

$$DB = \{D(t) \mid t \in [1,T]\}$$

Each data set $D(t)$ at time point t is a set of instances.

$$D(t) = \left\{I^k(t) \mid k \in \kappa(t)\right\}, t \in [1,T]$$

where

- k is the id of the instance
- $\kappa(t)$ is the set of instance's ids at time point t
- $I^k(t)$ is the instance with id k at time point t. $I^k(t) = \left\{a_i^k(t) \mid i \in [1,n]\right\}, t \in [1,T], k \in \kappa(t)$; where $a_i^k(t)$ is the value of the i^{th} attribute. The attributes can be either numeric or nominal.

Corresponding to each instance, $c^k(t)$ denotes the class (or target attribute) of the instance $I^k(t)$ at time point t. The target attribute c is nominal, and its domain is denoted by $dom(c) = \{c_m \mid m \in [1,M]\}$.

2.2 Classification Rule

A classification rule is an implication of the form:

$$R^{c_m} = \left(\bigwedge_{i \in [1,n]} (a_i \; op \; A_i) \rightarrow (c = c_m) \right), c_m \in dom(c)$$

where:

- a_i is the i^{th} attribute which can be either nominal or numeric.
- op is a relational operator. For numeric attribute, $op \in \{<,=,>\}$, while for nominal attribute $op \in \{=,\neq\}$.
- A_i is a value that belongs to the domain of a_i.
- c is the class or target attribute.
- c_m is a class value that belongs to $dom(c)$. c_m is also considered the class of the rule R^{c_m}.

In a rule, each term $(a_i \text{ op } A_i), \forall i \in [1,n]$ is called an antecedent of the rule, whereas the rule consequent refers to the part after the implication arrow $(c = c_m)$. In this paper, when referring to a rule, the rule class and rule consequent are used interchangeably.

A rule might not contain all the normal attributes in its antecedents. In order to have a standard format for all the rules, the classification rules would be reformatted. Let N denote the set of numeric normal attributes and O denote the set of nominal normal attributes in each data set D. Then, the new formatted classification rules can be expressed as follows:

$$R^{c_m} = \left(\bigwedge_{\forall i \in N} (A_i^L \; op^L \; a_i \; op^U \; A_i^U) \wedge \bigwedge_{\forall j \in O} (a_j = A_j) \right.$$
$$\left. \rightarrow \quad (c = c_m) \right)$$

where:

- a_i is a numeric attribute
- A_i^L, A_i^U are the lower and upper boundary numeric values of attribute a_i
- op^L, op^U are the lower operator and upper operator respectively which can be either $<$ or $\leq$
- a_j is a nominal attribute
- A_j is the nominal value of attribute a_j

In a re-formatted rule, a pair (A_i^L, A_i^U) and a value A_j are used to determine the values of a numeric attribute a_i and a nominal attribute a_j respectively.

2.3 Set of Rules

A set of classification rules $RS(t)$ generated from the data set $D(t)$ can be defined as:

$$RS(t) = \left\{ R_j^{c_m}(t) \mid c_m \in dom(c), j \in [1, l_{c_m}] \right\}, t \in [1, T]$$

where:

- c_m is a class value in the domain of the target attribute $dom(c)$
- l_{c_m} is the number of rules which have class value c_m
- $R_j^{c_m}(t)$ is the j^{th} classification rule in the subset of rules which have class value c_m

From the above definitions, we can also define $RS^{c_m}(t)$ as a subset of $RS(t)$ which contains all the rules having class c_m in $RS(t)$. $RS^{c_m}(t) = \left\{ R_j^{c_m}(t) \in RS(t) \mid j \in [1, l_{c_m}] \right\}, c_m \in dom(c), t \in [1, T]$.

2.4 Problem Transformation

As stated in Section 1, our goal is to model and predict the target attribute based on historical data of both the normal and target attributes. By using this kind of data, underlying characteristics of the data and correlations between various attributes can be captured by the learnt model. In our approach, in order to generate the model satisfying the above constraints, the original problem is transformed to a new equivalent problem.

From each data set $D(t)$ at time point t, a set of rules $RS(t)$ will be generated, in which a rule $R(t)$ of $RS(t)$ is an implication from the normal attributes $(a_1(t), a_2(t), ..., a_n(t))$ at time point t to the target attribute $c(t+1)$ at the next time point $(t+1)$.

$$R^{c_m}(t) = \left(\bigwedge_{i \in [1,n]} (a_i(t) \; op \; A_i(t)) \rightarrow (c(t+1) = c_m) \right)$$

A set of classification rules can be generated from each data set by various supervised machine learning techniques. It is this characteristic that makes our approach highly flexible and configurable. Details on choosing suitable techniques are discussed in Section 4.2.

Assume T is the current time point, then the transformed problem can be stated as follows: Given the sets of rules $RS(1), RS(2), ..., RS(T-1)$ generated from data sets $D(1), D(2), ..., D(T-1)$ respectively, predict the k most important rules at the current time point T.

3 RULE EVOLUTION APPROACH

3.1 Preprocessing Data

In order to have a higher quality data to use in later steps of the knowledge discovery process, data preprocessing is required. In this step, we present two basic components of data preprocessing:

- Data cleaning: to identify and replace inconsistent and noisy values.
- Data normalization: to transform our raw data into appropriate forms for using in later steps.

Data Cleaning. Due to the huge amount of data processed, it is obvious that our time series database tends to be incomplete, noisy and inconsistent. In order to fill in missing values, smooth out noise while identifying outliers and correct inconsistencies in the data, data cleaning is performed.

Because of various reasons, the raw data have missing values. In order to have a complete and consistent database to process, those missing values need to be filled. Here we replace the missing values by the mean (for numerical attributes) or the median (for nominal attributes.

In order to detect and replace outliers in numerical data, we perform Boxplot Analysis (Tukey, 1977). Boxplot, invented in 1977 by John Tukey, is a popular way of visualizing a distribution. The analysis is based on *five-number summary* including five values: LowerWhisker, Q_1 (25th percentile), Median, Q_3 (75th percentile) and UpperWhisker. LowerWhisker and UpperWhisker are the most extreme observations of a distribution. Any value of the considered distribution falling outside this range (from LowerWhisker to UpperWhisker) can be considered an outlier and should be replaced by an appropriate value (LowerWhisker if it is smaller than LowerWhisker or UpperWhisker if it is greater than UpperWhisker).

Data Normalization. Because different numerical attributes have different ranges of values, data normalization is needed in order to have all the attribute values in a standard predefined range. In this step, we normalize all attribute data to the range [0,1]. Each attribute value u is normalized to $\bar{u}$ by the following equation:

$$\bar{u} = \frac{u - u_{min}}{u_{max} - u_{min}}$$

where u_{min} and u_{max} are the smallest and largest values of the attribute data respectively.

3.2 Pruning Rules

As discussed in the Section 2.4, from each data set $D(t)$, a set of classification rules $RS(t)$ is generated. Each rule $R^{c_m}(t)$ in the set shows that an instance will be classified as class c_m at timestamp t if all the antecedents of the rule $R^{c_m}(t)$ are satisfied. In literature, various rule induction techniques have been proposed. Two of the most successful algorithms are ID3 (Quinlan, 1986) and its successor C4.5 (Quinlan, 1993), basically due to their simplicity and comprehensibility. Both algorithms are used to generate decision trees, from which classification rules can be created. (Su and Zhang, 2006) proposed a fast decision-tree learning algorithm which is capable of reducing the time complexity compared to C4.5 while still preserving high accuracy. Other popular rule induction algorithms include Decision Table (Kohavi, 1995), OneR (Holte, 1993), PRISM (Cendrowska, 1987), CART (Breiman et al., 1984), etc. Generally, sets of classification rules can be generated with high accuracy and large data coverage by such rule induction algorithms. However, if the data set is huge, the number of rules generated in each $RS(t)$ will be large, leading to over-fitting the data, especially in noisy data sets, and also causing difficulties for human experts to evaluate the rules. In order to avoid these drawbacks, rule pruning is performed based on some Interestingness Measures.

Interestingness Measures are used to evaluate and rank the discovered rules (or patterns) generated by the data mining process. Various works have been done on choosing the right interestingness measure for specific kinds of discovered patterns (Tan et al., 2002). In (Geng and Hamilton, 2006), these measures are generally divided into three categories: objective measures based on the statistical properties of the rules, subjective measures based on the human knowledge, and semantic measures based on the semantics and explanations of the rules themselves. In our approach, we use the following objective measures:

- Support $S(R)$ of a rule R, which measures how often the rule holds in the data set.
- Confidence $C(R)$ of a rule R, which measures how often the consequent is true given that the antecedent is true.
- Data coverage $DC(R)$ of a rule R, which measures how comprehensive is the rule.

Based on these Interestingness Measures, we define a combined general measure $IM(R)$ of rule R which shows how interesting the rule is. $IM(R)$ is calculated as follows:

$$IM(R) = \frac{w_S \times S(R) + w_C \times C(R)}{w_S + w_C}$$

where w_S and w_c are the weights of Support and Confidence respectively.

Algorithm 1 shows a greedy method to prune uninteresting rules from a given set. First, the Interestingness Measure IM of each rule in the set is calculated. Rules are then sorted in an ascending order based on their IM. After sorting, rules with small IM value are pruned until the coverage of the remaining set of rules is less than a predefined threshold.

Algorithm 1: PRUNE UNINTERESTING RULES

```
Input: $uRS$ - An unpruned set of rules
Input: $\theta$ - The threshold for data coverage
Output: $pRS$ - The pruned set of rules
1  foreach $R \in uRS$ do
2      calculate $IM(R)$
3      calculate $DC(R)$
4  sort $RS$ in ascending order by $IM$
5  $pRS = uRS$
6  initialize $coverage = 1$
7  for $i = 1 \rightarrow | uRS |$ do
8      if $coverage - DC(R_i) > \theta$ then
9          prune $R_i$ from $pRS$
10 return $pRS$
```

After pruning, the remaining set contains rules which have high Interestingness Measures and guarantee that their data coverage is still greater than a user-defined threshold θ, $(0 < \theta < 1)$. The pruned sets of rules will be used in later steps to predict the set of rules at the current time point.

3.3 Mining Top-k Evolutionary Trends

Given the sets of rules $RS(1)$, $RS(2)$,... , $RS(T-1)$ in the previous time points, how can we predict the set of rules $RS(T)$ at the current time point T? In this step, we introduce a novel approach to predict $RS(T)$ using rule evolution methodology.

The problem of mining interesting rules (or patterns) from time series has been addressed by various works (Weiss and Hirsh, 1998), (Povinelli, 2000) and (Hetland and Saetrom, 2002) etc. (Keogh and Kasetty, 2002) classifies these approaches into two types: supervised and unsupervised. In supervised methods, the rule target is known and used as an input to the mining algorithm; while unsupervised approaches are trained with only the time series itself without any known target. Almost all of these approaches use some form of evolutionary computation such as genetic programming or genetic algorithms. In our approach as shown in the previous sections, from each data set $D(t)$, a set of rules $RS(t)$ is generated. Each set of rules $R(t)$ can be considered a model of the data at time point t which shows the correlations between the normal attributes at time point t and the target attribute at time point $(t+1)$. This ensures that the most updated characteristics and possible changes of the data can be captured by our generated rules or models. However, having the model of the latest data is not good enough. We also have to make use of the historical models. Here, we consider that, from the time point t to the next time point $(t+1)$, rules in the set $RS(t)$ evolve to rules int the set $RS(t+1)$. Our goal is to capture these evolutionary trends of rules over time.

As defined in section 2.2, our classification rule contains two main components: antecedents and consequent. The antecedents consist of ranges of n attributes, satisfying which an instance will be classified as the corresponding rule's consequent. Based on that, a classification rule can be considered a hypercube in an n-dimensional space, where each dimension corresponds to an attribute.

Minkowski-form distance measures the distance between two points in high dimensional space, and is defined as

$$d_{L_r}(H,K) = \left(\sum_{i=1}^{n} |h_i - k_i|^r\right)^{\frac{1}{r}} \qquad (1)$$

where H and K are two points in n-dimensional space with their coordinates h_i and k_i in the i^{th} dimension respectively. The two most commonly used Minkowski-forms are L_1 (*Manhattan distance*) and L_2 (*Euclidean distance*). In our approach, the distance between two arbitrary rules is defined as the Euclidean distance between the two hypercubes' centers corresponding to the two rules and is computed by the following equation:

$$d_{L_2}(R_1,R_2) = \left(\sum_{i=1}^{n} \left|\frac{A_{2i}^U - A_{2i}^L}{2} - \frac{A_{2i}^U - A_{2i}^L}{2}\right|^2\right)^{\frac{1}{2}} \qquad (2)$$

In equation (2), A_{1i}^U and A_{2i}^U are the upper boundary values of antecedent a_i of R_1 and R_2 respectively. Similarly, A_{1i}^L and A_{2i}^L are the lower boundary values of antecedent a_i of R_1 and R_2 respectively. The Euclidean distance $d_{L_r}(R_1,R_2)$ shows how far the two rules R_1 and R_2 are from each other in n-dimensional space. The smaller the distance is, the more similar the two rules are. Applying this property in temporal data, given two rules $R(t)$ and $R(t+1)$ from two consecutive time points t and $(t+1)$ respectively, a small $d_{L_r}(R(t),R(t+1))$ means there is a high probability that the first rule $R(t)$ will evolve to the second one $R(t+1)$ over the period from t to $(t+1)$.

Let's consider two sets of rules $RS(t)$ and $RS(t+1)$ generated from data sets $D(t)$ and $D(t+1)$ at two consecutive time points t and $(t+1)$ respectively. Based on (2), the distance between every two rules, each from $RS(t)$ and $RS(t+1)$ respectively can be computed. Each such pair of rules is considered an *evolutionary trend*, whose significant level can be expressed by the Euclidean distance. The smaller the distance is, the more similar the two rules are, and thus, the more significant the pair of rules is in the evolutionary process. Therefore, all the pairs of rules are sorted according to their distance and the k pairs with lowest distance form the *top-k evolutionary trends* which contain and represent the most important underlying changes of the data over two consecutive time points. Thus, from each pair of consecutive time points t and $(t+1)$, a set of k pairs of rules denoted by $evoK(t,t+1)$ is extracted and used to predict the top k evolutionary trends at the current time point T in the next step of our approach.

3.4 Predicting Top-k Evolutionary Trends

These k evolutionary trends are ranked in ascending order according to the Euclidean distance between each pair of rules in the trend. Using all the data sets from time point 1 to time point $(T-$

1), we can form a series of k evolutionary trends $\{evoK(t,t+1) \mid t \in [1, T-2]\}$. Based on these k series of trends, in this step we predict the k evolutionary trends from time point $(T-1)$ to the current time point T, with which the most important movements of stock prices in the near future can be forecasted.

Recall that, each evolutionary trend $evoK(t,t+1)$ over the period from time point t to $(t+1)$ contains k pairs of rules. Each pair contains $R(t)$ and $R(t+1)$, where $R(t)$ is considered to evolve to $R(t+1)$ over the period from t to $(t+1)$. In order to capture the underlying changes of the data that each pair of rules in $evoK(t,t+1)$ represents, we define the operators, by which each antecedent of $R(t)$ evolves to the corresponding antecedent of $R(t+1)$.

Let $evoOp^{a_i}_{(R(t),R(t+1))}$ denote the operator by which the antecedent a_i evolves from the value a_{it} of the rule $R(t)$ to the value $a_{i(t+1)}$ of the rule $R(t+1)$. The domain of $evoOp^{a_i}_{(R(t),R(t+1))}$ is denoted by $dom(evoOp)$ and contains three operators: *increased* (I), *unchanged* (U) and *decreased* (D). Each operator is defined based on comparing the values of antecedent a_i from time point t to $(t+1)$ as follows:

$$evoOp^{a_i}_{(R(t),R(t+1))} = \begin{cases} I \Leftrightarrow a_{it'} \geq a_{it} + \varepsilon \\ U \Leftrightarrow a_{it} + \varepsilon \geq a_{it'} > a_{it} - \varepsilon \\ D \Leftrightarrow a_{it} - \varepsilon > a_{it'} \end{cases}$$

By extracting all the evolutionary operators from every pair of classification rules in our k series of evolutionary trends, we can obtain k series of evolutionary operators. These series of operators describe how the data change over time and capture the most important underlying characteristics of our multivariate time series data. Using each generated series of evolutionary operators, regression models can be built by various machine learning techniques and used to predict the future evolutionary trends. Details about choosing the learning techniques are discussed in section 4.3.

4 EXPERIMENTAL RESULTS AND ANALYSES

4.1 Database

We conducted experiments on data from the S&P 500 database. The whole database consists of quarterly data sets from 1975 to 2006. Each quarterly data set contains 500 instances corresponding to 500 companies in the S&P 500 index, 13 normal attributes which are financial ratios, and a target attribute which indicates the status of the company.

Table 1: Financial ratios used in our database.

Type	Ratio
Liquidity ratio	Current ratio Quick ratio
Debt ratio	Debt ratio Long term debt common equity
Profitability ratio	Working capital per share Operating margin before depreciation Operating margin after depreciation Pretax profit margin Return on asset Return on equity Net profit margin
Market ratio	Price earning Price to book

Table 1 shows all financial ratios used in our experiments. They are divided into four categories and are verified by financial experts as being able to describe the intrinsic status of a company at a given time. The domain of the target attribute is $\{$*good*(G), *average*(A), *bad*(B)$\}$ showing the status of the company as based on its return.

4.2 Rule Generating

As discussed in Section 3, from each data set $D(t)$ at time point t, a set of classification rules $RS(t)$ is generated, using the C4.5 decision tree learning method (Quinlan, 1993). Each data set $D(t)$ is trained using J48 provided by WEKA [2] (Witten and Frank, 2005) to generate a decision tree. From each decision tree, a set of classification rules are obtained. The experiments are conducted with the default configurations in WEKA.

4.3 Top-k Evolutionary Trends Prediction

As stated in Section 3.4, for each series of evolutionary operators generated from our top-k evolutionary trends, a regression model is built. In our experiments, we apply three popular supervised machine learning methods

- Multi-nominal Logistic Regression (MNL)
- Multi-layered Perceptron Neural Network (MLP)
- Support Vector Machines (SVM)

[2] http://www.cs.waikato.ac.nz/ml/weka/

We use the implementations of the above three algorithms in WEKA with their default configurations. Results of these experiments are shown in Table 2.

Table 2: Results of top-*k* evolutionary trends prediction models using MNL, MLP and SVM.

Financial ratio	Accuracy (%)		
	MNL	MLP	SVM
1. Current ratio	61.83	64.12	**81.70**
2. Quick ratio	73.58	**84.38**	76.62
3. Working capital per share	**83.51**	76.26	67.91
4. Net profit margin	**83.99**	69.81	66.92
5. Operation margin before depreciation	77.96	72.80	**79.60**
6. Operation margin after depreciation	69.30	74.50	**81.88**
7. Pretax profit margin	76.37	**79.54**	76.39
8. Return on assets	74.87	65.92	**78.96**
9. Return on equity	**76.46**	64.35	66.40
10. Long term debt common equity	73.94	**81.87**	81.76
11. Price earning	74.48	**82.4**	74.78
12. Price to book	73.27	75.36	**77.29**
13. Debt equity	**78.49**	70.00	72.83
Average	75.23	73.95	**75.62**

On each financial ratio, three models using three above machine learning techniques are built. However, we observe that each technique achieves high accuracy only on certain financial ratios but performs badly on others. For example, SVM outperforms the two other techniques on *Current ratio* (81.7%) and *Operation margin after depreciation* (81.88%), but suffers a poor accuracy of 66.4% on *Return on equity*. MLP has higher accuracy than the other two on *Quick ratio* as well as *Price earning*. Although MNL successfully models *Net profit margin* (83.99%) and *Working capital per share* (83.51%), it has the lowest accuracy on *Current ratio* (61.83%). That's why the overall accuracies of the three techniques MNL (75.23%), MLP (73.95%) and SVM (75.62%) are relatively low and approximately equal.

In order to have a more accurate and reliable model, a combined model using three above techniques is used. The combined model is built by choosing the technique which has the highest accuracy for each financial ratio. Table 3 shows the chosen learning techniques with their corresponding accuracy for each financial ratio. As we can see, the overall accuracy of the combined model increases to 80.77%, which outperforms every individual model.

Table 3: Results of top-*k* evolutionary trends prediction models using combined learning techniques.

Financial ratio	Learning technique	Accuracy (%)
1. Current ratio	SVM	81.70
2. Quick ratio	MLP	84.38
3. Working capital per share	MNL	83.51
4. Net profit margin	MNL	83.99
5. Operation margin before depreciation	SVM	79.60
6. Operation margin after depreciation	SVM	81.88
7. Pretax profit margin	MLP	79.54
8. Return on assets	SVM	78.96
9. Return on equity	MNL	76.46
10. Long term debt common equity	MLP	81.87
11. Price earning	MLP	82.4
12. Price to book	SVM	77.29
13. Debt equity	MNL	78.49
Average		80.77

5 CONCLUSIONS

In this paper, we have proposed a new approach to model and predict the behavior of target attributes in a multi-attribute time series problem. In our approach, the original problem of predicting future values of target attributes is transformed to a new equivalent problem of predicting the set of rules.

We have developed a framework to solve the transformed problem based on the rule evolution methodology. The framework consists of a preliminary step - data preprocessing and three main steps including rule pruning - prune uninteresting rules, top-*k* evolutionary trends mining - determine *k* most important evolutionary trends of every consecutive data set, and top-*k* evolutionary trends prediction - predict the set of k most significant trends in the near future. Each step of our framework can be accomplished by different individual as well as combined learning techniques which makes it highly flexible and configurable. The framework is validated on real S&P 500 stock data to show its effectiveness and reliability.

For future work, although the experimental results are promising, many parts of the system framework can be re-configured and extended. Each step of our approach is flexible in term of choosing the method to solve. For example, generating classification rules from each data set at each time point can be done by other supervised machine learning techniques. In the rule pruning step, different dissimilarity measurement can be used to achieve different sets of rules. Or in the evolutionary trends predicting step, other regression techniques can be applied to compare with the existing results. We are also interested in applying our system to other databases which have the same properties to test its correctness and effectiveness.

ACKNOWLEDGEMENTS

Viet An's work has been supported by the Undergraduate Research Experience on Campus (URECA) programme from Nanyang Technological University.

REFERENCES

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and regression trees*. Chapman and Hall, New York, 368 p.

Cendrowska, J. (1987). Prism: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27(4):349–370.

Geng, L. and Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Survey*, 38(3):9.

Hetland, M. and Saetrom, P. (2002). Temporal rule discovery using genetic programming and specialized hardware. In *4th International Conference on Recent Advances in Soft Computing, RASC*.

Holte, R. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91.

Kadous, M. W. (1999). Learning comprehensible descriptions of multivariate time series. In *ICML*, pages 454–463.

Keogh, E. and Kasetty, S. (2002). On the need for time series data mining benchmarks: a survey and empirical demonstration. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 102–111, New York, NY, USA. ACM.

Kohavi, R. (1995). The power of decision tables. In *8th European Conference on Machine Learning*, pages 174–189. Springer.

Last, M., Kandel, A., and Bunke, H. (2004). *Data Mining in Time series databases*. Series in Machine Perception and Artificial Intelligence. World Scientific.

Last, M., Klein, Y., and Kandel, A. (2001). Knowledge discovery in time series databases. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 31(1):160–169.

Povinelli, P. (2000). Using genetic algorithms to find temporal patterns indicative of time series events. In *GECCO 2000 Workshop: Data Mining with Evolutionary Algorithsm*, pages 80–84.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc.

Su, J. and Zhang, H. (2006). A fast decision tree learning algorithm. In *Proceedings of The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference (AAAI/IAAI), July 16-20, 2006, Boston, Massachusetts, USA*.

Tan, P.-N., Kumar, V., and Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 32–41, New York, NY, USA. ACM.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley, Reading, MA.

Wah, B. W. and Qian, M. (2002). Constrained formulations and algorithms for stock-price predictions using recurrent FIR neural networks. In *Proceedings of The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference (AAAI/IAAI)*, pages 211–216, Edmonton, Alberta, Canada.

Wei, L., Kumar, N., Lolla, V. N., Keogh, E. J., Lonardi, S., Ratanamahatana, C. A., and Herle, H. V. (2005). A practical tool for visualizing and data mining medical time series. In *CBMS*, pages 341–346.

Weiss, G. and Hirsh, H. (1998). Learning to predict rare events in event sequences. In *4th Conference on Knowledge Discovery and Data Mining KDD*, pages 359–363, Menlo Park, CA, USA. AAAI Press.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition.