การเลือกตัวแทนของชุดข้อมูลด้วยวิธีโคไซน์และยูคลีเดียน

FEATURE SELECTION BASED ON COSINE AND EUCLIDEAN

อนิรุธ สืบสิงห์

ANIRUT SUEBSING

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

KMITL-2013-SC-D-002-005

การเลือกตัวแทนของชุดข้อมูลด้วยวิธีโคไซน์และยูคลีเดียน

FEATURE SELECTION BASED ON COSINE AND EUCLIDEAN

อนิรุธ สืบสิงห์

ANIRUT SUEBSING

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

KMITL - 2013 - SC - D - 002 - 005

FEATURE SELECTION BASED ON COSINE AND EUCLIDEAN

ANIRUT SUEBSING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2013

KMITL - 2013 - SC - D - 002 - 005

คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์	การเลือกตัวแทนของชุดข้อมูลด้วยวิธีโคไซน์และยูคลีเดียน Feature Selection based on Cosine and Euclidean
นักศึกษา	นายอนิรุธ สืบสิงห์
รหัสประจำตัว	49062952
ปริญญา	ปรัชญาดุษฎีบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.นวลสวาท หิรัญสกลวงศ์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	-

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ดร.อนันตพร	หรรษคุณาชัย	
ผศ.ดร.นันทิกา	เบญจเทพานันท์	
ผศ.ดร.กรกช	ประชุมรักษ์	
ดร.ชาคริต	วัชโรภาส	
ผศ.ดร.นวลสวาท	หิรัญสกลวงศ์	

วัน / เดือน/ ปี ที่สอบ 7 กุมภาพันธ์ พ.ศ. 2556 เวลา 16.00 – 19.00 น.
สถานที่สอบ ณ ห้อง 216 ชั้น 2 อาคารจุฬาภรณวลัยลักษณ์ 1

คณะวิทยาศาสตร์รับรองแล้ว

(รองศาสตราจารย์ ดร.ดุษณี ธนะบริพัฒน์)
คณบดีคณะวิทยาศาสตร์

วันที่ 1 **เดือน** มี.ค. **พ.ศ.** 56

หัวข้อวิทยานิพนธ์	การเลือกตัวแทนของชุดข้อมูลด้วยวิธีโคไซน์และยูคลีเดียน
นักศึกษา	นายอนิรุธ สืบสิงห์
รหัสประจำตัว	49062952
ปริญญา	ปริญญาดุษฎีบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2556
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผศ. ดร.นวลสวาท หิรัญสกลวงศ์

บทคัดย่อ

ปัจจุบันนี้การพัฒนาของระบบคอมพิวเตอร์และวิศวกรรมได้เปลี่ยนแปลงไปอย่างมาก ทำให้สามารถจัดเก็บข้อมูลจำนวนมากถูกจัดเก็บไว้ในรูปของดิจิตอล ทำให้นักวิทยาศาสตร์หาวิธีการที่จะแปลงข้อมูลเหล่านั้นไปเป็นองค์ความรู้ที่สามารถนำมาใช้ประโยชน์ได้ ซึ่งแนวคิดนี้รู้จักกันในนาม การค้นคืนองค์ความรู้ และ การทำเมืองข้อมูล แต่เนื่องจากเทคนิคทางด้านการทำเมืองข้อมูล มีอยู่ด้วยกันหลายเทคนิค และถ้าพูดถึงเทคนิคที่เรียกว่า การจำแนกข้อมูล ก็ต้องให้ความสำคัญกับขั้นตอนวิธีการเลือกตัวแทนของชุดข้อมูลโดยอาศัยวิธีการแมชชีนเลิร์นนิ่ง ซึ่งเป็นเทคนิคที่ซับซ้อนและใช้เวลาในการประมวลผลมาก ดังนั้นในงานวิจัยนี้จึงมีจุดประสงค์ที่จะทำการพัฒนาวิธีการที่ไม่ซับซ้อนเพื่อช่วยคัดเลือกตัวแทนชุดข้อมูลที่เหมาะสมสำหรับนำไปใช้ในการสร้างโมเดลทำนายผล และวิธีการที่ว่าก็คือ EU-COSSIM ถูกพัฒนาและถูกนำเสนอเพื่อช่วยแก้ปัญหาความซับซ้อนและล่าช้าที่เกิดขึ้นกับวิธีการแมชชีนเลิร์นนิ่ง วิธี EU-COSSIM เป็นวิธีการที่อาศัยขั้นตอนและวิธีการหาค่าความคล้ายด้วยวิธีโคไซน์และการหาค่าความต่างด้วยวิธียูคลีเดียน โดยใช้ C5.0, SVM และ RIPPER ประมวลผลกับข้อมูลจำนวน 6 ชุด เป็นตัววัดประสิทธิภาพของวิธีการที่นำเสนอ จากผลการทดลอง แสดงให้เห็นว่าวิธีการที่นำเสนอนี้ช่วยเพิ่มประสิทธิภาพในการทำนายผล อีกทั้งยังเป็นขั้นตอนที่มีความเรียบง่าย รวดเร็ว ไม่ซับซ้อน

คำสำคัญ: วิธีการเลือกตัวแทนของชุดข้อมูล, การจำแนกข้อมูล, ค่าความต่างด้วยวิธียูคลีเดียน, ความคล้ายด้วยวิธีโคไซน์, การค้นคืนองค์ความรู้, แมชชีนเลิร์นนิ่ง

Thesis Title	Feature Selection based on Cosine and Euclidean
Student	Mr. Anirut Suebsing
Student ID.	49062952
Degree	Doctor of Philosophy
Programme	Computer Science
Year	2013
Thesis Advisor	Asst. Prof. Dr. Nualsawat Hiransakolwong

ABSTRACT

In recent times, the rapid developments in computer science and engineering have led to expediency and efficiency in capturing huge accumulations of data. The new challenge is to transform the enormous of data into useful knowledge for practical applications known as knowledge discovery and data mining. At the present time, the growing importance of knowledge discovery and data mining has made the feature selection to play an important role because choosing a robust subset of the features may increase accuracy and reduce time complexity of the knowledge discovery and data mining. Furthermore, normally processing huge amount of collected data to identify pattern needs high computation cost and large storage space. Therefore, with the feature selection, the computation cost and storage space can be reduced by removing irrelevant and possibly redundant features. In the previous researches on feature selection, the criteria and algorithms for selecting the features from the raw data are mostly complicated and difficult to implement because of its base the machine learning techniques. Thus, this dissertation proposes a new feature subset selection approach by using EU-COSSIM (algorithm based on Euclidean distance and cosine similarity). This method is the simple algorithm using smaller storage space, reducing computation time and gaining higher prediction performance for classification. During the evaluation phase, six different benchmark data sets from UCI are used to evaluate the performance of the proposed

approach comparing with previous works and especially entire features of each benchmark data sets. Moreover, the popular classification algorithms—C5.0, SVM, and RIPPER are used for building predictive models to measure an efficiency of the proposed approach in this paper. Experimental results show that a novel method, the EU-COSSIM, can select a robust subset of features to improve the performance of a predictive model based on the C5.0, SVM, and RIPPER classifiers.

Keywords: Feature Selection, Classification algorithms, Euclidean distance, Cosine similarity, knowledge discovery, Machine learning.

ACKNOWLEDGEMENTS

The author would like to thank my family for giving opportunities, pushing me, understanding me, and helping me to achieve my Ph.D. Also, a warm thank to my parents for encouraging and fulfilling financial support during the entire period of study.

I would like to express my deeply many thanks to my thesis advisor, Asst. Prof. Dr. Nualsawat Hiransakolwong, whose all advises and very good support from the initial to the final level enabled me to develop an understanding of the research.

The authors also gratefully acknowledge the helpful comments and suggestions of the committee, which have improved the presentation.

I would like to thank Office of Academic Administration of King Mongkut's Institute of Technology Ladkrabang, Ubon Ratchathani University, Thai Higher Education Commission for supporting my Ph.D. study. Moreover, the authors are thankful to Venerable master monk Pet from Wat Prayongkiti Vanaram whom gave me scholarship.

Lastly I would like to thank my friends supporting and encouraging all aspects e.g., about programming, stationery.

Anirut Suebsing

TABLE OF CONTENTS

LIST OF TABLES

Tables **Page**

LIST OF FIGURES

Figures **Page**

CHAPTER 1

INTRODUCTION

1.1 Statements of Problem

The rapid developments in computer science and engineering have led to expediency and efficiency in capturing huge accumulations of data. The new challenge is how to exploit the ocean of data into useful knowledge for practical applications. Currently, data mining plays an important role in this issue. Data mining is composed of many tasks but one of the essential steps of data mining is feature selection. Feature selection is a method where only the core features are selected by discarding weak features. Minimum set of features, which is close enough to represent the original dataset, will be selected. Feature selection allows reducing the number of features, and also removing irrelevant, redundant and noisy features. The feature selection with the smallest dataset can reduce the storage space and time complexity. This allows for building simpler and more comprehensible classification models with enhancing classification performance. Selecting relevant attributes is a critical issue for competitive classifiers and for data reduction [1]. Automatic methods for selection a subset of features are often developed for searching an appropriate subset of containing relevant features because the possible number of feature subsets is $2^N - 1$ subsets for N features [5]. Therefore, it is impossible to search for the robust feature subset manually even after cleaning the data. Moreover, there are many reasons for using feature selection concluded as follows [1]-[5]:

- Getting the maximizing accuracy of the classifier
- Removing irrelevant or noise and redundant features
- Reducing the time complexity, computational cost and storage space

Algorithms for feature selection typically fall into two categories [1]-[7]; filter and wrapper approach. Filter approach filters irrelevant features out but keeping a good feature set before learning process [2]-[6]. On the other hand, wrapper approach searches for a good feature set using a learning algorithm. Utilizing filter approach to generate a feature set is generally faster than wrapper approach because filter approach uses heuristics based on general characteristics of

the data rather than wrapping a learning algorithm into the selection process to evaluate the merit of feature subsets [2, 3, 4, 5, 7].

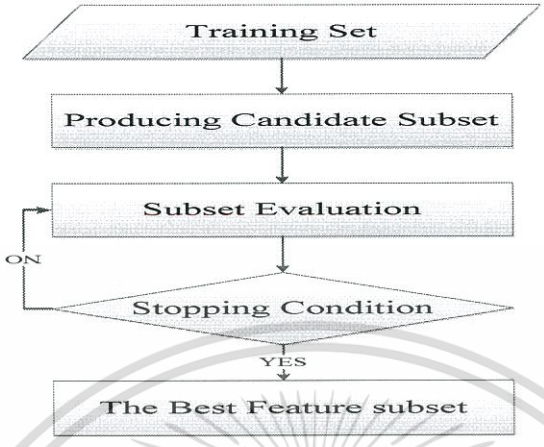


Figure 1.1 The process of feature selection.

Although the previous researches on feature selection, the criteria and way about how to select the features from the raw data are difficult to implement. Therefore, the EU-COSSIM algorithm was proposed in this thesis based on Euclidean Distance and Cosine Similarity to improve the efficiency of the feature selection method. The main objective of this paper is to develop a good method to select or extract significant features used to build a prediction model with more accuracy. The EU-COSSIM is less complex than other techniques, especially machine learning techniques, because those techniques are more complicated and more difficult to implement into real-world application as well. The benefits of the research are as follows:

- A novel method based on the Euclidean Distance and cosine similarity can improve the performance of a prediction rate with less false positive
- The proposed approach can select robust features providing higher performances than other previous techniques, and entire features.
- The proposed technique is a less complicated method with reducing storage space and computation costs.

For evaluating a proposed approach in this thesis, the C5.0, SVM, and RIPPER algorithms [8] are used for evaluating this technique that enhances prediction on UCI benchmark data sets [9] correctly, using smaller storage space and less computational cost. However, this thesis is focused on KDD 1999 cup – Computer network intrusion detection because this data set is selected as competition data set in 1999. It is reliable with dividing data as a training set and a

test set by data miners of KDD cup. Moreover, the data set is still attractive and challenge for researchers because the internet and local area networks are growing larger nowadays. People all over the world are connecting to the internet; they are unconsciously encountering the number of security threats such as viruses, worms and attacks from hackers [10]. Now firewalls, anti-virus software, message encryption, secured network protocols, password protection and so on are not sufficient to assure the security in computer networks, which some intrusions take the opportunity from weakness in computer system to threaten. Therefore, intrusion detection becomes more and more important technology which follows up network traffic and identifies network intrusion such as anomalous network behaviors, unauthorized network access, and malicious attacks to computer systems [11] - [13].

1.2 Research Objectives

The main objective of this thesis is to develop a good method to select or extract significant features used to build a prediction model. The EU-COSSIM is less complex than the other techniques, especially machine learning techniques, since those techniques are more complicated and more difficult to understand and to implement into the real-world application as well.

1.3 Scope of Thesis

The following scopes of the study are:

1. The proposed method applies Euclidean Distance and Cosine Similarity to create a robust feature subset.

2. The technique uses smaller storage space, uses less complicated method but getting higher prediction performance and avoiding high computational costs.

3. The proposed approach can improve the efficiency of prediction rate of UCI benchmark data sets.

1.4 Results

The benefits of the research are as follows:

1. A novel method based on the Euclidean Distance and cosine similarity can improve the performance of a prediction rate with less false positive especially

2. Although the approach can select a small robust feature subset, it still provides higher performances in contrast with other previous techniques; especially entire features.

3. The proposed technique introduced is a less complicated method so it uses smaller storage space and avoids high computational costs.

4. Since the proposed method is able to work independently, it can apply to column-block partitioning on parallel computing system easily without changing it.

1.5 Research Methodology

Feature selection is a method where only the relevant features will be selected. Feature selection algorithms typically fall into two categories; filter and wrapper approach. Filter approach filters irrelevant features out keeping a good feature set before learning process. In this research, a proposed approach refers to this algorithm. The approach is used to create a new feature selection method by applying Euclidean Distance and Cosine Similarity then adding with a new filtering feature technique for extracting a robust feature subset. Then, the robust feature subset is employed to build a model based on C5.0 for evaluating this technique that can enhance to predict UCI benchmark and KDD cup data sets correctly.

1.6 Organization of Thesis

The remaining chapters of this thesis are organized in the following way.

Chapter 2 is addressed a brief review of related work as follows: Feature Selection Algorithms, Literature Review, C5.0, SVM, and RIPPER Algorithms, and Benchmark Data Sets, Cosine Similarity, Euclidean Distance.

Chapter 3 presents a proposed approach used to select a suitable features subset.

Chapter 4 contains experimental evaluation following with the discussion on the results.

Chapter 5 presents conclusion and recommendation.

CHAPTER 2

LITERATURE REVIEWS

This chapter provides brief literature reviews as follows: Feature Selection Algorithms, Related Works, Evaluation Algorithms, and Benchmark Data Sets, Cosine Similarity, Euclidean Distance.

2.1 Feature Selection Algorithms

• The Filter Approach: the Filter algorithms [2, 3, 6] usually based on statistics consider the relevant features used in the classification. Statistics and Information theoretic measures such as information gain, Cross-entropy, Pearson's Chi-Square and so on are used to find the relationship of each feature in a data set with the target feature or class label assuming conditional independence with all other features. The robust subset of features selected from high rank features. Ranking features are ordered according to values of evaluation measures, such as accuracy, consistency, information, distance, and relevance. Figure 2.1 shows the Filter approach flowchart.

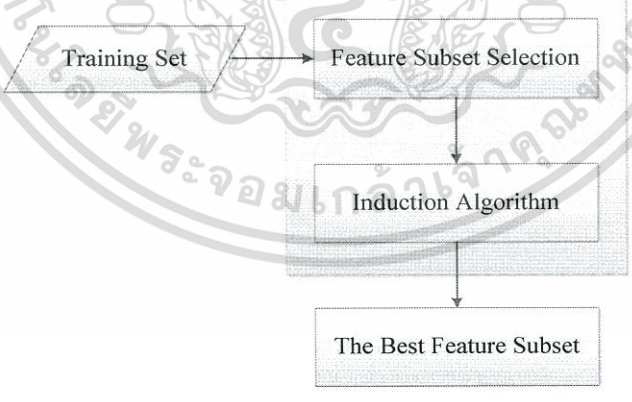


Figure 2.1 The filter approach flowchart.

• The Wrapper Approach: Machine learning algorithms [2, 3, 7] play an important role in this approach as evaluation functions. The Wrapper algorithms usually provide better accuracy with more complexity using higher computation cost. These algorithms typically start from an empty list of features and then add discovered relevant features. The wrapper approach flowchart is shown in Figure 2.2.

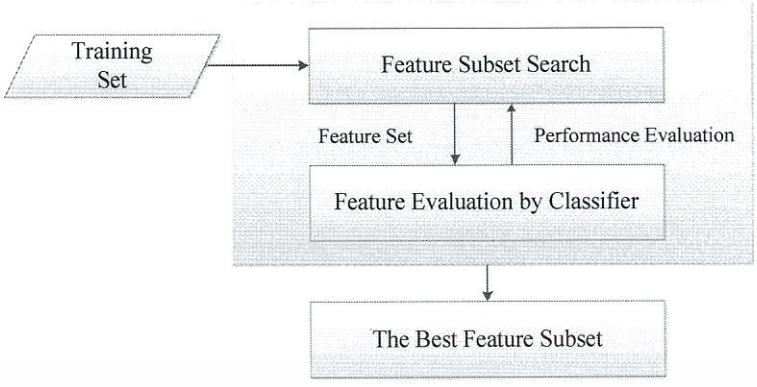


Figure 2.2 The wrapper approach flowchart

2.2 Related Works

In NIDSs (Network Intrusion Detection Systems), many researchers use a lot of techniques to gain more accuracy. The most popular techniques use data mining and machine learning [14].

"Agent-based Network Intrusion Detection System using Data Mining" was proposed by Cheung-Leung Lui and et al [15]. They proposed an adaptive NIDS by developing data mining technology, which accurately captured the actual behavior of network traffic. The proposed NIDS was constructed by a number of different types of agents which each agent was based on data mining techniques consisted of clustering, association rules and sequential association rules. Therefore, three data mining techniques were primary components in the adaptive NIDS. Clustering approaches were used to extract properties from traffic in terms of frames and try to make the normal traffic from isolated clusters. Then, each cluster had its representative feature vectors representing certain normal property. For an unknown traffic to be clustered, its traffic property with those trained clusters was compared. Finally, if the unknown traffic vector had distance too further away from normal clusters, it was classified as attack traffic. They called the agent using clustering techniques as "Clustering-based agent". In clustering technique, they chose k-means method using Euclidean distance to cluster the traffic because k-means clustering needs lower dimensional space and the computational complexity increases exponentially with increasing of number of features while association rules approach was employed to find out relationship between selected features and traffic property with a condition set into four selected features as follows: 1) If the number of unique accessed ports was larger than a threshold, it was

declared as attack traffic, 2) Traffic was announced as normal if the traffic in frame was smaller than average packet size, 3) If the number of packet in frame was larger than a threshold, it was responded as a victim, 4) Traffic was normal if time range covered by packets should show a burst in short time. The last agent was based on sequential association rules. Besides capturing the general behaviors of normal network traffic mentioned above, there are some common sequential patterns at connection level of a normal traffic and different sequential patterns in attack traffic. Hence, sequential rules-based agent was utilized to extract pattern rules for differentiating the normal traffic and intrusion. The method declared any traffic as intrusion when the number of abnormal connections matched within the packet/time frame was larger than a threshold, MIN_ATTACK. However, the adaptive NIDS would make any traffic as normal by using overlapped area between clustering based and rule based agents (association rule and sequential rule based agents). Thence, the adaptive NIDS was able to eliminated false positive (FP). They used the 1998 DARPA Intrusion Detection Evaluation program by MIT Lincoln Labs to evaluate their proposed approach. Their experimental results showed high detection rate with less false positive rate. Nevertheless, their proposed approach has disadvantages of its parameter setting with many thresholds, such as, the k-means clustering approach setting k=256, the rule-based agent setting its minimum support as 100% and depreciation percentage (depr) as 96%, etc. Others were shown in their section 4.2 in [15]. Moreover, since their proposed approach is consisted of many processes, it takes time before getting the final result.

"Modeling Intrusion Detection System by Discovering Association Rule in Rough Set Theory Framework" was proposed by Wang Xuren and et al. [16]. They studied and researched about the main shortcomings of common or commercial IDSs found that the primary defects were misuse detection, one of two main categories of IDS. It could not predict future attacks and had high false-alarm rate. Attack signatures were generated manually and updated difficultly. However, they found applying intelligent technology and soft computing, such as transition analysis, Statistical approaches, expert system, Model-Based approach, Pattern Matching, Artificial Neural Network, Support Vector Machines, Neuro-Fuzzy, Multivariate Adaptive Regression Splines, Linear Genetic Programming and hybrid system based on data mining to IDS can accomplish problems. Nevertheless, they improved association rules discovering system under rough set theory, a tool to deal with inexact, uncertain or vague knowledge framework [16]. They used KDD CUP 1999 data set, an originally benchmark, provided by MIT Lincoln Labs to evaluate their proposed approach. Their system achieves classification more accuracies. This

proposed approach was first used to reduce the raw data set in order to produce ultra data set. They selected 31 quantitative attributes from the data set of 41 attributes. Next, the selected 31 quantitative attributes was reduced as a set of continuous values was partitioned into a finite number of categories, commonly the rough set community. The data-preprocessing unit discretized automatically the numerical attributes by a discretization algorithm, Semi-naïve algorithm—which has more logics to handle value-neighboring objects belonging to different decision classes [16]. In the last process, classical Apriori algorithm for association rules is as follows. All item sets that have support greater than or equal to the user specified minimum support are generated. All the rules coming from frequent item sets that have minimum confidence are generated. According to the definition of association rules by rough set theory, only the rules whose right hand sides belong to classification attribute are finally selected. Those selected rules can be applied to classifying new connection data, called association classifying rules. The rules built by this proposed approach were brought to evaluate detection accuracy in experimental section. In experimental section, their proposed approach with different minimum supports 0.01, 0.005 and 0.0005 respectively was able to show high detection accuracy at 99.50% of normal, 96.39% of dos and 50.77% of Probing respectively (with minimum support 0.01) while U2R and R2L were not able to be detected correctly because maybe there are too few U2R data and R2L data in the data set of KDD CUP 1999. Moreover, the detection accuracy of each class getting from their experiment depended on minimum supports. The minimum support 0.005 gave the best accuracy of classifications in overall. However, the experimental result of their proposed approach was able to show quite satisfied detection accuracy. Their proposed approach is a complicated method because of based on rough set theory. Moreover, the rules count on the minimum support whether association rule can give robust rules. It is hard to approximate what is the best appropriate minimum support for each data set based on association rule. Furthermore, in detection accuracy of Probing class, there is a chance to increase more detection accuracy than their report.

"Network Intrusion Detection through Genetic Feature Selection" was proposed by Chi-Hoon Lee and et al. [17]. They presented a new feature selection method that maximizes class between normal and attack patterns on network connections. In this thesis, they focus on selecting a robust feature subset based on the genetic optimization procedure to improve a true positive intrusion detection rate. From their study, they found that evolutionary algorithms were generally effective for rapid global assessment of a large search space in multimodal optimization problems

such as the feature selection. They discovered that a seminal work of Siedlecki and Sklansky [18, 19] demonstrated the effectiveness of the GA-based feature selection, reducing the time for finding near-optimal feature subsets. However, they presented the critical issue from GA-based. The problem is how to design an appropriate fitness function to avoid local mina. Therefore, they designed method to solve the problem by using the concept of a *k* nearest neighbor classification rule. This rule considers the number of neighbors that are not used in the majority decision. In experimental investigation section, they used KDD CUP 1999 data set originally provided by MIT Lincoln Labs like other researchers to evaluate their method. Furthermore, the results from their proposed approach with 21 selected features were compared with the results from full attributes (41 attributes) excluding class label. From the experimental results, the selected features from their proposed approach indicated more efficiency of predictive accuracy than entire features. The selected features were 51.22% of entire features. They concluded their proposed approach applied GA was able to select the robust features in NIDS. GA has more parameters that need to be set, such as crossover, mutation, population size and so on. Moreover, definition of fitness function in problem domain is a major factor that GA achieves any goals. Therefore, how to set up parameter values properly is a problem. The GA approach is a time consuming method also.

"A Feature Approach for Network Intrusion Detection" was proposed by Kok-Chine Khor and et al. [20]. They proposed an approach for obtaining optimal number of features to build an efficient model for intrusion detection system (IDS). From their study in IDSs, they found most other proposed methods for solving shortcomings in IDS about misuse and anomaly techniques. The misuse detection has defects in detecting new intrusions as the patterns are not known by IDS. The anomaly detection was proposed to solve the problem occurring in misuse; however, this technique causes IDSs suffer from high false alarm rates (FP) [27]. Therefore, with the problems of IDSs, network security experts play important role in IDSs. Nevertheless, since the amount of network data to be examined by IDSs is too huge and composes of various activities in computer network. Identifying intrusions entirely based on human eyes is extremely difficult. Therefore, data mining and intelligence techniques are proposed to alleviate the problems. However, data mining and intelligence techniques will use high computational costs in processing tasks if the number of features involved in network data increases. Therefore, they proposed an approach for reducing features in the collected data to obtain optimal number of features to build an efficient model for intrusion detection system. However, to acquire a set of

robust features and achieve high accuracy of intrusion detection while keeping the minimum number of features, some very important features were filtered by domain experts for adding into optimal features selected by their method. The method was designed by using filter approach—one of two basic approaches in selecting a good feature subset, because it is faster than wrapper approach [9]. Moreover, Classification and Regression Tress (CART) was used to evaluate their features selected by their proposed approach. Furthermore, the 1998 DARPA intrusion detection dataset was the primary dataset for their empirical study conducted in their research work. In their experiment, their approach for determining an optimal feature set by using feature selection method with intervention from domain experts showed the results that computational cost of processing network data was reduced still preserving the classification accuracies. They summarized that the feature set provided by domain expert was a chief factor leading to achieve high accuracy with only 7 selected features. However, as long as their proposed approach still depends on domain experts. This still has a handicap because when patterns of intrusions are changed, without domain experts, the approach maybe cannot predict correctly. The limitation for a major obstacle is that their approach cannot adapt without human when it is applied to use in the real world or commercial IDSs.

"Euclidean-based Feature Selection for Network Intrusion Detection" was proposed by Anirut Suebsing and Nualsawat Hiransakolwong [21]. Euclidean Distance was used for selecting a subset of robust features with using smaller storage space and getting higher intrusion detection performance. In this proposed approach, the Euclidean distance was used to compute ranking score between each attribute and class label by defining each attribute of KDD Cup 1999 training set, 41 attributes, represented as A_1, A_2, A_3,...,A_{41} respectively and class label represented as B; moreover, let x is a value in any attribute and y is a values in class label.

Let any $A_j = \{x_{1,j}, x_{2,j}, x_{3,j}, ..., x_{m,j}\}$ be a vector of attribute j, where j ($1 \le j \le 41$) is an ordinal number of attributes of training set, and also m ($m \ge 0$) is the number of instances of training set.

Let $B = \{y_1, y_2, y_3, ..., y_m\}$ be a vector of class label, where m ($m \ge 0$) is the number of instances of training set.

Thus, the ranking score is $\{d_1(A_1,B), d_2(A_2,B), d_3(A_3,B), ..., d_{41}(A_{41},B)\}$,

$$d_j(A_j,B) = \sqrt{\sum_{i=1}^{m}(x_{i,j} - y_i)^2} \qquad (1)$$

Where j ($1 \le j \le 41$) is an ordinal number of attributes of training set, and also m ($m \ge 0$) is the number of instances of training set.

After computing distance measure, then getting the ranking score of known detection method of each attribute, $\{d(A_1,B), d(A_2,B), d(A_3,B), \ldots d(A_{41},B)\}$. Scores of the ranking score are sorted from highest to lowest.

Then, features whose scores are higher than a threshold are selected to build a model. This model was used to detect accurately for known and unknown attacks.

Finally, the method of C4.5 was used to evaluate selected features.

	Attributes					Class Label
Instances	A_1	A_2	A_3	...	A_{41}	B
	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,41}$	y_1
	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{2,41}$	y_2
	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,41}$	y_3
	⋮	⋮	⋮	⋮	⋮	⋮
	$x_{m,1}$	$x_{m,2}$	$x_{m,3}$	...	$x_{m,41}$	y_m

Figure 2.3 Vectors of each attribute and a vector of class label.

"Feature Selection Using Euclidean Distance and Cosine Similarity for Intrusion Detection Model" was proposed by Anirut Suebsing and Nualsawat Hiransakolwong [22]. Euclidean distance was used for selecting a subset of robust features to build model for the detection of "known attacks", while Cosine similarity was used for selecting a subset of robust features to build model for the detection of "unknown attacks" for getting higher intrusion detection performance. Cosine similarity was used to compute ranking score between each attribute and class label by representing each attribute of KDD cup 1999 training set as A_1, A_2, A_3,..., A_{41} respectively and class label as B; moreover, let x is a value in any attribute and y is a values in class label.

Let any $A_j = \{x_{1,j}, x_{2,j}, x_{3,j}, \ldots, x_{m,j}\}$ be a vector of attribute j, where j ($1 \le j \le 41$)is an ordinal number of attributes of training set, and also m ($m \ge 0$) is the number of instances of training set.

Let $B = \{y_1, y_2 y_3, \ldots, y_m\}$ be a vector of class label, where m ($m \ge 0$) is the number of instances of training set.

Then, the ranking score of unknown detection method is $\{Sim_1(A_1,B),\ Sim_2(A_2,B),\ Sim_3(A_3,B),\ ...,\ Sim_{41}(A_{41},B)\}$,

$$\mathrm{Sim}_j(A_j,B) = \frac{\sum_{i=1}^{m}(\mathbf{x}_{i,j} \bullet y_i)}{\sqrt{\sum_{i=1}^{m}\mathbf{x}_{i,j}^2 \bullet \sum_{i=1}^{m} y_i^2}} \qquad (2)$$

Where j $(1 \le j \le 41)$ is an ordinal number of attributes of training set, and also m $(m \ge 0)$ is the number of instances of training set. After computing similarity measure, the scores of unknown detection method of each attribute is represented as *a set* $\{Sim_1(A_1,B),\ Sim_2(A_2,B),\ Sim_3(A_3,B),\ ...,\ Sim_{41}(A_{41},B)\}$.

Then, scores are arranged from highest to lowest. Finally, chose features whose scores are higher than a threshold to build a model.

Note that the Euclidean distance proposed in [21] was used for selecting a subset of features to build a model for the detection of the known and unknown attacks or patterns while the Euclidean distance proposed in [22] was for selecting a subset of features to build a model for the detection of the known attacks or patterns only but for the detection of unknown attacks or patterns, the model was built from a subset of features provided by Cosine similarity instead. However, a technique to select a subset of features by using Euclidean distance proposed in [21] and [22], or by using Cosine similarity is the same.

The last two approaches are our previous works shown impressive results with easier to implement. However, our works have drawbacks as follows: (1) since our previous methods proposed in [21, 22] select a subset of features from ranking score getting from computing values of distance between each attribute and a class label by using equation (1) and equation (2), these methods need the class label. (2) selected features depend on a threshold parameter for selecting the final feature subset (choosing features whose scores are higher than a threshold to build a model) because the criteria of threshold was defined that it could separate between high scores for selected features and low scores for unselected features while the proposed method in this thesis does not rely on the class label, and without using any threshold parameters for selecting a feature subset. The proposed method is described more clearly in Section 3.

Even though all of previous works show the efficiency of each works, those works have the disadvantage to rely on threshold parameter, unable to set up thresholds automatically or need a domain expert. Moreover, many previous papers in field of IDSs presented solution by using mostly machine learning techniques through data mining, such as neural network, fuzzy logic,

genetic algorithm and so on. However, those techniques are so complex and difficult to implement. Furthermore, those techniques are time consuming.

Therefore, a novel approach called EU-COSSIM is presented to solve all of the above problems. The EU-COSSIM processes without using threshold parameter, and more easy to implement with less processing time. This approach can select more robust features to improve the performance of a predictive model. Moreover, the proposed method EU-COSSIM can apply to select not only an intrusion data set but also other data sets.

2.3 Evaluation Algorithms

In this thesis, C5.0, SVM, and RIPPER algorithms are used to evaluate features in this thesis since these algorithms are widely used in the classifier.

(A) C5.0 Algorithm: Classification [8] is one of the most popular data mining techniques. It is a process of learning a function mapping a data item into one of some predefined classes. A decision tree form is most useful in classification problems [8]. With this method, a decision tree is built to model the classification process. Well-known tree algorithms used widely are ID3, CART (Classification and Regression Tree) and C4.5 algorithms while the C5.0 algorithm is a commercial version extended from C4.5 proposed by J.R. Quinlan [23]. Now it is widely used as the inductive learning tools in Clementine, Rule Quest and so on. The C5.0 algorithm is based on the information theory [8, 23]. Decision trees are built by calculating the information gain ratio. The C5.0 algorithm works by separating the sample into subsamples based on the result of a test on the value of a single feature. The specific test is selected by an information theoretic heuristic.

This procedure is iterated on each of the new subsample and keeps on until a subsample cannot be separated or the partitioning tree has reached the threshold. The information gain ratio is defined as:

$$\text{Information Gain Ratio}\,(\mathrm{D},\ \mathrm{S}) = \frac{Gain(D,S)}{H\left(\frac{|D_1|}{D},\ldots,\frac{|D_S|}{D}\right)} \qquad (3)$$

Where D is a database state, H $(\cdot)$ finds the amount of order in that state. The state is separated into new states $\mathrm{S} = \{\mathrm{D}_1,\ \mathrm{D}_2,\ldots,\ \mathrm{D}_S\}$.

The algorithm of C5.0 is very robust for handling missing data and in a large number of input fields [8, 23].

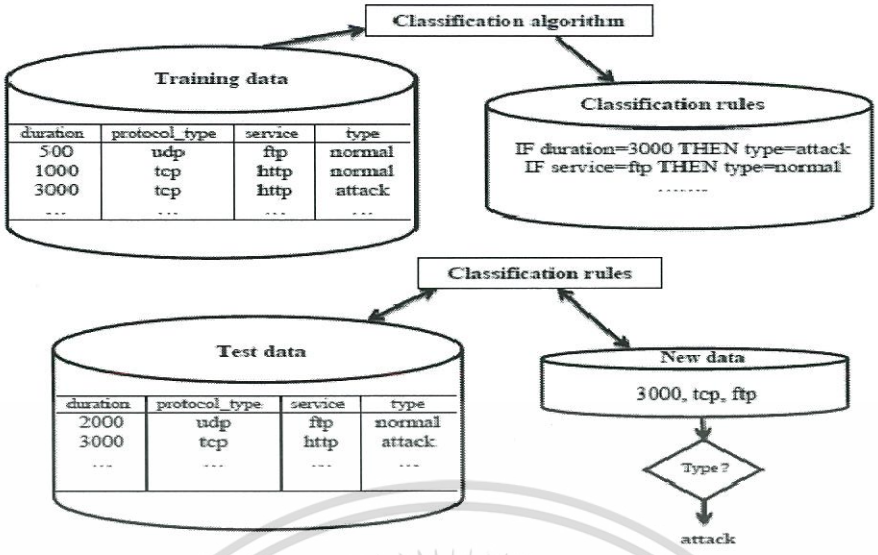


Figure 2.4 The data classification process.

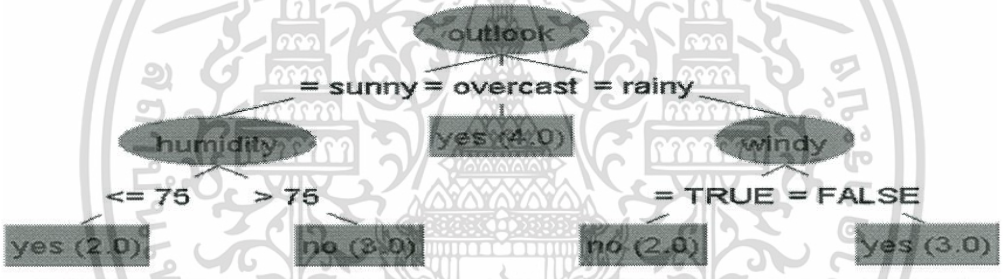


Figure 2.5 Basis structure of C5.0.

(B) SVM Algorithm [24]: SVM (Support Vector Machine) learning algorithms were first proposed by Vapnik. It is a novel learning method based on the statistical learning theory. It can successfully solve the problem of 'over fitting', local optimal solution and low-convergence rate. The primary idea of SVM is using a high dimension space to find a hyper plane to do binary division, where the achieved error rate is minimum. An SVM can handle the problem of linear in separability. An SVM uses a portion of the data to train the system and finds several support vectors that represent training data. These support vectors will be formed into a model by the SVM, representing a category. In classifier, SVM is adopted widely because it can provide high classification accuracy.

(C) RIPPER Algorithm [24]: RIPPER (Repeated Incremental Pruning to Produce Error Reduction), was proposed by William W. Cohen. It consists of two main stages: the first stage, it

constructs an initial rule set using a rule induction algorithm called IREP; the second stage further optimizes the rule set initially obtained. These stages are repeated for k times. IREP is called inside RIPPER-k for k times. For each iteration, the current dataset is randomly partitioned in two subsets: a growing set, that usually consists of 2/3 of the examples and a pruning set, consisting in the remaining 1/3. These subsets are used for two different purposes: the growing set is used for the initial rule construction (the rule at growth phase) and the pruning set is used for the pruning (the rule in pruning phase). MDL (Minimal Description Length) is used as a criterion for stopping the process in IREP.

2.4 Benchmark Data Sets

To evaluate the proposed approach, the six benchmark data sets from the UCI repository [9]—KDD 1999 cup, KDD 2004 cup, KDD 2008 cup, CoverType, Zoo and Arcene are used In this thesis. However, the KDD 1999 cup–Computer network intrusion detection, is used to compare with other four methods because it is still popular, and researchers provided their subset of features. While the rest of data sets are used to evaluate the proposed approach compared with the ground truth, the entire features.

(A) KDD Cup 1999 data set: the KDD Cup 1999 data set is used to study and evaluate research in intrusion detection in terms of unauthorized usage, denial of service, and anomalous behavior. The data set is the real data which is captured in the real network. It includes many kinds of attack data, also including the normal data (Stolfo et al. distinguishing normal connections from attacks [25]). The raw data was processed in to 22 known attack and 17 unknown attack types. These attacks are divided into four categories: DoS (denial-of-service, e.g., SYN flood), probing (surveillance and other probing, e.g., port scanning), U2R (unauthorized access from a user to root privilege, e.g., various "buffer overflow" attacks) and R2L (unauthorized access from remote to local machine, e.g., guessing password). For each TCP/IP connection, 41 input features plus one class label. This domain expert is extracted in the data set belonging to four kinds [26]-[28] as following:

• Basic Features: Basic features can be derived from packet headers without inspecting the payload.

• Content Features: Content features are features that look for suspicious behavior in the data portions—suggested by domain knowledge, such as the number of failed login attempts.

• Time-based Traffic Features: These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval.

• Host-based Traffic Features: Utilize a historical window estimated over the number of connections – 100 patterns are used– instead of time. Host based features are therefore designed to assess attacks, which span intervals longer than 2 seconds.

However, this study focuses on known attack types because in real world, known patterns always occur in network systems; moreover, most previous researchers were interested only in the known patterns.

Therefore, this data set consists of three components as follows: only "10% KDD Cup" data set is utilized for the purpose of training set. This data set is composed of 22 attack types. The data set was randomly divided from version of the "100% KDD Cup" dataset. It contains more examples of attacks than normal connections and the attack types are not represented equally. Because of their nature, denial of service attacks account for the majority of the dataset. On the other hand, the "Corrected KDD Cup" data set provides a dataset with different statistical distributions than either "10% KDD Cup" or "100% KDD Cup" and contains 17 additional attacks. Therefore, the test data set is not from the same probability distribution as the training data set, and it includes specific attack types not in the training data. This makes the task more realistic.

Training data set (10% KDD Cup) In this thesis contains 49,451 records, which are randomly generated from the KDD Cup 1999 for 10% training data set that consists of 9,768 normal patterns, 39,085 known DoS patterns, 435 known Probe patterns, 111 known R2L patterns and 52 known U2R patterns.

Test data set in this thesis composes of three different test data sets, which are randomly selected from the test data set of KDD Cup (corrected KDD Cup), 100% test data set. Table 2.1 gives the number of records on three different test data sets

Table 2.1 The number of records on three different test data sets.

Data Set Name	No. Record
D1	186,745
D2	49,438
D3	25,419

(B) The other UCI benchmark data sets: To increase reliability of evaluating the proposed approach, the other UCI benchmark data sets in the Table 2.2 were used too.

Table 2.2 The details of other UCI benchmark data sets.

Data Set	No. Instances	No. Attributes	Data Set Description
KDD 2004 cup—Bioinformatics	145,751	77	The goal is to predict which proteins are homologous to a native sequence.
KDD 2008 cup—Breast cancer from X-ray images	102,294	118	It focuses on the problem of early detection of breast cancer from X-ray images of the breast.
CoverType	581,012	54	Predicting forest cover type from cartographic variables only.
Zoo	101	18	Here is a breakdown of which animals are in which type.
Arcene	100	10,001	It is to distinguish cancer versus normal patterns from mass-spectrometric data.

2.4.1 K-Fold Cross-Validation

However, since data sets in Table 2.2 were not divided into a training data set and testing data set clearly, k-fold cross validation is performed to validate classification accuracy. Cross-Validation [29, 30] is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model. In typical cross-validation, the training and validation sets must cross-over in successive rounds such that each data point has a chance of being validated against. The basic form of cross-validation is k-fold cross-validation. Other forms of cross-validation are special cases of k-fold cross-validation or involve repeated rounds of k-fold cross-validation. In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k-1 folds are used for learning. Figure 2.6 shows division of data set for cross-validation. In data mining and machine learning 10-fold cross-validation (k = 10) is the most common. Cross-validation is used

to evaluate or compare learning algorithms as follows: in each iteration, one or more learning algorithms use k-1 folds of data to learn one or more models, and subsequently the learned models are asked to make predictions about the data in the validation fold.

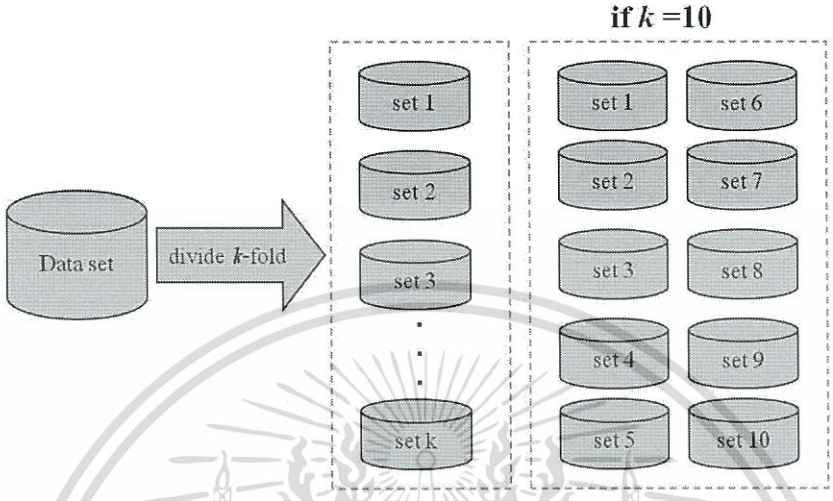


Figure 2.6 k-fold cross-validation for each data set

The goals of the cross-validation are (1) to estimate performance of the learned model from available data using one algorithm. In other words, to gauge the generalizability of an algorithm, and (2) to compare the performance of two or more different algorithms and find out the best algorithm for the available data, or alternatively to compare the performance of two or more variants of a parameterized model.

In this thesis, each data set in Table 2.2 is divided into ten folds as Figure. 2.4. Afterwards, divided data set are crossed-over to reduce bias of data for testing classification accuracy. For example, the data set is divided into 20% as training set and the rest (80%) for test set shown in Figure 2.7. The classification accuracy of each training data set derives from the average of classification accuracy from all test data set.

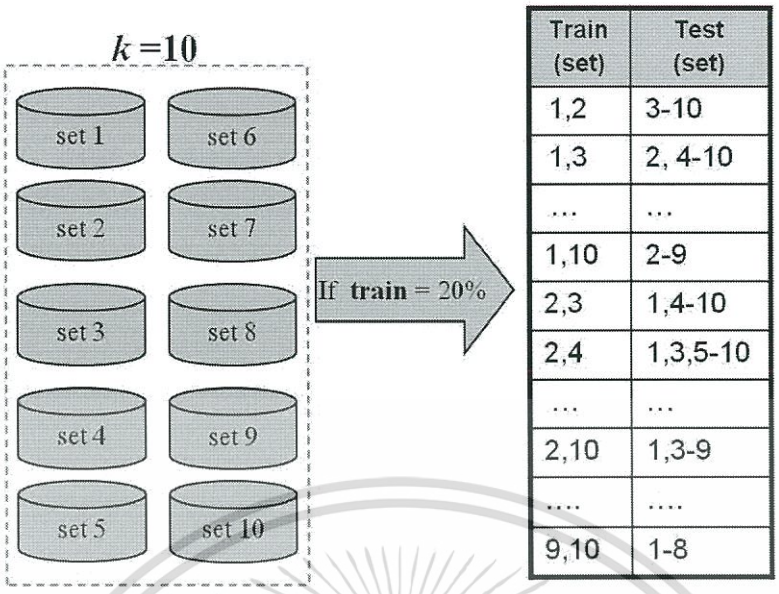


Figure 2.7 The divided data set as training 20%

2.4.2 Numerization

Nevertheless, the data sets used in this thesis which meet the demands of proposed methods must be numerical value. Therefore, the symbolic or nominal or categorical data should be transformed into numerical data, otherwise they are not computed. Therefore, all categories of a nominal attribute of all data sets in this thesis are converted by a technique of numerization of nominal attributes. There are two ways to perform numerization of nominal attributes [31-34]. One method is to map the values of a nominal attribute to integers, which is named "integer coding"; where integer values are between 0 to k-1; where k is the number of values in the nominal attribute. For example, if an attribute can take three possible values such as an attribute of protocol type in the KDD 1999 data set which has three values as follows: icmp, tcp, udp, then these nominal values are turned into a set of integers, i.e., {0, 1, 2} respectively. The disadvantage of this approach, however, lies in the fact that it imposes an order that does not exist in the original data. Another method is to divide a nominal attribute into n binary attributes, which is called "binary coding" or "dummy coding", if there are possible values (here, $n > 2$), with 0/1 representing the absence or presence of each value.

For instance, an attribute of protocol type in the KDD 1999 data set which has three values as follows: icmp, tcp, udp, then this attribute can be turned into numeric based on binary coding as follows:

ID	Protocol type	…
1	icmp	
2	tcp	
3	udp	
4	tcp	

Before mapping

Map into

ID	Protocol_ type_icmp	Protocol_ type_tcp	Protocol_ type_udp	…
1	**1**	0	0	
2	0	**1**	0	
3	0	0	**1**	
4	0	**1**	0	

After mapping

Figure 2.8 An example of mapping nominal based on binary coding technique

However, although this method overcomes the shortcomings of the first integer approach, it will generate a large set of derived attributes if n is large; moreover, it does transform the structure of data set. Hence, in this thesis, the integer approach is employed.

2.4.3 Min-Max Normalization

After mapping nominal data into numerical data, all values of attributes in each data set are made equivalent by using the Min-max normalization. The Min-max normalization is the simplest normalization technique and the most commonly used to standardize the range of independent attributes or features of data sets [34, 35]. Min-max normalization is the simplest normalization technique that is best-suited for the cases where the bounds of the scores produced by a classifier are known.

- $V_{\min}$ = minimum of the value in the data set;
- $V_{\max}$ = maximum of the value in the data set;

In this case, given a set of matching values x_i, i=1, 2,…, M, the set of normalized scores is given by as follows:

$$x_i' = \frac{x_1 - V_{\min}}{V_{\max} - V_{\min}} \qquad (4)$$

2.5 Cosine Similarity

Cosine similarity [36-38] is a measure of similarity between two vectors of n dimensions by finding the cosine of the angle between them. The cosine measure has the range [-1, 1]. The cosine of the angle between two vectors thus determines whether two vectors are pointing in roughly the same direction.

This is often used to compare documents in text mining. Given two vectors of attributes, $A = \{x_1, x_2, ..., x_n\}$ and $B = \{y_1, y_2, ..., y_n\}$, the cosine similarity θ, is the measure of the angle between the two vectors and is defined as:

$$Sim(A, B) = Cos\theta = \frac{\vec{A} \bullet \vec{B}}{\left\|\vec{A}\right\| \bullet \left\|\vec{B}\right\|} = \frac{\sum_{i=1}^{n} (x_i \bullet y_i)}{\sqrt{\sum_{i=1}^{n} x_i^2 \bullet \sum_{i=1}^{n} y_i^2}} \quad (5)$$

The resulting similarity ranges from -1 meaning exactly opposite, to 1 meaning exactly the same, with 0 usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity. For text matching, the attribute vectors A and B are usually the term frequency vectors of the documents.

In this thesis, A and B are represented any attributes in each data set while x_i and y_i are referred to each value in A and B. Moreover, the cosine similarity of two attributes will range from 0 to 1, since the similarity value cannot be negative. The angle between two vectors cannot be greater than 90°.

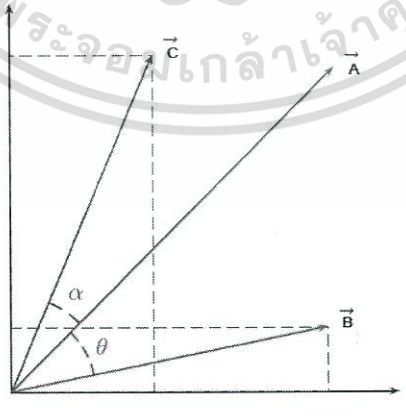


Figure 2.9 The victor space model: Cosine Similarity

2.6 Euclidean Distance

In mathematics, Euclidean space [39-41] is the Euclidean plane, the three-dimensional space of Euclidean geometry. Classical Greek geometry defined the Euclidean plane and Euclidean three-dimensional space using certain postulates, while the other properties of these spaces were deduced as theorems. In modern mathematics, it is more common to define Euclidean space using Cartesian coordinates and the ideas of analytic geometry. This approach brings the tools of algebra and calculus to bear on questions of geometry, and has the advantage that it generalizes easily to Euclidean spaces of more than three dimensions. From the modern viewpoint, there is essentially only one Euclidean space of each dimension. In dimension one this is the real line; in dimension two it is the Cartesian plane and in higher dimensions, it is a coordinate space with three or more real number coordinates, an n-dimensional real coordinate space. A point in Euclidean space may be identified by a tuple of real numbers, and distances are defined using the Euclidean distance formula. Mathematicians often denote the n-dimensional Euclidean space by $\mathbb{R}^n$, or sometimes $\mathbb{E}^n$ if they wish to emphasize its Euclidean nature.

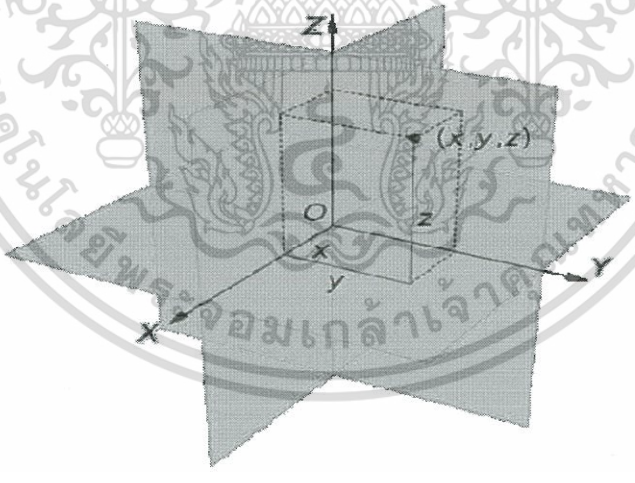


Figure 2.10 Every point in three-dimensional Euclidean

Euclidean space is more than just a real coordinate space. In order to apply Euclidean geometry one needs to be able to talk about the distances between points and the angles between lines or vectors. The natural way to obtain these quantities is by introducing and using the standard inner product (also known as the dot product) on $\mathbb{R}^n$. The inner product of any two real n-vectors x and y is defined by as follows:

$$x \cdot y = \sum_{i=1}^{n} x_i y_i = x_1 y_1 + x_2 y_2 + ... + x_n y_n \quad (6)$$

The result is always a real number. Furthermore, the inner product of x with itself is always nonnegative. This product allows us to define the "length" of a vector x as:

$$\|x\| = \sqrt{x \cdot x} = \sqrt{\sum_{i=1}^{n} (x_i)^2} \quad (7)$$

This length function satisfies the required properties of a norm and is called the Euclidean norm on $\mathbb{R}^n$. The (non-reflex) angle $\theta (0° \le \theta \le 180°)$ between x and y is then given by:

$$\theta = \cos^{-1}\left(\frac{x \cdot y}{\|x\| \cdot \|y\|} \right) \quad (8)$$

Where $\cos^{-1}$ is the arccosine function.

Finally, one can use the norm to define a metric (or distance function) on R^n by

$$d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (9)$$

This distance function is called the Euclidean metric or Euclidean distance.

Euclidean distance is the most common use of distance [42-45]. In most cases when people said about distance, they will refer to Euclidean distance. Euclidean distance or simply "distance" examines the root of square differences between coordinates of a pair of objects. In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points. The Euclidean distance between two points $A = (x_1, x_2, x_3, ..., x_n)$ and $B = (y_1, y_2, y_3, ..., y_n)$ is defined as:

$$d(A,B) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + ... + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (10)$$

In this thesis, A and B are represented any attributes in each data set while x_i and y_i are referred to each value in A and B. Moreover, the lowest distance value between two attributes will be "0".

2.7 Evaluation Measurement Techniques

To evaluation the proposed approach, the accuracy rate of classification, and *t*-Test are used in this thesis.

2.7.1 Accuracy Rate of Classification

The accuracy rate of classification is the degree of closeness to actual value. The accuracy rate is computed from number of correct classifications divided by the total number of classifications:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{11}$$

Note that the true positives (*TP*) and true negatives (*TN*) are correct classification while a false positive (*FP*) representing for the outcome is incorrectly predicted as positive when it is actually negative. A false negative (*FN*) meaning for the outcome is incorrectly predicted as negative when it is actually positive.

		Predicted Class	
		yes	*no*
Actual Class	*yes*	TP	FN
	no	FP	TN

Figure 2.11 Confusion Matrix of a two-class prediction

2.7.2 *t*-Test

A *t*-Test is any statistical hypothesis test in which the test statistic, *t*-Test can be computed by using equation (13). In this thesis, the *t*-Test is used for comparing whether the accuracy rate of the proposed approach is better than other methods at 0.05 level of statistical significance or 95% confidence interval.

Note that H_0 represents the null hypothesis. There is no difference between the mean of the accuracy rate of the proposed approach and the mean of the accuracy rate of the other approaches while H_1 represents the mean of the accuracy rate of the proposed approach is better than the mean of the accuracy rate of the other approaches.

$$H_0 : \mu_{proposed} = \mu_{other}$$

$$H_1 : \mu_{proposed} > \mu_{other}$$

μ represents the average of accuracy rate from either the proposed approach or the other approaches.

$$t = \frac{\overline{X} - \mu_D}{S.D./\sqrt{n}} \qquad (13)$$

; Where $\overline{X}$ =the mean value of summation of the difference

$S.D.$=the standard deviation of the difference

μ_D =the mean difference in the population, given a true H_0 {often μ_D =0, but not always}

n=the number of values

Since the accuracy rate of methods from the experimental results are slightly different, the assumption was set that the accuracy rate of the proposed approach is better than other methods at 0.05 level of statistical significance, the hypothesis test can be determined as follows:

H_0 : The accuracy rate of the proposed approach has no difference from each accuracy rate of other methods ($H_0 : \mu_{proposed} - \mu_{other} = \mu_D = 0$).

H_1 : The accuracy rate of the proposed approach greater than each accuracy rate of other methods ($H_1 : \mu_{proposed} - \mu_{other} > \mu_D > 0$).

For example,

Table 2.3 An example of accuracy rate of two methods.

Accuracy Rate (%)		
Data Set	GA	Proposed Approach
1	90.51	99.8
2	90.49	99.9
3	91.1	99.8

Step 1: Determine Hypothesis:

H_0 : The accuracy rate of the proposed approach has no difference from accuracy rate of GA methods ($H_0 : \mu_{proposed} - \mu_{GA} = \mu_D = 0$).

H_1 : The accuracy rate of the proposed approach has no difference from accuracy rate of GA methods ($H_1 : \mu_{proposed} - \mu_{GA} > \mu_D > 0$).

Step 2: Compute *t*-Test by using equation (13):

Table 2.4 The different values between proposed approach and GA.

Accuracy Rate (%)			
Data Set	GA	Proposed Approach	X_i
1	90.51	99.8	9.29
2	90.49	99.9	9.41
3	91.1	99.8	8.7
			$\sum X$ =27.4

$$\bar{X} = \frac{\sum X}{n} = \frac{27.4}{3} = 9.1333333$$

$S.D.$=0.3800439

$$\therefore t = \frac{\bar{X} - \mu_D}{S.D./\sqrt{n}} = \frac{9.1333333 - 0}{0.3800439/\sqrt{3}} = 41.625$$

Step 3: Determine the level of statistical significance:

$\alpha = 0.05$

; Where α represents level of statistical significance

Step 4: Find the values of $t_{\alpha,df}$ by using Critical Values of t (see the appendix B):

; Where $\alpha = 0.05$, df (degree of freedom) = n-1 = 2

$\therefore\ t_{0.05,2} = 4.303$

Step 4: Compare t with $t_{0.05,2}$:

$t = 41.625$

$t_{0.05,2} = 4.303$

$\therefore\ t > t_{0.05,2}$

Step 5: Interpretation and conclusion of hypothesis:

Since t is more than $t_{0.05,2}$, H_0 is rejected and H_1 is accepted that means the accuracy rate of proposed approach is better than the accuracy rate of the GA method.

The above example shows the method of computing t-Test without using any statistical programs. However, in this thesis, IBM SPSS Statistics is the statistical program used to compute t-Test; therefore, Sig.(2-tailed) value obtained from the program is used to interpret and conclude the hypothesis as follows:

1. If Sig.(2-tailed) value/2 is more than the 0.05 level of statistical significance, H_0 is accepted that means the accuracy rate of the proposed approach is not better than the accuracy rate of any methods that are used to compare.

2. If Sig.(2-tailed) value/2 is less than the 0.05 level of statistical significance, H_0 is rejected and H_1 is accepted that means the accuracy rate of the proposed approach is better than the accuracy rate of any methods that are used to compare.

	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
				Lower	Upper			
EUCOSSIM - GA	9.1333333	.3800439	.2194184	8.1892521	10.0774146	41.625	2	.001

Figure 2.12 Sig.(2-tailed) of *t*-Test from IBM SPSS Statistics program

Figure 2.12 shows an example of output of *t*-Test from IBM SPSS Statistics program. According to Figure 2.12, the value of Sig.(2-tailed)/2 is used to make a decision to accept hypothesis H_0 or H_1. From this table, it can be concluded that H_0 can be rejected; thus, the accuracy rate of the EUCOSSIM approach is better than the accuracy rate of the GA method.

CHAPTER 3

A PROPOSED APPROACH

The proposed approach is a good method to select or extract significant features used to build a prediction.

The EU-COSSIM is less complexity, smaller storage space, higher accuracy, and more reliable than the other techniques, especially machine learning techniques, since those techniques are more complicated and more difficult to understand and to implement into real-world application as well. Furthermore, the proposed approach does not rely on the class label like our previous method and also the proposed approach can select a subset of features without using the threshold parameter.

The EU-COSSIM is based on hypothesis as follows: (1) each feature in a group should be relevant with other features in a group and (2) any feature will be determined as a significant feature if it is relevant to other features in a group more than once. Therefore, with this proposed approach, the threshold is not used. Moreover, the method that can help us to calculate a relevant value of each feature is led to use. EU-COSSIM is divided into two approaches; each one uses different equations. Both employ the same training set. Finally, a set of selected robust features is a union set of feature sets from each approach. A robust feature set is applied to build a predictive model. Figure 3.1 shows the overall for system architecture in EU-COSSIM.

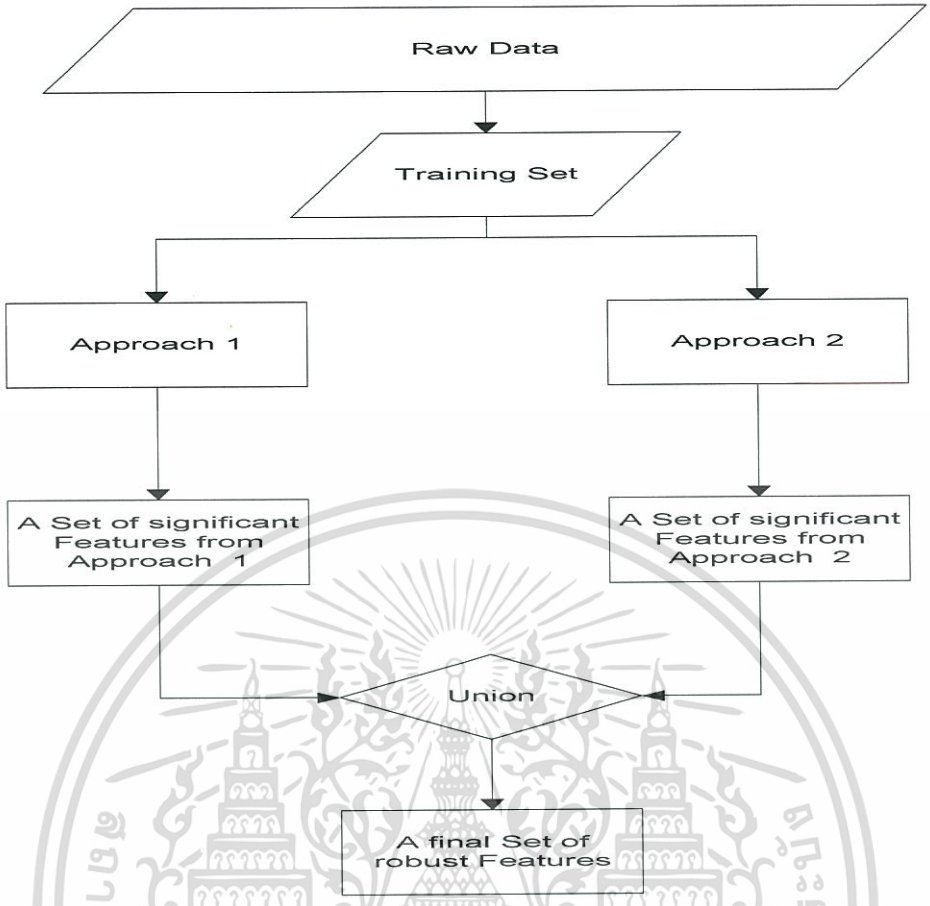


Figure 3.1 The system architecture for EU-COSSIM

In this thesis, Euclidean Distance and Cosine Similarity are used to compute a relevant value of each feature (attribute) with among other features in training set to find which feature is most relevant with each feature by considering which feature gives a maximum value of the distance. Now, the most relevant feature of each feature in training set is known. There is a set of relevant features after that negligible features are filtered out in order to keep only significant features. A criterion used to remove unimportant features is "each relevant feature, whose frequency is less than two, is removed from a set of relevant features. Therefore, a set of relevant features get rid of insignificant features. A relevant feature whose frequency is more than one is promoted as a significant feature". Nevertheless, since there are two methods (Euclidean distance and Cosine similarity), there are two significant subsets of features. Thus, to get a set of robust features, two significant subsets of features are union together.

To understand more clearly the proposed approach has an algorithm descripted step by step as follows:

In the proposed approach 1, each attribute of n attributes without class attribute in training data set is represented as A_1, A_2, A_3,..., A_n respectively as shown in Figure 3.2. The algorithm is as follows:

- Let L be the number of attributes. Therefore L is equal to n. Then, $C_{i,k}$ is computed by using an equation (14) , where $1 \leq i \leq L$ and $1 \leq k \leq L$. $C_{i,k}$, for any i, is an element in C_k as shown in Figure 3.3. $C_{i,k}$ must have a value in rank of 0 and 1. $C_{i,k} = 1$ where $i = k$. $C_{i,k}$ should be computed when i is less than k because of $C_{i,k} = C_{k,i}$. Figure 3.4 looks like a lower triangular matrix. The number of elements in this matrix is equal to (L(L-1))/2 where the number of attributes, L = n.
- Then find the maximum value $C_{i,k}$ where $i \neq k$ in each C_k and then put i, the selected attribute index, in the set R1 by keeping its frequency also.
- Each attribute index, whose frequency is less than two, is removed from R1. Therefore, R1 is a set of attributes whose frequency is more than one.
- Set R1 is modified be a set of reduced redundant attributes.

$$C_{i,k} = \frac{\sum_{j=1}^{m} (x_{j,i} \bullet x_{j,k})}{\sqrt{\sum_{j=1}^{m} x_{j,i}^2 \bullet \sum_{j=1}^{m} x_{j,k}^2}} \qquad (14)$$

$$C_{i,k} = \sqrt{\sum_{j=1}^{m} (x_{j,i} - x_{j,k})^2} \qquad (15)$$

The proposed approach 2 is similar to the proposed approach 1. Each attribute of n attributes without class attribute in training data set is represented as A_1, A_2, A_3,..., A_n respectively as shown in Figure 3.2. The algorithm is as follows:

- Let L be the number of attributes. Therefore, L is equal to n. Then, $C_{i,k}$ is computed by using an equation (15) where $1 \leq i \leq L$ and $1 \leq k \leq L$. $C_{i,k}$, for any i, is an element in C_k as shown in Figure 3.3. $C_{i,k}=0$ where $i = k$. $C_{i,k}$ should be computed when i is less than k because of $C_{i,k} = C_{k,i}$. Figure 3.4 looks like a lower triangular matrix. The number of elements in this matrix is equal to (L(L-1))/2.
- Then find the maximum value $C_{i,k}$ where $i \neq k$ in each C_k and then put i, the selected attribute index, in the set R2 by keeping its frequency also.

- Each attribute, whose frequency is less than two, is removed from R2. Therefore, R2 is a set of attributes whose frequency is more than one.
- Set R2 is modified be a set of reduced redundant attributes.

R1 and R2 are sets of significant features from both proposed approaches. For the next step, both sets of significant features are utilized to achieve a final set of robust features by using union set operation. S = R1UR2. Finally, a final set S consists of robust features that are used to generate a model.

A pseudo code of the proposed algorithm

```
PROGRAM

PROCEDURE R1
BEGIN
/* compute each C_{i,k} using equation (14) */
1.    n = the number of attributes;
2.    m= the number of instances;
3.    L=n;
4.    R1 as array[L];
5.    FOR   k=1  TO L
6.        FOR   i=1  TO L
7.        t1=0;
8.        t2=0;
9.        t3=0;
10.           FOR   j=1 TO m
11.               t1=(x_{j,i}*x_{j,k})+t1;
12.               t2=(x_{j,i})^2+t2;
13.               t3=(x_{j,k})^2+t3;
                  END LOOP
14.           C_{i,k}=t1/sqrt(t2*t3);
          END LOOP
      END LOOP
/* to find maximum value in each C_k to get Set R1 */
15.   FOR   k=1  TO  L
16.       FOR   i=1 TO  L
17.           IF   i≠ k  THEN
18.               IF  C_{i,k} = MAXIMUM(C_k)  THEN
19.                   R1_k=i;
                  END IF
```

```
                END IF
            END LOOP
        END LOOP
/* to get the significant feature subset from Set R1 */
20.  Remove each attribute index, whose frequency is less than two in
R1, from R1;
END
END PROCEDURE R1
PROCEDURE R2
BEGIN
/* compute each C_{i,k} using equation (15) */
21.  n = the number of attributes;
22.  m= the number of instances;
23.  L=n;
24.  R2 as array[L];
25.  FOR   k=1  TO  L
26.       FOR    i=1  TO  L
27.    t1=0;
28.         FOR    k=1 TO      m
29.           t1=(x_{j,i}- x_{j,k})^^2+t1
           END LOOP
30.         C_{i,k}= sqrt(t1);
       END LOOP
    END LOOP
/* to find maximum value in each C_k to get Set R2 */
31.  FOR   k=1  TO  L
32.       FOR    i=1 TO  L
33.         IF   i≠ k  THEN
34.           IF  C_{i,k} = MAXIMUM(C_k)  THEN
35.             R2_k=i;
                END IF
            END IF
        END LOOP
    END LOOP
/* to get the significant feature subset from Set R2 */
36.  Remove each attribute index, whose frequency is less than two in
R2, from R2;
END
END PROCEDURE R2

BEGIN
37. S as array[L]
```

```
/* S consists of the robust features getting from R1 U R2 */
38. S = R1 U R2;
END
```

Figure 3.2 A pseudo code of the proposed algorithm

	Attributes				
Instances	A_1	A_2	A_3	...	A_n
	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,n}$
	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{2,n}$
	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,n}$
	⋮	⋮	⋮	⋮	⋮
	$x_{m,1}$	$x_{m,2}$	$x_{m,3}$	...	$x_{m,n}$

Figure 3.3 Vectors of each attribute

An Ordinal Number of Attributes	C_1	C_2	C_3	...	C_n
	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	...	$C_{1,n}$
	$C_{2,1}$	$C_{2,2}$	$C_{2,3}$	...	$C_{2,n}$
	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$	...	$C_{3,n}$
	⋮	⋮	⋮	⋮	⋮
	$C_{m,1}$	$C_{m,2}$	$C_{m,3}$	...	$C_{m,n}$

Figure 3.4 Structure of any vector C_k

Figure 3.5 shows structure of any vector C_k used to keep each value computed by the proposed equation (14) or (15).

A Computation Loop				
1st	2nd	3rd	...	Nth
C_1	C_2	C_3	...	C_n
$C_{1,1}$			...	
$C_{2,1}$	$C_{2,2}$		...	
$C_{3,1}$	$C_{3,2}$	$C_{3,3}$	...	
⋮	⋮	⋮	⋮	⋮
$C_{m,1}$	$C_{m,2}$	$C_{m,3}$	...	$C_{m,n}$

Figure 3.5 A computation loop of any vector C_k

3.1 Example of Applying the EU-COSSIM

Below it shows how to select a feature subset of KDD 1999 Cup data set using the proposed method as follows:

From the proposed approach 1, each attribute of 41 attributes in KDD Cup 1999 training set is represented as A_1, A_2, A_3,..., A_{41} respectively as shown in Figure 3.6.

Next let L be the number of attributes. Therefore L is equal to 41. Then, $C_{i,k}$ is computed by using an equation (14) , where $1 \leq i \leq L$ and $1 \leq k \leq L$. $C_{i,k}$, for any i, is an element in C_k as shown in Figure 3.7. $C_{i,k}$ must have a value in rank of 0 and 1. $C_{i,k}=1$ where i = k. $C_{i,k}$ should be computed when i is less than k because of $C_{i,k} = C_{k,i}$. Figure 6 looks like an upper triangular matrix. The number of elements in this matrix is equal to (L(L-1))/2. Then find the maximum value $C_{i,k}$ in each C_k, and then put i, the selected attribute index, in the set R1 by keeping its frequency also. After that each attribute, whose frequency is equal to one, is removed from R1. Therefore, R1 is a set of attributes whose frequency is more than one. Finally set R1 is modified be a set of reduced redundant attributes.

From the proposed approach 2, this approach is similar to the proposed approach 1. Each attribute of 41 attributes in KDD Cup 1999 training set is represented as A_1, A_2, A_3,..., A_{41} respectively as shown in Figure 3.6. Then let L be the number of attributes. Therefore, L is equal to 41. Then, $C_{i,k}$ is computed by using an equation (15) where $1 \leq i \leq L$ and $1 \leq k \leq L$. $C_{i,k}$, for any i, is an element in C_k as shown in Figure 3.7. $C_{i,k}=0$ where i = k. $C_{i,k}$ should be computed when i is less than k because of $C_{i,k} = C_{k,i}$. Next find the maximum value $C_{i,k}$ in each C_k, and then put i, the selected attribute index, in the set R2 by keeping its frequency also. After that each attribute, whose frequency is equal to one, is removed from R2. Therefore, R2 is a set of attributes whose frequency is more than one. Finally set R2 is modified be a set of reduced redundant attributes.

Instances	Attributes				
	A_1	A_2	A_3	...	A_{41}
	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,41}$
	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{2,41}$
	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,41}$
	⋮	⋮	⋮	⋮	⋮
	$X_{m,1}$	$X_{m,2}$	$X_{m,3}$	...	$X_{m,41}$

Figure 3.6 Vectors of each attribute in KDD 1999 Cup data set

an Ordinal Number of Attributes	an Ordinal Number of Attributes				
	C_1	C_2	C_3	...	C_{41}
	$C_{1,1}$			...	
	$C_{2,1}$	$C_{2,2}$		...	
	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$	...	
	⋮	⋮	⋮	⋮	⋮
	$C_{1,41}$	$C_{2,41}$	$C_{3,41}$	...	$C_{41,41}$

Figure 3.7 Structure of any vector C_k in KDD 1999 Cup data set

Now, there are R1 and R2 that are sets of significant features from the proposed approache1 and the proposed approach 2 respectively. Next step, a union of two sets R1 and R2 is a robust feature subset that is used to generate a model for detecting intruders.

3.2 Discussion of the proposed approach

The proposed approach was separated into two methods because they can provide a feature subset that can cover all groups of class in any data sets. The only method is not sufficient to select features for building predictive patterns wrapping every class; thereby, if there is more than one method to select features, it would be better. In Figure 3.8, it shows only one pattern that is represented by a circle cannot wrap every class such as A, B, C, D, E because D and E class is not covered by a circle. Thus, it should have another method used to build a new pattern to capture the rest of the class.

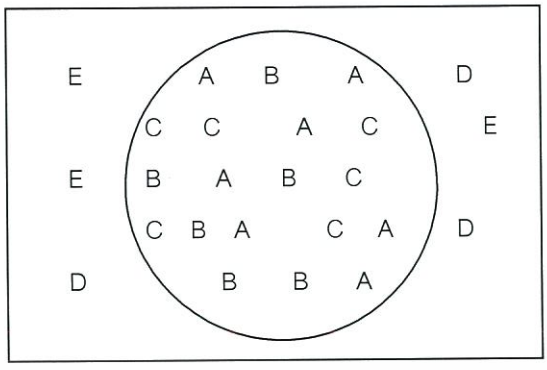


Figure 3.8 An example of pattern that cannot cover every class

Therefore, in this thesis, two methods were used to select a feature subset. The proposed approach 1 is based on the Cosine similarity method. The proposed approach 1 is a core technique used for select features that can wrap the class as much as possible. Since the proposed approach 1 discards all irrelevant features, it may discard too much features. Only the core relevant features maybe not cover all characteristics of the classes. Then it needs to find the essential irrelevant features by doing the proposed approach 2. The proposed approach 2 uses the Euclidean distance method find suitable features that cover the rest of the class. The main idea is the proposed method try to find the core features that cover for all characteristics of the classes.

Moreover, two proposed approaches use a maximum value of features computed by Cosine similarity and Euclidean distance to consider selecting feature based on using the Union operation. Since the proposed approach 1 is based on the Cosine similarity method, the maximum value help to know which feature is important or relevant most while the proposed approach 2 based on Euclidean distance method is used to find the outer features that are discarded from the proposed approach 1; hence, the rest of suitable features have to consider from the maximum value also because if the proposed approach 2 is considered by the minimum value, the selected features are possibly quite similar to the features selected by the proposed approach 1.

However, in Chapter 4, the Section 4.1 is shown the evaluation of our two proposed approach considered the only maximum value and the Union operation by comparing their accuracy rate to the accuracy rate of our two proposed approach using the other operations with utilizing the only maximum, the only minimum, or the maximum and minimum.

CHAPTER 4

EXPERIMENTAL EVALUATION

In order to evaluate a final set of robust features gained from our proposed approach, the six benchmark data sets from the UCI repository [29](KDD 1999 cup, KDD 2004 cup, KDD 2008 cup, CoverType, Zoo and Arcene) and C5.0, SVM, and RIPPER Algorithms described in Chapter 2 are used to evaluate the proposed approach. The Section 4.1 displays the experimental result of the proposed approach using different operations while the Section 4.2 shows the experimental result of KDD 1999 cup data set. Section 4.3 is about evaluating the proposed approach with other data sets. For environment of our experimental evaluation, the HP Pavilion P6000 Series Desktop with Intel® Core™ i5 CPU 2.40 GHz, and 4 GB of RAM were used. Moreover, for C5.0 algorithm, there was no set up any parameters. For SVM algorithm, the kernel type was set up as "radial basis function". For RIPPER algorithm, the pruning parameter was set as "False".

4.1 Evaluation of the proposed approach using different operations

Table 4.1 A feature subset of KDD cup 1999 data set based on the proposed approach using different operations.

No.	Type	Features	No. Features
1	$C_{max} \cup E_{max}$	{2,5,13,14,15,16,24,27,28,30,34,35,41}	13
2	$C_{min} \cup E_{min}$	{5,7,8,9,15,18,25,27,28,29,33,34,41}	13
3	$C_{max} \cup E_{min}$	{2,7,9,13,14,15,16,24,25,27,28,29,30,33,34,35,41}	17
4	$C_{min} \cup E_{max}$	{5,7,8,9,15,18,28,41}	8
5	$C_{max} \cap E_{max}$	{Ø}	0
6	$C_{min} \cap E_{min}$	{7,9,15,28,41}	5
7	$C_{max} \cap E_{min}$	{15,27,28,34,41}	5
8	$C_{min} \cap E_{max}$	{Ø}	0
9	C_{max} XOR E_{max}	{2,5,13,14,15,16,24,27,28,30,34,35,41}	13
10	C_{min} XOR E_{min}	{5,8,18,25,27,29,33,34}	8
11	C_{max} XOR E_{min}	{2,7,9,13,14,16,24,25,29,30,33,35}	12
12	C_{min} XOR E_{max}	{7,8,9,15,18,28,41}	7

Table 4.2 The accuracy rate of different predictive algorithms using KDD cup 1999 data set based on the proposed approach using different operations.

No.	Type	C5.0			SVM			RIPPER		
		D1	D2	D3	D1	D2	D3	D1	D2	D3
1	$C_{max} \cup E_{max}$	99.8000	99.5700	99.5800	99.9422	99.6298	99.5909	99.9829	99.6824	99.6695
2	$C_{min} \cup E_{min}$	94.7087	94.6751	94.5722	93.7376	93.3052	93.1525	94.9186	94.5850	94.3169
3	$C_{max} \cup E_{min}$	98.8163	98.7417	98.7124	98.3194	98.0841	98.0062	98.8335	98.4539	98.1306
4	$C_{min} \cup E_{max}$	90.4779	90.4726	90.4420	89.4375	89.1586	89.0052	90.5372	90.5207	90.5061
5	$C_{max} \cap E_{max}$	-	-	-	-	-	-	-	-	-
6	$C_{min} \cap E_{min}$	89.9714	89.9668	89.9460	89.4851	89.0647	88.9852	89.9719	89.9673	89.9511
7	$C_{max} \cap E_{min}$	93.3097	93.1199	93.0183	91.5607	90.4118	90.1487	93.3236	93.1607	93.0953
8	$C_{min} \cap E_{max}$	-	-	-	-	-	-	-	-	-
9	C_{max} XOR E_{max}	99.8000	99.5700	99.5800	99.9422	99.6298	99.5909	99.9829	99.6824	99.6695
10	C_{min} XOR E_{min}	94.5811	94.3881	94.3004	94.2964	93.9778	93.9622	94.6292	94.4296	94.3014
11	C_{max} XOR E_{min}	96.7879	96.2024	96.0965	96.9964	96.3778	96.0622	96.8292	96.4296	96.2014
12	C_{min} XOR E_{max}	89.9714	89.7140	89.4644	89.4535	89.0606	89.0002	89.9724	89.7144	89.4859

This section shows the accuracy rate of the proposed approach that uses different operations that are Union operation (U), Intersection operation (∩) and XOR operation (XOR); moreover, in procedure of selecting feature, it not only considers a maximum of each value of each feature computed from equations (14) and equation (15) in chapter 3, but also it uses the maximum and minimum because of needing to compare the effective accuracy rate of the proposed approach using the Union operation with employing the only maximum value of each feature to select features with the other accuracy rates of the proposed approach using the other operations with utilizing the only maximum, the only minimum, or the maximum and minimum to select features.

In Table 4.1, the second column shows each technique using different operations with employing the maximum or minimum of each feature to consider selecting features while the third column displays each subset of features selected by each technique in the second column. Final column shows the number of features of each technique.

Note that C_{max} represents the technique using the Cosine Similarity method and employing maximum to consider selecting features.

E_{max} means the technique using the Euclidean distance method and employing maximum to consider selecting features.

C_{min} denotes the technique using the Cosine similarity method and employing minimum to consider selecting features.

E_{min} is represents the technique using the Euclidean distance method and employing minimum to consider selecting features.

C_{max} U E_{max} symbolizes our proposed approach in the thesis considering the maximum value of features computed by Cosine similarity and Euclidean distance to select feature based on using the Union operation.

Table 4.2 shows the accuracy rate of KDD cup 1999 data set based on each technique's the feature subset in Table 4.1 acquired from building predictive models using C5.0, SVM, and RIPPER algorithms.

From the experimental result in Table 4.2, it shows our proposed approach in the number one considering the maximum value of features computed by Cosine similarity and Euclidean distance to select feature based on using the Union operation (C_{max} U E_{max}) can perform the

efficiency in selecting the features effectively because it can help all predictive models based on C5.0, SVM, and RIPPER algorithms to succeed in maximizing the accuracy rate when compare to the other techniques. However, a technique number 9 (C_{max} XOR E_{max}) considering the maximum value of features computed by Cosine similarity and Euclidean distance to select feature the same as our proposed approach but using a different operation that is XOR can show as the effective accuracy rate as our proposed approach. Because from Table 4.1, it find they provide the same as the feature subset. Nevertheless, the Union operation is considered to use for combining two feature subsets in the proposed approach because the computational cost of Union operation is less than XOR operation. Since the Union operation is composed of one operation while XOR is consisted of three operations that are Union, Minus, and Intersect operations ($\cup - \cap$), The Union operation must have the computational cost less than XOR operation.

Therefore, to use the only maximum as criteria to consider selecting features based on the Cosine similarity and Euclidean distance methods is one of major factors to acquire the robust subset of features when it is considered with the Union operation.

4.2 Experimental Result of KDD 1999 Cup Data Set

The KDD 1999 Cup data set (one train set and three test sets in Table 2.1) in Section 2.3(A) is used to evaluate the proposed approach.

The EU-COSSIM is compared with other five methods as follows:

1. Entire features containing 41 features.
2. GA method is proposed in [17],
3. Kok-Chine Khor approach is proposed in [20].
4. Euclidean-based method is proposed in [21].
5. Cosine-based method is proposed in [22].

Those methods are used for comparison in this thesis because all of them informed their selected features clearly. Note that the measurement in this thesis of the experimental results is based on an accuracy rate that is the standard metrics for evaluations of intrusion [27]: Detection rate (TP) refers to the ratio between the number of correctly detected attacks and the total number of attacks while false alarm rate (FP: false positive) means the ratio between the number of normal connections that are incorrectly misclassified as attacks and the total number of normal connections. Moreover, an independent-sample t-test is adopted to compare whether results are

significant difference between methods within three classification algorithms—C5.0, SVM, and RIPPER built predictive models using the proposed feature set and other feature sets.

In EU-COSSIM, the proposed approach 1 (R1) can extract 12 significant features out of 41 features while the proposed approach 2 (R2) can take only one significant feature. However, when both sets of significant features are collected together by using union operator, each robust feature is selected. The proposed approach 1 and approach 2 through union operator obtain 13 robust features displayed in Table 4.3. However, when consideration on the features obtained by the proposed method (EU-COSSIM), the Euclidean-Based method proposed in [21], and the Cosine-Based method proposed in [22] labeled sets as EUCO, E21, CO22 respectively found that EUCO is a subset of E21 union with CO22 or can labeled as EUCO $\subset$ {E21 $\cup$ CO22}, EUCO $\not\subset$ E21 and EUCO $\not\subset$ CO22. Therefore, the proposed method is the method that can filter only important features from Euclidean-Based and Cosine-Based to reduce the large number of features to the small number of features.

Note that each italic word in Table 4.3 represents the same features occurring in the proposed method, the Euclidean-Based method, and Cosine-Based method, while each bold word represents the same features occurring only in the proposed method and the Cosine-Based method.

In Table 4.4(a), the performance of C5.0 built by using five different feature sets with three different datasets found that the percentage of the accuracy rate of the proposed method (EU-COSSIM) is not quite different from the entire features, the Euclidean-Based method and the Cosine-Based method corresponding with Sig. values in Table 4.6(a) that shows there was no significant difference between the proposed method and the Euclidean-Based method, and the Cosine-Based method except the GA method and the Kok-Chine Khor approach. However, when the average of the percent accuracy rate is used to consideration, it found the proposed method is more effective than other methods in term of the accuracy and reliability. Regarding Table 4.4(b) and Table 4.6(b), the performance of SVM built by using five different feature sets with three different datasets shows that the proposed method is significant difference from other methods except the Kok-Chine Khor approach. These results show that the subset of features provided by the proposed method can work with SVM algorithm effectively.

Table 4.3 Entire feature set and selected features for four methods.

Method	A Subset of Features Selected	No. Features
Entire Features	duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login, count, serv_count, serror_rate, srv_serror_rate, reerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate	41
GA	service, flag, wrong_fragment, hot, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_srv_serror_rate, duration, src_bytes, srv_serror_rate, land, urgent, su_attempted, num_root, num_shells, num_access_files, isguest_login	21
Kok-Chine Khor approach	service, dst_bytes, logged_in, count, dst_host_count, root_shell, dst_host_rerror_rate	7
Euclidean-Based	duration, *protocol_type*, logged_in, serror_rate, srv_serror_rate, *reerror_rate*, *srv_rerror_rate*, *diff_srv_rate*, srv_diff_host_rate, dst_host_diff_srv_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, *dst_host_rerror_rate*, dst_host_srv_rerror_rate, src_bytes, dst_bytes, land, wrong_fragment, urgent, host, num_failed_logins, *num_compromised*, *root_shell*, *su_attempted*, *num_root*, num_file_creations, num_shells, num_access_files, is_guest_login	30
Cosine-Based	duration, *protocol_type*, logged_in, serror_rate, srv_serror_rate, *reerror_rate*, *srv_rerror_rate*, *diff_srv_rate*, srv_diff_host_rate, dst_host_diff_srv_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, *dst_host_rerror_rate*, dst_host_srv_rerror_rate, service, **flag** count, **serv_count**, same_srv_rate, dst_host_count, **dst_host_srv_count**, **dst_host_same_srv_rate**, dst_host_same_src_port_rate	24
EU-COSSIM	*protocol_type*, **flag**, *num_compromised*, *root_shell*, *su_attempted*, *num_root*, **serv_count**, *reerror_rate*, *srv_rerror_rate*, *diff_srv_rate*, **dst_host_srv_count**, **dst_host_same_srv_rate**, *dst_host_rerror_rate*	13

In Table 4.4(c) and Table 4.6(c), the performance of RIPPER built by using five different feature sets with three different datasets found that there is no significant difference between the proposed method and other methods. However, the proposed method can provides an effective subset of features with RIPPER algorithm for building a predictive model because the average of the percent accuracy rate is better than any others.

In Table 4.5(a)-(c), the less value is better. The results obtained from Table 4.5(a)-(c), show that the EU-COSSIM approach designed for selecting a set of robust features to build a model for detecting attacks on networking system was successfully able to extract a robust feature set for generating the model providing more accuracy of detection rate than other approaches with a lower false positive rate than other approaches also. Moreover, in Table 4.3, the number of feature set selected by the proposed approach is fewer than the number of feature sets extracted by the other methods except the Kok-Chine Khor approach. The feature set in EU-COSSIM is accounted for 31.7% of Entire feature set (13 of 41 features). This means use less storage space.

Table 4.4(a) The average of overall accuracy rate of C5.0.

Dataset	Overall accuracy rate (%) on five different feature sets					
	Entire Features	GA	Kok-Chine Khor approach	Euclidean-Based	Cosine-Based	EU-COSSIM
D1	99.8000	97.3200	99.2500	99.8100	99.7900	99.8000
D2	99.5800	97.3000	98.8200	99.4600	99.3500	99.5700
D3	99.5000	97.3100	98.6600	99.3000	99.2500	99.5800
AVG.	99.6200	97.3100	98.9100	99.5230	99.4633	99.6500

Table 4.4(b) The average of overall accuracy rate of SVM.

Dataset	Overall accuracy rate (%) on five different feature sets					
	Entire Features	GA	Kok-Chine Khor approach	Euclidean-Based	Cosine-Based	EU-COSSIM
D1	99.0104	96.9718	99.6541	97.0152	96.9873	99.9422
D2	98.5436	96.9639	99.1080	97.0812	96.9295	99.6298
D3	98.3674	96.9354	98.8513	97.0849	96.9865	99.5909
AVG.	98.6405	96.9570	99.2045	97.0604	96.9678	99.7210

Table 4.4(c) The average of overall accuracy rate of RIPPER.

Dataset	Overall accuracy rate (%) on five different feature sets					
	Entire Features	GA	Kok-Chine Khor approach	Euclidean-Based	Cosine-Based	EU-COSSIM
D1	99.8067	99.7992	99.6964	99.8067	99.3435	99.9829
D2	99.5692	99.5934	98.9826	99.5692	99.1444	99.6824
D3	99.4650	99.4846	98.8237	99.4650	99.0834	99.6695
AVG.	99.6136	99.6257	99.1676	99.6136	99.1904	99.7783

Table 4.5(a) The average of false positive rate of C5.0.

Dataset	Overall FP rate (%) on five different feature sets					
	Entire Features	GA	Kok-Chine Khor approach	Euclidean-Based	Cosine-Based	EU-COSSIM
D1	0.2000	2.6800	0.7500	0.1900	0.2100	0.2000
D2	0.4200	2.7000	1.1800	0.5400	0.6500	0.4300
D3	0.5000	2.6900	1.3400	0.7000	0.7500	0.4200
AVG.	0.3700	2.6900	1.0900	0.4800	0.5367	0.3500

Table 4.5(b) The average of false positive rate of SVM.

Dataset	Overall FP rate (%) on five different feature sets					
	Entire Features	GA	Kok-Chine Khor approach	Euclidean-Based	Cosine-Based	EU-COSSIM
D1	0.9896	3.0282	0.3459	2.9848	3.0127	0.0578
D2	1.4564	3.0361	0.8920	2.9188	3.0705	0.3702
D3	1.6326	3.0646	1.1487	2.9151	3.0135	0.4091
AVG.	1.3595	3.0430	0.7955	2.9396	3.0322	0.2790

Table 4.5(c) The average of false positive rate of RIPPER.

Dataset	Overall FP rate (%) on five different feature sets					
	Entire Features	GA	Kok-Chine Khor approach	Euclidean-Based	Cosine-Based	EU-COSSIM
D1	0.1933	0.2008	0.3036	0.1933	0.6565	0.0171
D2	0.4308	0.4066	1.0174	0.4308	0.8556	0.3176
D3	0.5350	0.5154	1.1763	0.5350	0.9166	0.3305
AVG.	0.3864	0.3743	0.8324	0.3864	0.8096	0.2217

Table 4.6(a) Significance test of classification accuracy between C5.0 built using the EU-COSSIM and other feature selection methods.

Independent-sample t-test			
EU-COSSIM	Std. Deviation	Std. Error Mean	Sig. (95%)
Entire Features	0.1225	0.0316	0.008
GA - EU-COSSIM	0.1598	0.0413	0.004
Kok-Chine Khor approach	0.1461	0.0377	0.001
Euclidean Based	0.0983	0.0254	0.001
Cosine-Based	0.1016	0.0262	0.001

Table 4.6(b) Significance test of classification accuracy between SVM built using the EU-COSSIM and other feature selection methods.

Independent-sample t-test			
EU-COSSIM	Std. Deviation	Std. Error Mean	Sig.(95%)
Entire Features	0.3363	0.0868	0.011
GA - EU-COSSIM	0.3678	0.0950	0.024
Kok-Chine Khor approach	0.3399	0.0878	0.004
Euclidean Based	0.3800	0.0981	0.032
Cosine-Based	0.3785	0.0977	0.03

Table 4.6(c) Significance test of classification accuracy between RIPPER built using the EU-COSSIM and other feature selection methods.

Independent-sample t-test			
EU-COSSIM	Std. Deviation	Std. Error Mean	Sig.(95%)
Entire Features	0.2357	0.0608	0.123
GA - EU-COSSIM	0.2362	0.0610	0.126
Kok-Chine Khor approach	0.2494	0.0644	0.021
Euclidean Based	0.2380	0.0614	0.318
Cosine-Based	0.2372	0.0612	0.292

4.3 Experimental Result of other UCI benchmark Data Sets

To increase reliability of evaluating the proposed approach, the other UCI benchmark data sets in Table 2.2 were used to do experiments with the proposed approach; moreover, the proposed approach is compared to an InfoGain method in WEKA, and the entire features. The accuracy rate and the number of features are employed as the evaluation measurement technique in this Section.

Table 4.7 shows each subset of features with other UCI benchmark data sets in Table 2.2 provided by EU-COSSIM that can reduce the number of attributes from entire attributes to the number of smaller size attributes which lead to using smaller storage space and cut down computation time. Note that each numbers in second column in Table 4.7 refers to an ordinal number of attributes.

Table 4.7 Each feature subset of other UCI benchmark data sets based on the proposed approach.

Data set	A Subset Features Selected by EU-COSSIM	NO. Features
KDD 2004	{4,5,9,10,14,18,21,23,26,27,29,30,31,33,47,51,52,53,59,66,24,49}	22
KDD 2008	{4,8,24,25,28,34,37,38,42,44,48,49,53,55,56,59,61,64,66,70,74,80,84 ,87,89,91,98,102,104,116,3,5,31,45,52,58,72,78,93,97,99,107,113}	43
Cover Type	{1,4,5,6,8,10,11,12,13,14,29}	11
Zoo	{2,5,7,10,14}	5
Arcene	please see at: *http://webserv.kmitl.ac.th/s9062952/FeaturesBasedEU-COSSIM.txt*	2,031

Each feature subset of other UCI benchmark data sets in Table 2.2 selected by InfoGain method is shown in Table 4.8. Note that InfoGain method is one of attribute selection in WAKA. It selects any features by measuring the information gain with respect to the class.

Table 4.8 Each feature subset of other UCI benchmark data sets Selected by InfoGain method.

Data set	A Subset of Features Selected by InfoGain	NO. Features
KDD 2004	{ 56,58,61,62,63,6,57,11,8,12,31,26,7,66,32,71,13,28,38,41,36, 42,27,60,72,67,48,43,4,69,37,51,59,46,77,68,70,52,16,9 }	40
KDD 2008	{83,104,70,59,38,36,37,41,42,39,40,31,29,30,34,35,32,33,52,50,51,5 5,56,53,54,45,43,44,48,49,46,47,10,8,9,13,14,11,12,3,1,2,6,7,4,5,24,2 2,23,27,28,25,26,17,15,16,20,21,18,19,95,93,94,98,99,96,97,88 }	68
Cover Type	{ 1,14,6,10,11,24,3,7,53,52,43,18,36,26,16,4,9,8,20,37,2,54,5,12,17 }	25
Zoo	{ 14,5,9,4,10,2,3,11,15,16,13 }	11
Arcene	please see at: *http://webserv.kmitl.ac.th/s9062952/FeaturesBasedInfo Gain.txt*	4,852

Table 4.9(a) The accuracy rate of five different data sets based on C5.0.

Dataset Name	Accuracy rate (%)		
	Entire Features	EU-COSSIM	InfoGain
KDD 2004	99.6000	99.6100	99.5046
KDD 2008	99.4300	99.4300	99.4154
CoverType	85.8300	92.3900	84.1569
Zoo	90.3200	90.3200	85.2941
Arcene	51.0000	83.0000	79.000

Table 4.9(b) The accuracy rate of five different data sets based on SVM.

Dataset Name	Accuracy rate (%)		
	Entire Features	EU-COSSIM	InfoGain
KDD 2004	99.8982	100.0000	99.8428
KDD 2008	99.5190	99.5464	99.5145
CoverType	86.1700	93.0400	86.0057
Zoo	97.0588	98.5294	82.3529
Arcene	56.0000	86.0000	56.0000

Table 4.9(c) The accuracy rate of five different data sets based on RIPPER.

Dataset Name	Accuracy rate (%)		
	Entire Features	EU-COSSIM	InfoGain
KDD 2004	99.7484	99.8532	99.5693
KDD 2008	99.6344	99.6569	99.6200
CoverType	86.0700	92.9400	85.1153
Zoo	97.0588	97.0588	89.7059
Arcene	53.0000	84.0000	66.0000

Table 4.10(a) The FP rate of five different data sets based on C5.0.

Dataset Name	FP rate (%)		
	Entire Features	EU-COSSIM	InfoGain
KDD 2004	0.3900	0.4000	0.4954
KDD 2008	0.5700	0.5700	0.5846
CoverType	14.1700	7.6100	15.8431
Zoo	9.6800	9.6800	14.7059
Arcene	49.0000	17.0000	21.0000

Table 4.10(b) The FP rate of five different data sets based on SVM.

Dataset Name	FP rate (%)		
	Entire Features	EU-COSSIM	InfoGain
KDD 2004	0.1018	0.0000	0.1572
KDD 2008	0.4810	0.4536	0.4855
CoverType	13.8300	6.9600	13.9943
Zoo	2.9412	1.4706	17.6471
Arcene	44.0000	14.0000	44.0000

Table 4.10(c) The FP rate of five different data sets based on RIPPER.

Dataset Name	FP rate (%)		
	Entire Features	EU-COSSIM	InfoGain
KDD 2004	0.2516	0.1468	0.4307
KDD 2008	0.3656	0.3431	0.3800
CoverType	13.9300	7.0600	14.8847
Zoo	2.9412	2.9412	10.2941
Arcene	47.0000	16.0000	34.0000

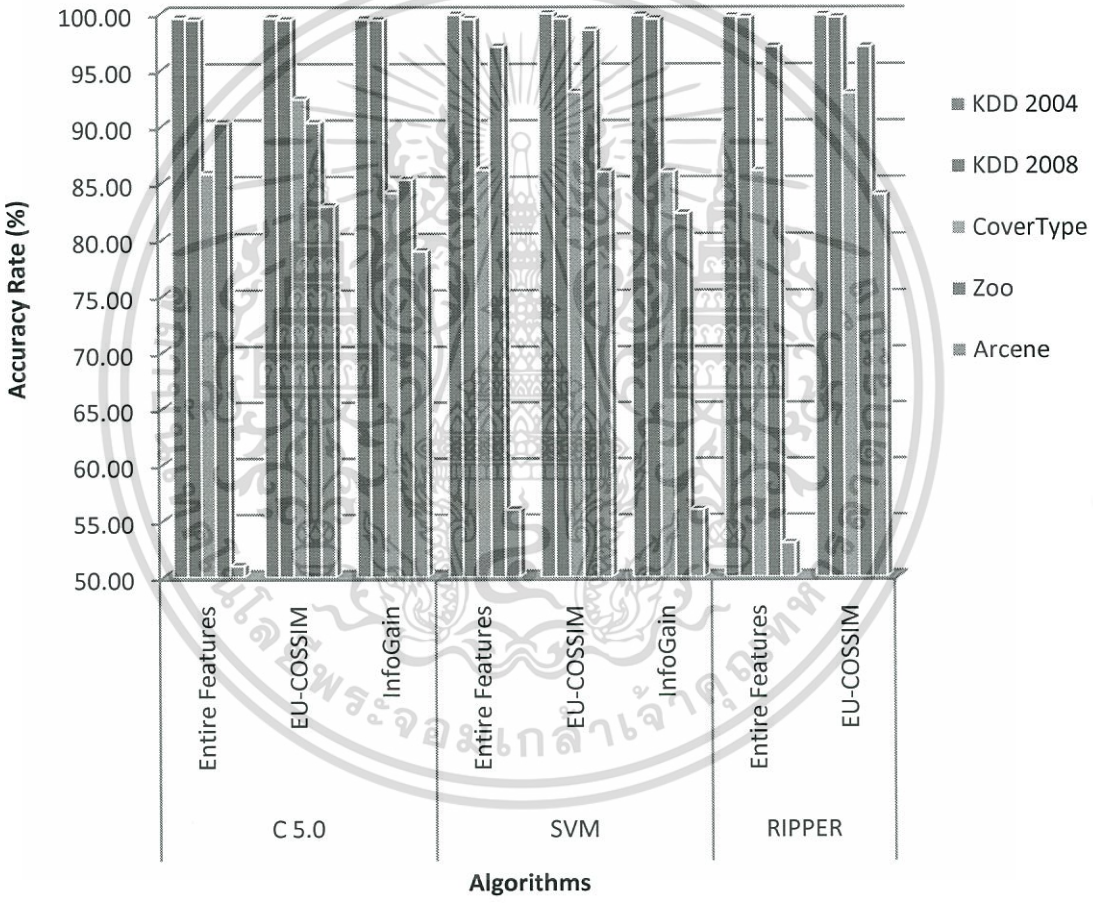


Figure 4.1 A comparison of accuracy rate among the entire features, the feature subset based on InfoGain method, and the feature subset of EU-COSSIM in each of the UCI benchmark data sets

Table4.9(a)-(c) and Figure 4.1 demonstrate a comparison of accuracy rate among the entire features, the feature subset based on InfoGain method, and the feature subset provided by EU-COSSIM using three different classification algorithms and different data sets. From experimental results in Table 4.9(a)-(c), the proposed method can show an efficiency of

prediction because its percentage of accuracy rates of the UCI benchmark data sets is quite higher compared to the entire features. However, it found that the accuracy rate of the proposed approach is quite similar to the accuracy rate based on the InfoGain and entire feature methods in KDD 2004 cup, KDD 2008 cup, and zoo data sets while in CoverType and Arcene data sets, the accuracy rate based on EU-COSSIM is better than the other techniques. Moreover, when the number of features is considered to measure the efficiency of the proposed approach found the proposed approach can reduce the number of features to the smallest size when compares to the other methods, especially the Arcene data set.

However, although the proposed approach shows the efficiency of accuracy rate based on three classification algorithms in every benchmark data sets in this thesis, when the raw data of each data set were considered found the proposed approach can still show the performance of accuracy rate while the other methods cannot do when a data type of all attributes in the data set is binary or integer, in other word, the proposed approach presents more effective than other methods if it is applied to a binary data set or an integer data set.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

To improve the performance of accuracy rate, a new feature selection method called EU-COSSIM is proposed by applied Euclidean Distance, Cosine Similarity and added new filtering feature techniques for extracting a robust feature set. EU-COSSIM is divided into two small methods. Each method has the same procedures except using the different equation mentioned thoroughly in Chapter 3. From the experimental results, the EU-COSSIM approach yielded a higher performance compared with other techniques. Besides, EU-COSSIM approach can help to solve the threshold problem because features can be selected without using a threshold parameter. Moreover, EU-COSSIM approach is not complicated technique and easier to understand because the proposed approach has few procedures composed of the two common standard equations. All procedures are simple and understandable. Furthermore, time processing is quite important. In real-world applications, any techniques extracting feature sets rapidly are always beneficial in terms of computational cost for processing data. EU-COSSIM processes fewer steps with easy equations. In the previous Chapter, the time complexity for selected feature from our proposed approach is less than other approaches. In addition, in Chapter 4, Section 4.1, the EU-COSSIM approach generates a set of robust features that is smaller size than other methods, except with Kok-Chine Khor approach. However, from experimental results, EU-COSSIM gives effectiveness of the accuracy rate and FP rate. This help to enhance performance of NIDS in the real world. Furthermore, in Section 4.2, the EU-COSSIM approach still shows its performance of higher accuracy rate with smaller size of features even if it is applies with the other benchmark data sets. The last advantage of the proposed approach is selecting a robust feature set without depending on attribute class or class label. From the experimental results, it can conclude that the EU-COSSIM approach is a simple feature selection algorithm using smaller storage space, reducing computation time, and gaining higher predictive performance.

5.2 Recommendation

In real-world applications, there are a multitude of attributes and instances. It leads to take more time to exact the robust subset of features; hence, in the future work, the proposed method will be developed to parallel computing system because the proposed method was designed to support to parallel computing system that is column-block partitioning. Thus, a proposed method based on the parallel system will help to reduce time to select the robust subset of features.

REFERENCES

[1] G. H. John, R. Kohavi and K. Pfleger, "Irrelevant features and the subset selection problem", Int. Conf. on Machine Learning, pp. 121-129, 1994.

[2] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.19, no.2, pp. 153-158, 1997.

[3] H. Liu, E. R. Dougherty, J. G. Dy, K. Torkkola, E. Tuv, H. Peng, C. Ding, F. Long, M. Berens, L. Parsons, Z. Zhao, L. Yu and G. Forman, "Evolving feature selection", Intelligent Systems, IEEE, vol.20, no.6, pp. 64-76, 2005.

[4] Y. Caballero, D. Alvarez, R. Bello and M. M. Garcia, "Feature selection algorithms using rough set theory", in Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on, pp. 407-411, 2007.

[5] T. May, A. Bannach, J. Davey, T. Ruppert and J. Kohlhammer, "Guiding feature subset selection with an interactive visualization", in Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on, pp. 111-120, 2011.

[6] W. Duch, T. Winiarski, J. Biesiada and A. Kachel, "Feature ranking, selection and discretization", Int. Conf. on Artificial Neural Networks and Int. Conf. on Neural Information Processing, pp. 251-254, 2003.

[7] K. Ron, "Feature subset selection using the wrapper method: overfitting and dynamic search space topology", Proc. AAA1 Fall Symposium on Relevance, pp. 109-113, 1994.

[8] L. H. Witten and E. Frank eds., Data mining: Practical machine learning tools and techniques, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[9] A. Asuncion and D. Newman, "Uci machine learning repository", Irvine, CA: University of California, School of Information and Computer Science, http://www.ics.uci.edu/mlearn/MLRepository.html, accessed Feb. 8. 2012.

[10] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks", Proc. Computers & Security, vol.24, pp. 31-43, 2005.

[11] Y. Bai and H. Kobayashi, "Intrusion detection systems: Technology and development", in Advanced Information Networking and Applications, 2003. AINA 2003. 17th International

Conference on, pp. 710-715, 2003.

[12] G. Ren Hui, M. Zulkernine and P. Abolmaesumi, "A software implementation of a genetic algorithm based approach to network intrusion detection", in Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on, pp. 246-253, 2005.

[13] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin and W.-Y. Lin, "Review: Intrusion detection by machine learning: A review", Expert Syst. Appl., vol.36, no.10, pp. 11994-12000, 2009.

[14] D. Junping and G. Wensheng, "Data mining on patient data", in Neural Networks and Brain, 2005. ICNN&B '05. International Conference on, pp. 84-87, 2005.

[15] L. Cheung-Leung, F. Tak-Chung and C. Ting-Yee, "Agent-based network intrusion detection system using data mining approaches", in Information Technology and Applications, 2005. ICITA 2005. Third International Conference on, pp. 131-136 vol.131, 2005.

[16] X. Wang, F. He and R. Xu, "Modeling intrusion detection system by discovering association rule in rough set theory framework", in Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on, pp. 24-24, 2006.

[17] L. Chi Hoon, S. Sung Woo and C. Jin Wook, "Network intrusion detection through genetic feature selection", in Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on, pp. 109-114, 2006.

[18] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection", Pattern Recogn. Lett., vol.10, no.5, pp. 335-347, 1989.

[19] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection", Proc. Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2, Montreal, Quebec, Canada, pp. 1137-1143, 1995.

[20] K. Kok-Chin, T. Choo-Yee and S. P. Amnuaisuk, "A feature selection approach for network intrusion detection", in Information Management and Engineering, 2009. ICIME '09. International Conference on, pp. 133-137, 2009.

[21] A. Suebsing and N. Hiransakolwong, "Euclidean-based feature selection for network intrusion detection", Proc. International Conference on Machine Learning and Computing, pp. 222-229, 2009.

[22] A. Suebsing and N. Hiransakolwong, "Feature selection using euclidean distance and cosine similarity for intrusion detection model", in Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on, pp. 86-91, 2009.

[23] J. R. Quinlan, C4.5: Programs for machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[24] J. Spilka, Chuda, x, x030C, V. ek, Kuz, J. ilek, Lhotska, L. and M. Hanuliak, "Detection of inferior myocardial infarction: A comparison of various decision systems and learning algorithms", in Computing in Cardiology, 2010, pp. 273-276, 2010.

[25] S. J. Stolfo, F. Wei, L. Wenke, A. Prodromidis and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the jam project", in DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings, pp. 130-144 vol.132, 2000.

[26] S. Mukkamala and A. H. Sung, "A comparative study of techniques for intrusion detection", in Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on, pp. 570-577, 2003.

[27] H. Wei, L. Jianhua and S. Jianjun, "Optimal evaluation of feature selection in intrusion detection modeling", in Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, pp. 5919-5922, 2006.

[28] W. Juan, Y. Qiren and R. Dasen, "An intrusion detection algorithm based on decision tree technology", in Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on, pp. 333-335, 2009.

[29] L. T. Payam Refaeilzadeh, and Huan Liu, "Cross validation", in Encyclopedia of Database Systems, ed. M. T. Ö. a. L. Liu, Springer, 2009.

[30] J. D. Rodriguez, A. Perez and J. A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.32, no.3, pp. 569-575 2010.

[31] Z. Chi, X. Weimin, T. M. Tirpak and P. C. Nelson, "Evolving accurate and compact

classification rules with gene expression programming", Evolutionary Computation, IEEE Transactions on, vol.7, no.6, pp. 519-531 2003.

[32] A. Jain, K. Nandakumar and A. Ross, "Score normalization in multimodal biometric systems", Pattern Recogn., vol.38, no.12, pp. 2270-2285 2005.

[33] S. Mei-Ling, K. Sarinnapakorn, I. Kuruppu-Appuhamilage, C. Shu-Ching, C. Liwu and T. Goldring, "Handling nominal features in anomaly intrusion detection problems", in Research Issues in Data Engineering: Stream Data Mining and Applications, 2005. RIDE-SDMA 2005. 15th International Workshop on, pp. 55-62, 2005.

[34] C. Li and G. Biswas, "Unsupervised learning with mixed numeric and nominal data", Knowledge and Data Engineering, IEEE Transactions on, vol.14, no.4, pp. 673-690 2002.

[35] R. Modugno, G. Pirlo and D. Impedovo, "Score normalization by dynamic time warping", in Computational Intelligence for Measurement Systems and Applications (CIMSA), 2010 IEEE International Conference on, pp. 82-85, 2010.

[36] Y. Soe-Tsyr and S. Jerry, "Ontology-based structured cosine similarity in speech document summarization", in Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on, pp. 508-513, 2004.

[37] Y. Soe-Tsyr and S. Jerry, "Ontology-based structured cosine similarity in document summarization: With applications to mobile audio-based knowledge management", Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol.35, no.5, pp. 1028-1040 2005.

[38] M. Loog, "On distributional assumptions and whitened cosine similarities", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.30, no.6, pp. 1114-1115 2008.

[39] S. Zhude, M. Qing, L. Xinyang and M. Jingsong, "An algorithm for shortest raster distance in euclidean space with obstacles", in Geoinformatics, 2011 19th International Conference on, pp. 1-4, 2011.

[40] K. Zeger and A. Gersho, "How many points in euclidean space can have a common nearest neighbor?", in Information Theory, 1994. Proceedings., 1994 IEEE International Symposium on, pp. 109, 1994.

[41] A. Srivastava, E. Klassen, S. H. Joshi and I. H. Jermyn, "Shape analysis of elastic curves in

euclidean spaces", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.33, no.7, pp. 1415-1428 2011.

[42] W. Liwei, Z. Yan and F. Jufu, "On the euclidean distance of images", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.27, no.8, pp. 1334-1339 2005.

[43] L. Qi, V. Kecman and R. Salman, "A chunking method for euclidean distance matrix calculation on large dataset using multi-gpu", in Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on, pp. 208-213, 2010.

[44] J. Javed, H. Yasin and S. F. Ali, "Human movement recognition using euclidean distance: A tricky approach", in Image and Signal Processing (CISP), 2010 3rd International Congress on, pp. 317-321, 2010.

[45] W. Bin, W. Lei and X. Cunliang, "Possibilistic clustering using non-euclidean distance", in Control and Decision Conference, 2009. CCDC '09. Chinese, pp. 938-940, 2009.

APPENDIX

APPENDIX A

Data Dictionary of the Evaluation Data Sets

1. KDD Cup1999 data set

Download from: *http://www.sigkdd.org/kddcup/index.php?section=1999&method=data*

Table A.1: Data dictionary of KDD Cup 1999 data set

Attribut No.	Data Type
1	Integer
2-4	Nominal
5-6	Integer
7	Binary
8-11	Integer
12	Binary
13	Integer
14-15	Binary
16-21	Integer
22	Binary
23-24	Integer
25-31	Real
32-33	Integer
34-1	Real
42	Class Label

2. KDD 2004 Cup data set

Download from: *http://www.sigkdd.org/kddcup/index.php?section=2004&method=data*

Table A.2: Data dictionary of KDD 2004 Cup data set

Attribut No.	Data Type
1-2	Unique
3	Class Label
4-77	Real

3. KDD 2008 Cup data set

Download from: *http://www.sigkdd.org/kddcup/index.php?section=2008&method=data*

Table A.3: Data dictionary of KDD 2008 Cup data set

Attribut No.	Data Type
1-117	Real
118	Class Label

4. CoverType data set

Download from: *http://archive.ics.uci.edu/ml/datasets/Covertype*

Table A.4: Data dictionary of CoverType data set

Attribut No.	Data Type
1-53	Integer
54	Class Label

5. Zoo **data set**

Download from: *http://archive.ics.uci.edu/ml/datasets/Zoo*

Table A.5: Data dictionary of Zoo data set

Attribut No.	Data Type
1	Unique
2-13	Binary
14	Integer
15-17	Binary
18	Class Label

6. Arcene data set

Download from: *http://archive.ics.uci.edu/ml/datasets/Arcene*

Table A.5: Data dictionary of Arcene data set

Attribut No.	Data Type
1-10000	Real
10001	Class Label

APPENDIX B

t Distribution Table

t Table

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.000	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.000	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	3.232	3.460
80	0.000	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	3.195	3.416
100	0.000	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.174	3.390
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

APPENDIX C

Publications

International Conference

1. A. Suebsing and N. Hiransakolwong, "Feature selection using euclidean distance and cosine similarity for intrusion detection model", in Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on, pp. 86-91, 2009.
2. A. Suebsing and N. Hiransakolwong, "Euclidean-based feature selection for network intrusion detection", Proc. International Conference on Machine Learning and Computing, pp. 222-229, 2009.

International Journal

1. A. Suebsing and N. Hiransakolwong, "A Novel Technique for Feature Subset Selection Based on Cosine Similarity", Applied Mathematical Sciences, Vol. 6, 2012, no. 133, pp. 6627 – 6655, 2012.

2009 First Asian Conference on Intelligent Information and Database Systems

Feature Selection Using Euclidean Distance and Cosine Similarity for Intrusion Detection Model

Anirut Suebsing
Department of Mathematics and Computer Science
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
s9062952@kmitl.ac.th

Nualsawat Hiransakolwong
Department of Mathematics and Computer Science
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
khnualsa@kmitl.ac.th

***Abstract*—Nowadays, data mining plays an important role in many sciences, including intrusion detection system (IDS). However, one of the essential steps of data mining is feature selection, because feature selection can help improve the efficiency of prediction rate. The previous researches, selecting features in the raw data, are difficult to implement. This paper proposes feature selection based on Euclidean Distance and Cosine Similarity which ease to implement. The experiment results show that the proposed approach can select a robust feature subset to build models for detecting known and unknown attack patterns of computer network connections. This proposed approach can improve the performance of a true positive intrusion detection rate.**

Keywords-Intrusion Detection System (IDS); Feature Selection; Data Mining; Cosine Similarity and Euclidean Distance

I. INTRODUCTION

The Internet and local area networks are growing larger in recent years. As about 1,463,632,361 people all over the world use the Internet, they are unconsciously encountering the number of security threats such as viruses, worms and attacks from hackers [1, 2]. Now firewalls, anti-virus software, message encryption, secured network protocols, password protection and so on are not sufficient to guarantee the security in computer networking, which some intrusions take advantages of weaknesses in computer systems. Therefore, intrusion detection is becoming a more and more important technology which monitors network traffic and identifies network intrusion such as anomalous network behaviors, unauthorized network access, and malicious attacks to computer systems [3].

The techniques of intrusion detection can be categorized into two categories [4]: anomaly detection and misuse detection. Anomaly detection identifies deviations from normal network behaviors and alert for potential unknown attacks, and misuse detection (signature-based detection) detects intruders with known patterns.

Lately, data mining are introduced to help IDS to detect intruders correctly [5, 2], and accordingly IDSs have shown to be successful in detecting known attacks. On the contrary, many unknown attacks IDSs still undergo from false negative (FN: detect an attack as a normal connection), though some intrusion experts believe that most novel attacks can be adequate to catch by using a signature of known attacks [6].

Although data mining can help IDS to detect correctly intruders, data mining relies on feature selection which is one of the important procedures of data mining. Feature selection is the technique, commonly used in machine learning, of selecting a subset of essential features for building robust learning models.

The goal of this paper is to effectively utilize Euclidean Distance and Cosine Similarity for selecting essential feature subsets with smaller size and giving higher performance for intrusion detection.

This paper is organized as follows. A background of feature selection, following with the fields of intrusion detection, Euclidean Distance, Cosine Similarity, and C5.0 algorithm are addressed in Section II. In Section III, the data set is introduced. The proposed method is described in Section IV. In Section V, the experimental results are reported, and ending with the remarkable conclusions.

II. BACKGROUND

Technological innovations in computer have led to expediency and efficiency in capturing huge accumulations of data. The new challenge is to transform the enormous of data into useful knowledge for practical applications.

A preceding generic task in data mining is to extract outstanding features for decision making. This function can be broken into two groups: feature transformation and feature selection [7]. Feature transformation refers to the process of creating a new set of combined features, especially feature construction and feature extraction.

On the other hand, feature selection is different from feature transformation because it does not produce new variables. Feature selection also known as variable selection, feature reduction, attribute selection or variable subset selection, is a widely used dimensionality reduction technique, which many researches focus on machine learning and data mining and found applications in text classification, web mining, and so on. These essential feature subsets not only allow for faster building model by reducing the number of features, but also help remove irrelevant, redundant and noisy features. This allows for building simpler and more comprehensible classification

978-0-7695-3580-7/09 $25.00 © 2009 IEEE
DOI 10.1109/ACIIDS.2009.23

models which improves classification performance. Hence, selected essential attributes are a critical issue for competitive classifiers and for data reduction. In the meantime, feature weighting is a variance of feature selection. It involves assigning a real-valued weight to catch feature. The weight associates with a feature measures its relevance or significance in the classification task [8]. Feature selection algorithms typically fall into two categories; Feature Ranking and Subset Selection. Feature Ranking ranks the features by a metric and eliminates all features that do not achieve an adequate score. Subset selection searches the set of possible features for the optimal subset. Feature Ranking methods are based on statistics, information theory, or on some function of classification [9]. In statistics, the most popular form of feature selection is stepwise regression. It is a greedy algorithm that adds the best feature (or deletes the worst feature) at each round. The main control issue is deciding when to stop the algorithm. In machine learning, this is typically done by cross validation [10].

In this paper, Cosine Similarity and Euclidean Distance are proposed to select robust features which can improve performance for intrusion detection. Euclidean Distance is used to select features for building a model for the detection of known attacks. On the other hand, Cosine Similarity is used to select features for building a model for the detection of unknown attacks. Also, the C5.0 method is used for evaluation. The introduction for Intrusion Detection System is addressed in following section.

A. Intrusion Detection

Network based and Host based IDSs are mainly two main types of IDS being used now. Individual packets going through networks are analyzed in a network-based system in Network Intrusion Detection System (NIDS). The malevolent packets which might be passed by a firewall filtering rules can be detected by the NIDS. While in a Host based system, the IDS examines the activity on each individual computer or host [11]. The techniques of intrusion detection can be categorized into two categories [4]: anomaly detection and misuse detection.

Anomaly detection tries to determine whether deviation from established normal usage patterns can be flagged as intrusions [4]. Anomaly detection techniques is based on the assumption that misuse or intrusive behavior deviates from normal system procedure [12]. The advantage of anomaly detection is that it can detect attacks notwithstanding whether the attacks have been seen before. But the disadvantage of anomaly detection is ineffective in detecting insiders' attacks.

Misuse Detection or Signature-Based Intrusion Detection, traditional technique, employs patterns of known attacks or weak spots of the system to match and identify attacks [4]. This means that there are some ways to represent attacks in the form of a pattern or an attack signature so that even variations of the same attack can be detected. The major drawback of misuse detection is that it cannot predict new and unknown attacks and has high false alarm rate.

In the view of the fact that Intrusion Detection System has some faults, especially misuse detection that cannot detect unknown attacks. Data mining and intelligent computing techniques, such as Statistical approaches, expert system, Pattern matching, Artificial Neural Network, Support Vector Machines, Neuro-Fuzzy, and Genetic Algorithm are being used to avoid above shortcomings of intrusion detection.

B. Euclidean Distance

Euclidean Distance is the most common use of measurement for distance [13, 14]. In most cases when people said about distance, they will refer to Euclidean distance. Euclidean distance or simply 'distance' equals to the Equal 1. In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points. The Euclidean distance between two points $A = (x_1, x_2, x_3, \ldots, x_n)$ and $B = (y_1, y_2, y_3, \ldots, y_n)$ is defined as:

$$d(AB) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+(x_3-y_3)^2+\ldots+(x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \quad (1)$$

C. Cosine Similarity

A popular measure metric of similarity between two vectors of n dimensions is the cosine similarity metric [15]. Cosine similarity is used in many applications, such as text mining and information retrieval [16, 17]. Given two vectors of attributes, $A = \{x_1, x_2, \ldots, x_n\}$ and $B = \{y_1, y_2, \ldots, y_n\}$, the cosine similarity θ, is the measure metric of the angle between these two vectors defined as:

$$Sim(A,B) = Cos\,\theta = \frac{\vec{A}\cdot\vec{B}}{\left|\vec{A}\right|\left|\vec{B}\right|} = \frac{\sum_{i=1}^{n}(x_i \cdot y_i)}{\sqrt{\sum_{i=1}^{n} x_i^{\,2} \cdot \sum_{i=1}^{n} y_i^{\,2}}} \quad (2)$$

D. C5.0 Algorithm

Classification is an important technique in data mining. The decision tree is the most efficient approach to classification problems—Friedman 1997 [18]. The input to a classifier is a training set of records, each of which is attribute values tagged with a class label. A set of attribute values defines each record. A decision tree has the root and each internal node labeled with a question. The arcs emanating from each node represent each possible answer to the associated question. Each leaf node represents a predication of solution to the problem under consideration. C5.0, one of methods that be used to build a decision tree, is a commercial version of C4.5 now widely used in many data mining packages [18]. A C5.0 model works by splitting the sample based on the field that provides the maximum information gain. The Gain Ratio is defined as:

$$\text{Gain Ratio (D, S)} = \frac{Gain(D,S)}{H(\frac{|D_1|}{D},...,\frac{|D_S|}{D})} \quad (3)$$

, where D is a database state, $H(\cdot)$ finds the amount of order in that state, when the state is split into s new states, S $= \{D_1, D_2, \ldots, D_S\}$. C5.0 uses the larger than average information gain. This is to compensate for the fact that the Gain Ratio values is skewed toward splits where the size of one subset is close to that of the starting one. The algorithm C5.0 begins to split the sub sample which is defined by the first split, then divides it again by another different field. One repeats this step until the subsample cannot be split. It would re-examine the lower level split in the end, then remove any subsample that does not contribute significantly to the value of the model.

The method of C5.0 is very robust for handling missing data and in a large number of input fields [18]. Therefore, we select the method of C5.0 to evaluate our features.

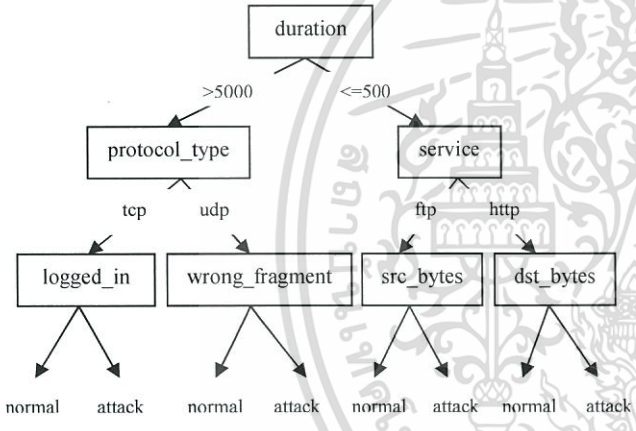


Figure 1. Basis structure of C5.0

III. INTRUSION DATA SET

In this paper, data set is the KDD Cup 1999 data set which was originally provided by MIT Lincoln Labs (The 1998 DARPA Intrusion Detection Evaluation Program) as the evaluation data set [2, 11, 12].

The data set was later prepared for KDD competition) (see http://www.ics.uci.edu/-kdd/databases/kddcup99/kddcup99.html" for more detail)

The data set is the real data which captured in the real network. It includes many kinds of attack data, also includes the normal data. The raw data was processed in to 39 attack types. These attacks are divided into four categories: probing (surveillance and other probing, e.g., port scanning), DoS (denial-of-service, e.g., syn flood), U2R (unauthorized access from a user to root privilege, e.g., various "buffer overflow" attacks) and R2L (unauthorized access from remote to local machine, e.g., guessing password). For each TCP/IP connection, 41 input features plus one class label were extracted in the data set belonging to each of four categories (9 basic Features, 13 Content Features, 9 Time-based Features and 10 Host-based Features) [12]. Table I summarizes a total of 22 training known attack types, with additional 17 unknown types.

TABLE I. DETAIL ATTACK TYPES

Class	Known attack	Unknown attack
Probe	ipsweep, nmap, portsweep, satan	saint, mscan
DoS	back, land, Neptune, pod, smurf, teardrop	apache2, processtable, udpstorm, mailbomb
U2R	buffer_overflow, loadmodule, perl, rootkit	xterm, ps, sqlattack
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster	snmpgetattack, named, xlock, xsnoop, sendmail, httptunnel, worm, snmpguess

In this paper, data set was samples only 10% of data sets which were randomly selected from 4,940,000 connections as training set. This training data set consists of 97,277 normal patterns plus 396,743 known attacks. Test data use 100% of test data sets of KDD Cup 1999 intrusion detection evaluation program.

IV. PROPOSED APPROACH

In this section, an approach, which is used to select robust features, is proposed. The proposed approach has two methods. First, known detection method is used to select features to build model for the detection of known attacks. Second, unknown detection method is used to select features to build model for the detection of unknown attacks. Note that the data which meets the goal of the proposed method must be numerical value. Thus, the symbolic data is transformed into numerical and make them under the same evaluation standard.

A. Known detection method

The proposed approach uses the Euclidean Distance from the Equation 1 to compute ranking score between each attribute and the class label by defining each attribute of KDD Cup 1999 training set, 41 attributes, as A_1, A_2, $A_3, \ldots, A_{41}$ respectively and the class label as B; moreover, let x is a value in any attribute and y is a values in the class label.

Let any $A_j = \{x_{1,j}, x_{2,j}, x_{3,j}, \ldots, x_{n,j}\}$ be a vector of attributes, where j $(1 \le j \le 41)$ is an ordinal number of attributes of training set, and also n $(n \ge 0)$ is the number of instances of training set.

Let $B = \{y_1, y_2, y_3, \ldots, y_n\}$ be a vector of class label, where n $(n \ge 0)$ is the number of instances of training set.

Thus, the ranking score of known detection method is $\{d_1(A_1,B), d_2(A_2,B), d_3(A_3,B), \ldots, d_{41}(A_{41},B)\}$, where any

$$d_j(A_j,B) = \sqrt{\sum_{i=1}^{n} (x_{i,j} - y_i)^2} \quad (4)$$

Where j ($1 \le j \le 41$) is an ordinal number of attributes of training set, and also n ($n \ge 0$) is the number of instances of training set.

After distance is computed, and then rank scores of known detection method of each attribute, $\{d(A_1,B), d(A_2,B), d(A_3,B), \ldots d(A_{41},B)\}$. Then, sort scores of the ranking scores from highest to lowest. Finally, select features that have high scores to build model, which is used to detect accurately known attacks.

B. *Unknown detection method*

To pick vigorous features for building model, which is used to detect unknown attacks, this paper uses the Cosine Similarity from the Equation 2 to compute ranking score between each attribute and the class label by representing each attribute of KDD cup 1999 training set as $A_1, A_2, A_3, \ldots, A_{41}$ respectively and the class label as B; moreover, let x is a value in any attribute and y is a values in the class label.

Let any $A_j = \{x_{1,j}, x_{2,j}, x_{3,j}, \ldots, x_{n,j}\}$ be a vector of attributes, where j ($1 \le j \le 41$) is an ordinal number of attributes of training set, and also n ($n \ge 0$) is the number of instances of training set.

Let $B = \{y_1, y_2, y_3, \ldots, y_n\}$ be a vector of attributes, where n ($n \ge 0$) is the number of instances of training set.

Thus, the ranking score of unknown detection method is $\{Sim_1(A_1,B), Sim_2(A_2,B), Sim_3(A_3,B), \ldots, Sim_{41}(A_{41},B)\}$, where any

$$Sim_j(A_j,B) = \frac{\sum_{i=1}^{n} (x_{i,j} \cdot y_i)}{\sqrt{\sum_{i=1}^{n} x_{i,j}^2} \cdot \sum_{i=1}^{n} y_i^2} \qquad (5)$$

Where j ($1 \le j \le 41$) is an ordinal number of attributes of training set, and also n ($n \ge 0$) is the number of instances of training set.

After compute the cosine similarity metric, and then rank scores of unknown detection method of each attribute, $\{Sim_1(A_1,B), Sim_2(A_2,B), Sim_3(A_3,B), \ldots, Sim_{41}(A_{41},B)\}$. Then, arrange score of the ranking score from highest to lowest. Finally, choose features that give high score to build a model.

	Attributes					Class Label
Instances	A_1	A_2	A_3	...	A_{41}	B
	$x_{1,1}$	$x_{1,1}$	$x_{1,1}$	...	$x_{1,41}$	y_1
	$x_{2,1}$	$x_{2,1}$	$x_{2,1}$	...	$x_{2,41}$	y_2
	$x_{3,1}$	$x_{3,1}$	$x_{3,1}$	...	$x_{3,41}$	y_3
	⋮	⋮	⋮	⋮	⋮	⋮
	$x_{n,j}$	$x_{n,j}$	$x_{n,j}$	...	$x_{n,41}$	y_n

Figure 2. Vectors of attributes and a vector of class label of known and unknown detection methods

V. EXPERIMENTS

In this section, investigate on the performance of the proposed approach, which consists of two methods (known detection method and unknown detection method). Data set was described in Section III and C5.0 in Section II-D is used to evaluate the proposed method.

Note that the measurement in this paper of the experimental results is based on the standard metrics for evaluations of intrusion: Detection rate for truth positive (TP) refers to the ratio between the number of correctly detected attacks and the total number of attacks while false positive (FP) rate means the ratio between the number of normal connections that are incorrectly misclassified as attacks and the total number of normal connections.

A. *Results*

Some important features of the known detection method are different from some important features of the unknown detection method. The known detection method in Section IV-A extracts 30 important features out of 41 features. The unknown detection method in Section IV-B extracts 24 important features out of 41 features. Table II (A) shows features selected by both methods (the known detection method and the unknown detection method), (B) shows the difference features between these two selection methods.

TABLE II. (A) FEATURES SELECTED BY BOTH METHODS (THE KNOWN DETECTION METHOD AND THE UNKNOWN DETECTION METHOD), (B) THE DIFFERENCE FEATURES BETWEEN THESE TWO SELECTION METHODS.

(A)

Known detection method	Unknown detection method
duration	
protocol_type	
logged_in	
serror_rate	
srv_serror_rate	
reerror_rate	
srv_rerror_rate	
diff_srv_rate	
srv_diff_host_rate	
dst_host_diff_srv_rate	
dst_host_srv_diff_host_rate	
dst_host_serror_rate	
dst_host_srv_serror_rate	
dst_host_rerror_rate	
dst_host_srv_rerror_rate	

(B)

Known detection method	Unknown detection method
src_bytes	service
dst_bytes	flag
land	count
wrong_fragment	serv_count
urgent	same_srv_rate
host	dst_host_count
num_failed_logins	dst_host_srv_count
num_compromised	dst_host_same_srv_rate
root_shell	dst_host_same_src_port_rate
su_attempted	
num_root	
num_file_creations	
num_shells	
num_access_files	
is_guest_login	

Table III demonstrates the detail classification results generated by C5.0 (in Section II-D) using 30 features of known detection method (in Section IV-A), which was computed with Euclidean Distance (in Section II-B), to detect known attack connection and using 24 features of unknown detection method (in Section IV.B), which was computed with Cosine Similarity (in Section II-C), to detect unknown attack connection. For the known attack test set, the known detection method predicts virtually perfectly. On the other hand, for the unknown attack test set, the unknown detection method did not show impressive detection rate.

TABLE III. CONFUSION MATRIX OF THE PROPOSED APPROACH

Known attack (using known detection method)

Class	Attack	Normal	TP %
Attack	226,890	4,817	97.9
Normal	675	59,916	98.9

Unknown attack (using unknown detection method)

Class	Attack	Normal	TP %
Attack	6,869	11,860	36.67
Normal	13	18,687	99.93

TABLE IV. PERFORMANCE SUMMARY

	Known attack		Unknown attack	
	Full Set (41)	*Known detection method (30)*	*Full Set (41)*	*Unknown detection method (24)*
Overall TP %	97.95	98.12	53.31	68.28
Overall FP %	2.04	1.87	46.69	31.72

The number in the parenthesis denotes the number of features used.

TABLE V. TIME TAKEN TO BUILD MODELS

	Full Set	Known detection method	Unknown detection method
Second(s)	75	51	45

Table IV shows overall accuracy of C5.0 using the proposed approach features comparing with the performance of C5.0 using full features (using 41 features). The overall detection rate of the proposed approach was the best in both test sets (known attack test set and unknown attack test set), especially in unknown attack test set. Furthermore, the Table V shows the time taken to build models.

VI. CONCLUSION

The proposed approach shows that the feature selection method computed with the Euclidean Distance can select the robust features to build model for the detection of known patterns while the feature selection method computed with the Cosine Similarity can select the robust features to build model for the detection of unknown patterns. The experimental results show that the known detection method based on Euclidean Distance and the unknown detection method based on Cosine Similarity are very promising over the known and unknown attack patterns respectively. In addition, both the methods produced smaller features showing other advantages because in the real-world applications, the smaller features are always advantageous in terms of both data management and verification of the prediction model, as long as the proposed approach gives more accuracy and processes faster with an efficient performance.

The future work, the following directions are proposed: 1) setting the threshold by the system instead of human, 2) generating robust features to build model that can detect unknown patterns correctly and 3) developing a real-time intrusion detection model.

REFERENCES

[1] S. Hansman and R. Hunt, "A Taxonomy of network and computer attacks," Computers & Security, 2005, 24, 31-43.

[2] C. H. Lee, S. W. Shin, and J. W. Chung, "Network Intrusion Detection Through Genetic Feature Selection," Proceeding of the Seventh ACIS International Conference on Software Engineering, Artificial Interlligence, Networking, and Parallel/Distributed Computing (SNPD'06), 2006.

[3] R. H. Gong, M. Zulkernine, and P. Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection," Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05), 2005.

[4] Y. Bai and H. Kobayashi, "Intrusion Detection Systems: Technology and Development," Proceeding of the 17th International Conference on Advanced Information Networking and Applications (AINA'03), 2003.

[5] J. S. Han and B. Cho, "Detecting intrusion with rule-based integration of multiple models," Computer & Security, 2003, 22,613-623.

[6] S. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project," DARPA Information Survivability Conference, 2000.

[7] S. Mukkamala and A. H. Sung, "A comparative study of techniques for intrusion detection," Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03), 2003.

[8] G. John, R. Kohavi, and Pfleger, "Irrelevant features and the subset selection problem," Int. Conf. on Machine Learning, Morgan Kaufman, San Francisco, 1994, 121-129.

[9] W. Duch, T. Winiarski, J. Biesiada, and A. Kachel, "Feature Ranking Selection and Discretization," Int. Conf. on Artificial Neural Networks (ICANN) and Int. Conf. on Neural Information Processing (ICONIP), Istanbul, June 2003, pp, 251-254.

[10] K. Ron, "A study of cross-validation and bootstrap for accuracy estimation and model selection," Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1995, 2, 1137-1143.

[11] W. Hu, J. Li, and J. Shi, "Optimal Evaluation of Feature Selection in Intrusion Detection Modeling," Proceeding of the 6th world congress on Intelligent Control and Automation, Dalian, China, June 21- 23 2006.

[12] W. Xuren, H. Famei, and X. Rongsheng, "Modeling Intrusion Detection System by Discovering Association Rule in Rough Set Theory Framework," International Conference on Computational Intelligence for Modelling Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), 2006.
[13] http://planetmath.org/encyclopedia/EuclideanDistance.html
[14] http://people.revoledu.com/kardi/tutorial/similarity/EuclideanDistance.html
[15] A. Karnik, S. Goswami. and R. Guha, "Detecting Obfuscated Viruses Using Cosine Similarity Analysis," Proceedings the First Asia International Conference on Modelling & Simulation (AMS'07), 2007.
[16] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," KDD workshop on text mining, 2000.
[17] http://www.stanford.edu/class/cs276/handouts/lecture13-vector-classify.ppt
[18] R. Yeh, C. Liu, B. Shla, Y. Cheng, and Y. Huwang, "Imputing manufacturing material in data mining," Springer Science+Business Media, LLC, 2007.

2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Euclidean-based Feature Selection for Network Intrusion Detection

Anirut Suebsing, Nualsawat Hiransakolwong

Department of Mathematics and Computer Science
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

Abstract. Nowadays, data mining has been playing an important role in the various disciplines of sciences and technologies. For computer security, data mining are introduced for helping intrusion detection System (IDS) to detect intruders correctly. However, one of the essential procedures of data mining is feature selection, which is the technique (commonly used in machine learning) for selecting a subset of relevant features for building robust learning models, due to the fact that feature selection can help enhance the efficiency of prediction rate. In the previous researches on feature selection, the criteria and way about how to select the features in the raw data are mostly difficult to implement. Therefore, this paper presents the easy and novel method, for feature selection, which can be used to separate correctly between normal and attack patterns of computer network connections. The goal in this paper is to effectively apply Euclidean Distance for selecting a subset of robust features using smaller storage space and getting higher Intrusion detection performance. During the evaluation phase, three different test data sets are used to evaluate the performance of proposed approach with C5.0 classifier. Experimental results show that the proposed approach based on the Euclidean Distance can improve the performance of a true positive intrusion detection rate especially for detecting known attack patterns.

Keywords: Intrusion Detection System (IDS), Feature Selection, Data Mining, Euclidean Distance, C5.0

1. Introduction

The internet and local area networks are growing larger in recent years. As a great variety of people all over the world are connecting to the Internet, they are unconsciously encountering the number of security threats such as viruses, worms and attacks from hackers [1, 2]. Now firewalls, anti-virus software, message encryption, secured network protocols, password protection and so on are not sufficient to assure the security in computer networks, which some intrusions take advantages of weaknesses in computer systems to threaten. Therefore, intrusion detection is becoming a more and more important technology which follows up network traffic and identifies network intrusion such as anomalous network behaviors, unauthorized network access, and malicious attacks to computer systems [3].

The techniques of intrusion detection can be categorized into two categories [4]: anomaly detection and misuse detection. Anomaly detection identifies deviations from normal network behaviors and alert for potential unknown attacks, and misuse detection (signature-based detection) detects intruders with known patterns.

In the last, data mining are introduced for helping IDS to detect intruders correctly [5, 2], and accordingly IDSs have shown to be successful in detecting known attacks. On the contrary, many unknown attacks IDSs still undergo from false positive (FP: detect a normal as an attack connection), also known as a false detection or false alarm. Though some intrusion experts believe that most novel attacks can be adequate to catch by using a signature of known attacks [6]. Although data mining can help IDS to detect correctly intruders, data mining relies on feature selection which is one of the important procedures of data mining.

+ Corresponding author. Tel.: +6623267439.
E-mail address: s9062952@kmitl.ac.th

Feature selection is intended to suggest which features are more important for the prediction, to find out and get rid of irrelevant features that reduce classification accuracy, discover relations between features and throw out highly correlated features which are redundant for prediction.

The goal in this paper is to effectively apply Euclidean Distance to select better feature subsets with using smaller storage space and getting higher Intrusion detection performance.

The paper is organized as follows: In Section 2, a background of feature selection is addressed, following with the fields of intrusion detection, Euclidean Distance and C5.0 algorithm. In Section 3, the data set used in this paper is addressed. The proposed method is described in Section 4. In Section 5, the experimental results are reported, and the remarkable conclusions are addressed in the final Section.

2. Background

The rapid developments in computer science and engineering have led to expediency and efficiency in capturing huge accumulations of data. The new challenge is to transform the enormous of data into useful knowledge for practical applications.

An earlier general task in data mining is to extract outstanding features for the prediction. This function can be broken into two groups—feature extraction or feature transformation, and feature selection [7]. Feature extraction (for example, principal component analysis, singular-value decomposition, manifold learning, and factor analysis) refers to the process of creating a new set of combined features (which are combinations of the original features).

On the other hand, feature selection is different from feature extraction because it does not produce new variables. Feature selection also known as variable selection, feature reduction, attribute selection or variable subset selection, is a widely used dimensionality reduction technique, which has been the focus of much research in machine learning and data mining and found applications in text classification, web mining, and so on. It allows for faster model building by reducing the number of features, and also helps remove irrelevant, redundant and noisy features. This allows for building simpler and more comprehensible classification models with classification performance. Hence, selecting relevant attributes are a critical issue for competitive classifiers and for data reduction. In the meantime, feature weighting is a variant of feature selection. It involves assigning a real-valued weight to catch feature. The weight associated with a feature measures its relevance or significance in the classification task [8]. Feature selection algorithms typically fall into two categories; Feature Ranking and Subset Selection. Feature Ranking ranks the features by a metric and eliminates all features that do not achieve an adequate score (selecting only important features). Subset selection searches the set of possible features for the optimal subset. Feature Ranking methods are based on statistics, information theory, or on some function of classifier's outputs [9]. In statistics, the most popular form of feature selection is stepwise regression. It is a greedy algorithm that adds the best feature (or deletes the worst feature) at each round. The main control issue is deciding when to stop the algorithm. In machine learning, this is typically done by cross validation [10].

In this paper, we adapt Euclidean Distance to select robust features which can bring to a successful conclusion of intrusion detection. Euclidean Distance is used to select features to build model for the detection of known and unknown attacks. And also, method of C5.0 is used to evaluation in this paper. Note that our proposed approach is categorized as the feature ranking selection. The following section is the introduction to intrusion detection.

2.1. Intrusion Detection

Network based and Host based IDSs are mainly two main types of IDS being used now. Individual packets going through networks are analyzed in a network-based system in NIDS. The malevolent packets which might be passed by a firewall filtering rules can be detected by the NIDS. In a Host based system, the IDS examines the activity on each individual computer or host [11]. The techniques of intrusion detection can be grouped into two groups [4]: anomaly detection and misuse detection.

Anomaly detection [4] tries to determine whether deviation from established normal usage patterns can be flagged as intrusions. Anomaly detection techniques are based on the assumption that misuse or intrusive

behavior deviates from normal system procedure [12]. The advantage of anomaly detection is that it can detect attacks notwithstanding whether or not the attacks have been seen before. But the disadvantage of anomaly detection is ineffective in detecting insiders' attacks.

Misuse Detection or Signature-Based Intrusion Detection, traditional technique, [4] employs patterns of known attacks or weak spots of the system to match and identify attacks. This means that there are some ways to represent attacks in the form of a pattern or an attack signature so that even variations of the same attacks can be detected. The major drawback of misuse detection is that it cannot predict new and unknown attacks and has high false alarm rate.

In the view of the fact that Intrusion Detection System has some faults, especially misuse detection that cannot detect unknown attacks, intelligent computing techniques, such as statistical approaches, expert system, pattern matching, Artificial Neural Network, Support Vector Machines, Neuro-Fuzzy, Genetic Algorithm with above techniques and data mining, are being used to avoid above shortcomings of intrusion detection.

2.2. Euclidean Distance

Euclidean Distance is the most common use of distance [13, 14, 15]. In most cases when people said about distance, they will refer to Euclidean distance. Euclidean distance or simply 'distance' examines the root of square differences between coordinates of a pair of objects. In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points. The Euclidean distance between two points $A = (x1, x2, x3, \ldots, x_n)$ and $B = (y1, y2, y3, \ldots, y_n)$ is defined as:

$$d(A,B) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + \ldots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

2.3. C 5.0 Algorithm

Classification is an important technique in data mining, and the decision tree is the most efficient approach to classification problems—Friedman 1997 [16]. The input to a classifier is a training set of records, each of which is a tuple of attribute values tagged with a class label. A set of attribute values defines each record. A decision tree has the root and each internal node labeled with a question. The arcs emanating from each node represent each possible answer to the associated question. Each leaf node represents a predication of solution to the problem under consideration. C5.0, one of methods that be used to build a decision tree, is a commercial version of C4.5.

A C5.0 model is based on the information theory [17, 18]. Decision trees are built by calculating the information gain ratio. The algorithm C5.0 works by separating the sample into subsamples based on the result of a test on the value of a single feature. The specific test is selected by an information theoretic heuristic. This procedure is iterated on each of the new subsample and keeps on until a subsample cannot be separated or the partitioning tree has reached the threshold. The information gain ratio is defined as:

$$\text{Information Gain Ratio (D, S)} = \frac{Gain(D,S)}{H(\frac{|D_1|}{D},\ldots,\frac{|D_S|}{D})} \quad (2)$$

, where D is a database state, H $(\cdot)$ finds the amount of order in that state, when the state is separated into S new states $S = \{D_1, D_2, \ldots, D_S\}$.

The method of C5.0 is very robust for handling missing data and in a large number of input fields [16]; therefore, C5.0 is used to evaluate our features in this paper.

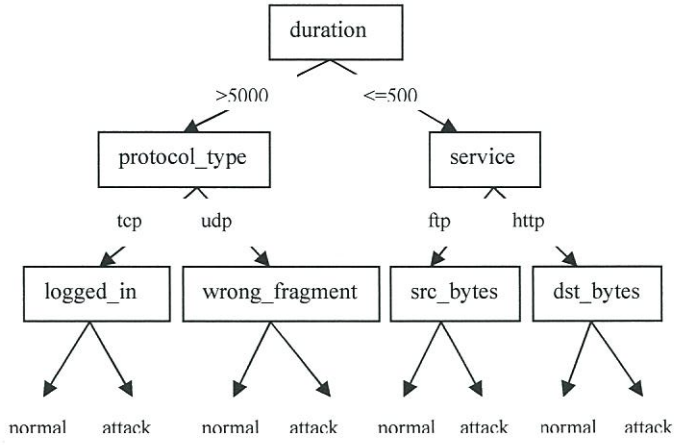


Fig. 1: Basis structure of C5.0

3. Intrusion Data set

In this paper, we choose the KDD Cup 1999 data set which was originally provided by MIT Lincoln Labs (The 1998 DARPA Intrusion Detection Evaluation Program) as the evaluation data set [2, 11, 12]. The data set was later prepared for KDD competition) (see "http://www.ics.uci.edu/~kdd/databases/kddcup99/kddcup99.html" for more detail)

The data set is the real data which captured in the real network. It includes many kinds of attack data, also includes the normal data. The raw data was processed in to 39 attack types. These attacks are divided into four categories: probing (surveillance and other probing, e.g., port scanning), DoS (denial-of-service, e.g., SYN flood), U2R (unauthorized access from a user to root privilege, e.g., various "buffer overflow" attacks) and R2L (unauthorized access from remote to local machine, e.g., guessing password). For each TCP/IP connection, 41 input features plus one class label were extracted in the data set belonging to four kinds (9 basic Features, 13 Content Features, 9 Time-based Features and 10 Host-based Features) [12]. In Table 1, a total of 22 training known attack types, and additional 17 unknown types are summarized.

Table 1: Detail attack types [2]

Class	Known attack	Unknown attack
Probe	ipsweep, nmap, portsweep, satan	saint, mscan
DoS	back, land, Neptune, pod, smurf, teardrop	apache2, processtable, udpstorm, mailbomb
U2R	buffer_overflow, loadmodule, perl, rootkit	xterm, ps, sqlattack
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster	snmpgetattack, named, xlock, xsnoop, sendmail, httptunnel, worm, snmpguess

In this study, Training data set in the paper contained 49, 451 records, which were randomly generated from the KDD Cup 1999 for 10% training data set that consists of 9, 768 normal patterns, 39, 085 known DoS patterns, 435 known Probe patterns, 111 known R2L patterns and 52 known U2R patterns.

Test data set in the paper composed of three different test data sets, which were randomly selected from the KDD Cup 1999, 100% test data set. Table 2 gives the number of records on three different test data sets

Table 2: The number of records on three different test data sets

Dataset Name	Known attack	Unknown attack
Dataset-1	186, 745	19, 820
Dataset-2	49, 438	14, 781
Dataset-3	25, 419	10, 031

4. Proposed Approach

The proposed approach is used to select robust features to build model for the detection of known and unknown attacks. Note that the data which meets the demands of proposed methods must be numerical value. Therefore, the symbolic data should be transformed into numerical and make them under the same evaluation standard.

In proposed approach, the Euclidean Distance from equation (1) is used to compute ranking score between each attribute and class label by defining each attribute of KDD Cup 1999 training set, 41 attributes, as $A_1, A_2, A_3, \ldots, A_{41}$ respectively and class label as B; moreover, let x is a value in any attribute and y is a values in class label.

Let any $A_j = \{x_{1,j}, x_{2,j}, x_{3,j}, \ldots, x_{n,j}\}$ be a vector of attributes, where j ($1 \le j \le 41$) is an ordinal number of attributes of training set, and also n ($n \ge 0$) is the number of instances of training set.

Let $B = \{y_1, y_2, y_3, \ldots, y_n\}$ be a vector of class label, where n ($n \ge 0$) is the number of instances of training set.

Thus, the ranking score is $\{d_1(A_1, B), d_2(A_2, B), d_3(A_3, B), \ldots, d_{41}(A_{41}, B)\}$, where any

$$d_j(A_j, B) = \sqrt{\sum_{i=1}^{n} (x_{i,j} - y_i)^2} \tag{3}$$

where j ($1 \le j \le 41$) is an ordinal number of attributes of training set, and also n ($n \ge 0$) is the number of instances of training set.

After computing distance measure, the distance is score of known detection method of each attribute, $\{d(A_1, B), d(A_2, B), d(A_3, B), \ldots d(A_{41}, B)\}$. Then, sort scores of the ranking score from highest to lowest. Finally select features that have high scores to build model, which is used to detect accurately known and unknown attacks. The method of C5.0 is used to evaluate features that are taken from last step.

	Attributes					Class Label
Instances	A_1	A_2	A_3	...	A_{41}	B
	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,41}$	y_1
	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{2,41}$	y_2
	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,41}$	y_3
	⋮	⋮	⋮	⋮	⋮	⋮
	$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	...	$x_{n,41}$	y_n

Fig. 2: Vectors of each attributes and a vector of class label

5. Experiments

In this section, an investigation on the performance of proposed feature selector based Euclidean is studied. The data sets (one train set and three test sets) described in Section 3 and C5.0 described in Section 2.4 are used to evaluate the proposed approach.

Note that the measurement in this paper of the experimental results is based on the standard metrics for evaluations of intrusion, Detection rate (TP) refers to the ratio between the number of correctly detected attacks and the total number of attacks while false alarm rate (FP: false positive) means the ratio between the number of normal connections that are incorrectly misclassified as attacks and the total number of normal connections.

From Table 3, the different between scores 441.72 and 360.65 is the highest value. Therefore, features that have scored more than four hundred scores are selected, getting 30 important features out of 41 features that show in Table 4. From Fig. 3, the proposed approach shows impressive detection rate of known attack for normal, DoS and Probe while the proposed approach does not demonstrates impressive detection rate for R2L and U2L. Maybe the number of records of R2L and U2L is 52 from 5 million records in the dataset. It is quite small. Moreover, the *warezclient* attack belonging to R2L is the majority patterns in the training set. However, in the test set, *guess_passwd* and *warezmaster* comprises most patterns of R2L. On the other hand, form Fig. 4, the proposed approach does not shows efficiency when it is used to detect unknown attack, but it can detect unknown attack for normal, Probe and U2L especially normal. It can detect quite excellent. Fig 5

shows the overall detection rate of the proposed approach on three different test sets. Fig 6 shows overall false positive rate of the proposed approach on three different test sets.

Table 5 (overall accuracy of C5.0 using the proposed approach to select features based Euclidean) shows capability of the proposed approach when it is used to detect known attack although it cannot show impressive used to detect unknown attack when comparing with detection rate of known attack. Furthermore, Table 6 shows overall false positive rate of C5.0 using the proposed approach to select features based Euclidean. Results from table 6 demonstrate that the proposed approach has drawback when used to detect unknown attack since percentage of overall false positive rate (FP) of unknown attack is quite high even if it is not more than 50 percent.

Table 3: The ranking score computed by Euclidean (in Section 4)

Feature name	Scores
dst_host_srv_serror_rate	499.24
srv_serror_rate	499.23
serror_rate	499.04
dst_host_serror_rate	498.99
srv_rerror_rate	486.19
reerror_rate	486.07
dst_host_srv_rerror_rate	485.75
dst_host_rerror_rate	485.38
logged_in	484.72
root_shell	483.39
land	483.38
urgent	483.37
num_compromised	483.37
su_attempted	483.37
src_bytes	483.37
num_failed_logins	483.37
num_root	483.37
num_shells	483.36
num_file_creations	483.36
num_access_files	483.32
dst_bytes	483.30
is_guest_login	483.09
host	483.03
duration	483.01
wrong_fragment	482.46
dst_host_srv_diff_host_rate	481.21
srv_diff_host_rate	480.04
diff_srv_rate	476.68
dst_host_diff_srv_rate	474.17
protocol_type	441.72
service	360.65
dst_host_count	256.91
serv_count	225.08
dst_host_same_src_port_rate	224.81
same_srv_rate	223.86
dst_host_same_srv_rate	222.52
dst_host_srv_count	219.26
count	190.59
flag	152.15
num_outbound_cmds	NaN
is_host_login	NaN

Table 4: 30 features extracted by Euclidean

duration, protocol_type, logged_in, serror_rate, srv_serror_rate, reerror_rate, srv_rerror_rate, diff_srv_rate, srv_diff_host_rate, dst_host_diff_srv_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate, src_bytes, dst_bytes, land, wrong_fragment, urgent, host, num_failed_logins, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, is_guest_login

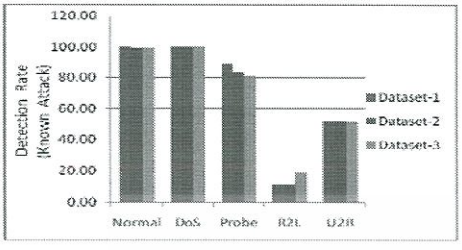


Fig. 3: Detection rate of known attack on three different test sets

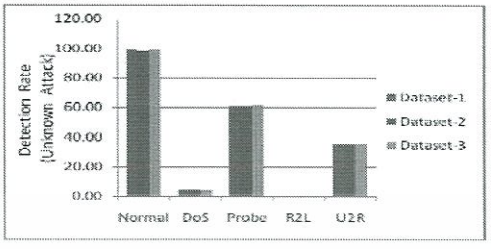


Fig. 4: Detection rate of unknown attack on three different test sets

Table 5: Overall detection rate of the proposed approach on three different test sets

Dataset Name	Known attack	Unknown Attack
Dataset-1	99.77%	56.52%
Dataset-2	99.32%	56.45%
Dataset-3	99.21%	56.59%

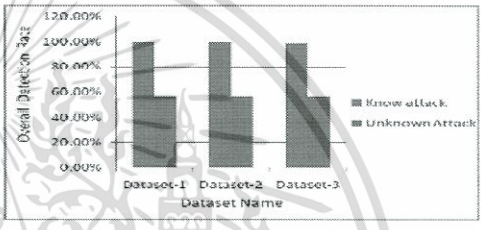


Fig. 5: Overall detection rate of the proposed approach on three different test sets

Table 6: Overall false positive rate of the proposed approach on three different test sets

Dataset Name	Known attack	Unknown Attack
Dataset-1	0.23%	43.48%
Dataset-2	0.68%	43.55%
Dataset-3	0.79%	43.41%

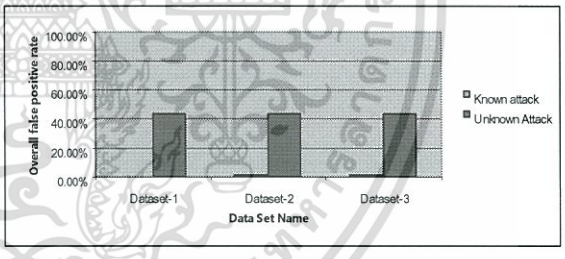


Fig. 6: Overall false positive rate of the proposed approach on three different test sets

6. The remarkable Conclusions

The proposed approach presented in this paper show that the feature selection method applied Euclidean Distance can extract the robust features to build model for the detection of known and unknown patterns, especially known patterns.

From the experimental results obtained, it is evident that the Euclidean-based feature selection is very promising over the known attack patterns. In addition, it produced smaller features showing other advantages because, in the real-world applications, the smaller features are always advantageous in terms of both data management and reduce the computing time. Therefore, the proposed approach can select a subset of robust features using smaller storage space and getting higher Intrusion detection performance, improving the performance of a true positive intrusion detection rate especially for detecting known attack patterns.

For the future work, the following directions are proposed: (1) setting the threshold by the automatic system and also (2) generating robust features to build model that can detect unknown patterns correctly.

7. Acknowledgements

We would like to esteem thanks Salvatore J. Stolfo, MIT Lincon Lab, and UCI KDD group for the KDDCUP 1999 data set.

8. References

[1] S. Hansman and R. Hunt. A Taxonomy of network and computer attacks. Computers & Security. 2005, 24, 31-43.

[2] C. H. Lee, S. W. Shin, and J. W. Chung. Network Intrusion Detection Through Genetic Feature Selection. *Proceeding of the Seventh ACIS International Conference on Software Engineering, Artificial Interlligence, Networking, and Parallel/Distributed Computing (SNPD'06).* 2006.

[3] R. H. Gong, M. Zulkernine, and P. Abolmaesumi. A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection. *Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05).* 2005.

[4] Y. Bai and H. Kobayashi. Intrusion Detection Systems: Technology and Development. *Proceeding of the 17th International Conference on Advanced Information Networking and Applications (AINA'03).* 2003.

[5] J. S. Han and B. Cho. Detecting intrusion with rule-based integration of multiple models. Computer & Security. 2003, 22, 613-623.

[6] S. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan. Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project. *DARPA Information Survivability Conference.* 2000.

[7] S. Mukkamala and A. H. Sung. A comparative study of techniques for intrusion detection. *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03).* 2003.

[8] G. John, R. Kohavi, and Pfleger. Irrelevant features and the subset selection problem. *Int. Conf. on Machine Learning, Morgan Kaufman, San Francisco.* 1994, 121-129.

[9] W. Duch, T. Winiarski, J. Biesiada, and A. Kachel. Feature Ranking Selection and Discretization. *Int. Conf. on Artificial Neural Networks (ICANN) and Int. Conf. on Neural Information Processing (ICONIP), Istanbul.* June 2003, pp, 251-254.

[10] K. Ron. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence.* 1995, 2, 1137-1143.

[11] W. Hu, J. Li, and J. Shi. Optimal Evaluation of Feature Selection in Intrusion Detection Modeling. *Proceeding of the 6th world congress on Intelligent Control and Automation, Dalian, China.* June 21- 23 2006.

[12] W. Xuren, H. Famei, and X. Rongsheng. Modeling Intrusion Detection System by Discovering Association Rule in Rough Set Theory Framework. *International Conference on Computational Intelligence for Modelling Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06).* 2006.

[13] http://www.itl.nist.gov/div897/sqg/dads/HTML/euclidndstnc.html

[14] http://people.revoledu.com/kardi/tutorial/similarity/EuclideanDistance.html

[15] A. Karnik, S. Goswami. and R. Guha. Detecting Obfuscated Viruses Using Cosine Similarity Analysis. *Proceedings of the First Asia International Conference on Modelling & Simulation (AMS'07).* 2007.

[16] R. Yeh, C. Liu, B. Shla, Y. Cheng, and Y. Huwang. Imputing manufacturing material in data mining. *Springer Science+Business Media, LLC.* 2007.

[17] C. Chan, Y. Liu, and S. Luo. Investigation of Diabetic Microvascular Complications Using Data Mining Techniques. *International Joint Conference on Neural Networks (IJCNN 2008).* 2008.

[18] J. Du and W. Guo. Data Mining on Patient Data. IEEE, 2005.

[19] A. Suebsing and N. Hiransakolwong. Feature Selection Using Euclidean Distance and Cosine Similarity for Intrusion Detection Model. *Asian Conference on Intelligent Information and Database Systems (ACIIDS 09).* 2009.

Applied Mathematical Sciences, Vol. 6, 2012, no. 133, 6627 - 6655

A Novel Technique for Feature Subset Selection Based on Cosine Similarity

Anirut Suebsing

Department of Computer Science, Faculty of Science
King Mongkut's Institute of Technology
Ladkrabang, Bangkok, 10520, Thailand
s9062952@kmitl.ac.th

Nualsawat Hiransakolwong

Department of Computer Science, Faculty of Science
King Mongkut's Institute of Technology
Ladkrabang, Bangkok, 10520, Thailand

Abstract

Nowadays, data mining has been playing an important role in the various disciplines of sciences and technologies. Data mining is composed of many tasks but one of the essential procedures of data mining is feature selection, which is the technique mostly based on the machine learning for selecting a subset consisted of significant features, building a stronger learning model, and enhancing the efficiency of prediction rate. Normally, a processing of building a learning model from the huge amount of collected data needs high computation cost. Therefore, with feature selection, the computation cost can be reduced by selecting relevant features. In the previous researches on feature selection, the criteria and algorithms for selecting the features from the raw data are mostly complicated and difficult to implement. Therefore, this paper presents a novel method by applied Cosine similarity to feature selection method. The proposed algorithm begins with selecting a robust feature subset using the Cosine similarity. This method is the simple algorithm using smaller storage space, reducing computation time and gaining higher predictive performance. During the evaluation phase, the ten data sets from UCI benchmark data sets are used to evaluate the performance of proposed approach by using the C5.0, CART and Neural Networks classifiers. Experimental results show that the method based on the Cosine similarity can improve the performance of accuracy detection rate with less error rate.

Keywords: Selection, Cosine similarity, Classification, Accuracy detection rate

1 Introduction

With the technological evolution in 21st century, the amount of information that can be gathered and stored increases very rapidly every day. The new challenge is how to exploit the ocean of data or how to transform the enormous data into useful knowledge for practical applications. So at the present time, data mining plays an important role in this issue. An earlier general task in data mining is to extract outstanding features to avoid high computation costs for classification task. This function can be broken into two groups: feature transformation, and feature selection [1], [2]. Feature transformation (for example, principal component analysis, singular-value decomposition, manifold learning, and factor analysis) refers to the process of creating a new set of combined features (which are combinations of the original features).

On the other hand, feature selection is different from feature transformation because it does not produce new variables but the method selects a subset of original attributes, filters out trivial attributes. Therefore feature selection reduces the feature space. Feature selection allows for faster model building by reducing the number of features, and also helps remove irrelevant, redundant and noisy features. This allows for building simpler and more comprehensible classification models with enhancing classification performance. Hence, selecting relevant attributes are a critical issue for competitive classifiers and for data reduction. Feature selection, sometimes, knows as feature weighting. Feature weighting assigns a real-valued weight to each selected feature. The weight associated with a feature measures its relevance or significance in the classification task [1]-[4].

Therefore, Feature selection plays an important role in data mining. The selected features will form the smallest size of data set to enable an efficient result. Hence, Automated methods for feature subset selection are often developed and used for searching an appropriate feature subset containing relevant features because the number of possible feature subsets in each data set is $2^N - 1$ subsets for N features [4]; Thus, it is impossible to search for the robust feature subset manually even after cleaning the data. Moreover, there are many reasons for using feature selection concluded as follows [2]-[4]:

1. Getting the maximizing accuracy of the classifier
2. Enhancing accuracy by reducing irrelevant and redundant features
3. Reducing the complexity and computational cost

Feature selection algorithms typically fall into two categories [1]-[6]; filter and wrapper approach. Filter approach filters irrelevant features out keeping a good feature set before learning process [5]. On the other hand, wrapper approach

searches for a good feature set using a learning algorithm. Utilizing filter approach to generate a feature set is generally faster than wrapper approach because filter approach uses heuristics based on general characteristics of the data rather than wrapping a learning algorithm into the selection process to evaluate the merit of feature subsets [6].

In this paper, the proposed approach is a novel method to select or extract significant features for the classification task in data mining. The method is less complex than the other techniques, especially machine learning techniques, since those techniques are more complicated and more difficult to understand and to implement into the real-world application as well. Moreover, those methods rely on a class label or class attribute also. The proposed approach is based on hypothesis as follows: (1) each feature in a group should be relevant with other features in a group and (2) any feature will be determined as a significant feature if it is relevant to other features in a group at least once. Therefore, the method that can help us calculate a relevant value of each feature is led to be used. However, in order to select relevant features, the Cosine similarity is used to compute a relevant value of each feature in this paper because the Cosine similarity method is so well-known in information retrieval.

The objective in this paper is to create a new feature selection method by applying the Cosine similarity then adding with a new filtering feature technique for extracting a robust feature set with using smaller storage space, using automatic thresholds, using less complicated method, getting higher detection performance and avoiding high computational costs but it does not depend on the class label.

The rest of this paper is organized as follows. A brief review of related work is addressed in the next Section. In Section 3, a new approach is presented to select a better feature subset used for building a predicting model in the classification task. In Section 4, the experimental result is presented following with the discussion on the results. In Section 5, the remarkable conclusions are presented.

2 Related Work

This Section provides the feature selection algorithms in Section 2.1. Classification Algorithms is Section 2.2. In Section 2.3, it mentions C5.0 Algorithm. Section 2.4 presents CART Algorithm. Section 2.5 proposes Neural Networks and in Section 2.6, it presents Cosine Similarity.

2.1 Feature Selection Algorithms

Feature selection is a method which only the relevant features will be selected, discarding the irrelevant or weak features in the data set. Minimum set of features, which is close enough to represent the original data set, will be selected. The selected features will form the smallest size of data set to enable an efficient result.

Moreover, the basic process of feature selection method shows in Fig. 1 [7].

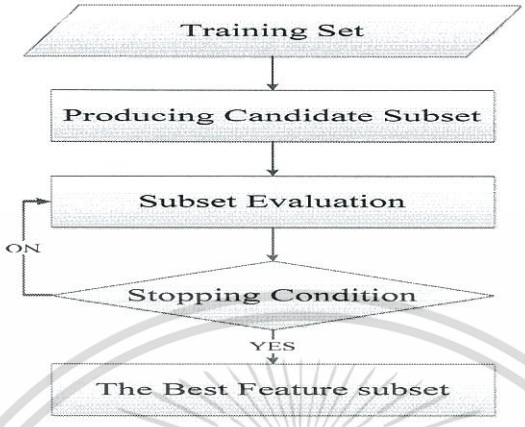


Fig. 1 The process of feature selection.

Feature selection approaches can be divided into two types: the filter approach and the wrapper approach.

• The Filter Approach: The Filter algorithms usually based on statistics consider the features relevance with the classifiers that use them. Statistical and Information theoretic measures such as information gain, Cross-entropy, Pearson's Chi-Square and so on are used to find the relationship of each feature in a data set with the target feature or class label assuming conditional independence with all other features. The robust subset of features selected from high rank feature. Ranking a list of features, which are ordered according to evaluation measures. The measure can be any of accuracy, consistency, information, distance, and relevance [5]. Fig. 2 shows the Filter approach flowchart.

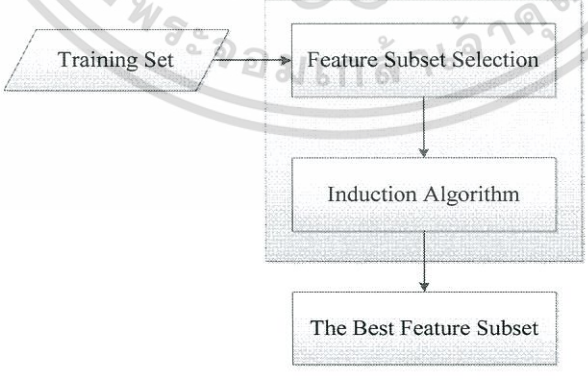


Fig. 2 The filter approach flowchart.

• The Wrapper Approach: Machine learning algorithms play an important role in this approach because they are used as evaluation function. The Wrapper

algorithms usually provide better accuracy but they are more complex and use more computation cost. These algorithms typically start from an empty list of features and add relevant features discovered [6]. The wrapper approach flowchart is shown in Fig. 3.

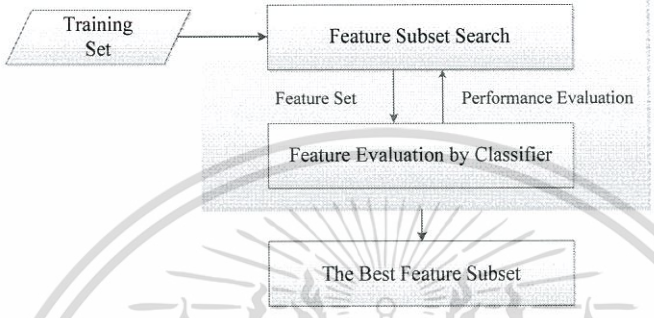


Fig. 3 The wrapper approach flowchart

2.2 Feature Selection Algorithms

Classification [8]-[11] is one of the most popular data mining techniques. Examples of classification applications based on classification include pattern recognition, medical diagnosis, detecting faults in industry application, and classifying financial market trends. Classification is a process of learning a function mapping a data item into one of some predefined classes.

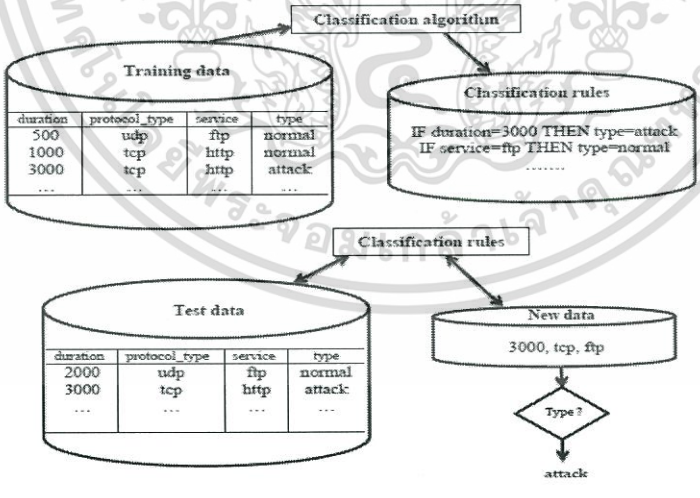


Fig.4 The data classification process

Every classification based on supervised learning is given as input a set of samples consisting of vectors of attribute values and a corresponding class. The input of a classification is a training set which each record consists of attributes and a class label. The target of classification is to build a classification model or

function called a classifier, which is used for predicting a class of objects whose class label is unknown. In other words, classification is the process of finding a model which describes and distinguishes data classes in order to employ the model to detect class label. The built model may be represented in diverse forms such as IF-THEN rules, neural networks or decision trees.

A decision tree form is most useful in classification problems [11]. With this method a tree is built to model the classification process. There are two basic steps in the technique: building the tree and applying the tree to the database. The decision tree is a flow-chart-like tree structure, where a root and each internal node labeled with a question. Each branch from each node represents each possible outcome of the associated question. Each leaf node represents a predication of solution to the problem under consideration. Decision trees can easily be converted to classification rules. Although there are many tree algorithms adopted for generating decision trees, well-known tree algorithms [8], [9] used widely are ID3, CART (Classification and Regression Tree) and C5.0 algorithms.

2.3 C5.0 Algorithm

The C5.0 algorithm is a commercial version extended from C4.5 proposed by J.R. Quinlan [9], [10], [12]. Now it is widely used as the inductive learning tools in Clementine, Rule Quest and so on. C5.0 algorithm is the process of generating an initial decision tree from the set of training samples. As a result, the algorithm generates a classifier in the form of a decision tree; a structure with two types of nodes: a leaf, indicating a class, or a decision node that specifies some test to be carried out on a single-attribute value, with one branch and sub tree for each possible outcome of the test. A decision tree can be used to classify a new sample by starting at the root of the tree and moving through it until a leaf is encountered. At each non-leaf decision node, the outcome of features for the test at the node is determined and attention shifts to the root of the selected sub tree.

The C5.0 algorithm is based on the information theory [9], [10], [12]. Decision trees are built by calculating the information gain ratio. The C5.0 algorithm works by separating the sample into subsamples based on the result of a test on the value of a single feature. The specific test is selected by an information theoretic heuristic. This procedure is iterated on each of the new subsample and keeps on until a subsample cannot be separated or the partitioning tree has reached the threshold. The information gain ratio is defined as:

$$\text{Information Gain Ratio } (D, S) = \frac{Gain(D,S)}{H(\frac{|D_1|}{D},...,\frac{|D_S|}{D})} \qquad (1)$$

Where D is a database state, $H(\cdot)$ finds the amount of order in that state. The state is separated into new states $S = \{D_1, D_2, ..., D_S\}$.

Although the C5.0 algorithm developed from C4.5, in C5.0, several new techniques were introduced as follows:

- Speed—C5.0 is significantly faster than C4.5

- Memory Usage—C5.0 is more memory efficient than C4.5
- Boosting—C5.0 is more accurate than C4.5
- New Attributes—C5.0 supports dates, times, timestamps, ordered discrete attributes.
- Smaller Decision Trees—C5.0 gets similar results to C4.5 with considerably smaller decision trees.
- Weighting—C5.0 allows you to weight different attributes and misclassification types.
- Handling Data—C5.0 can automatically winnows the data to help reduce noise.

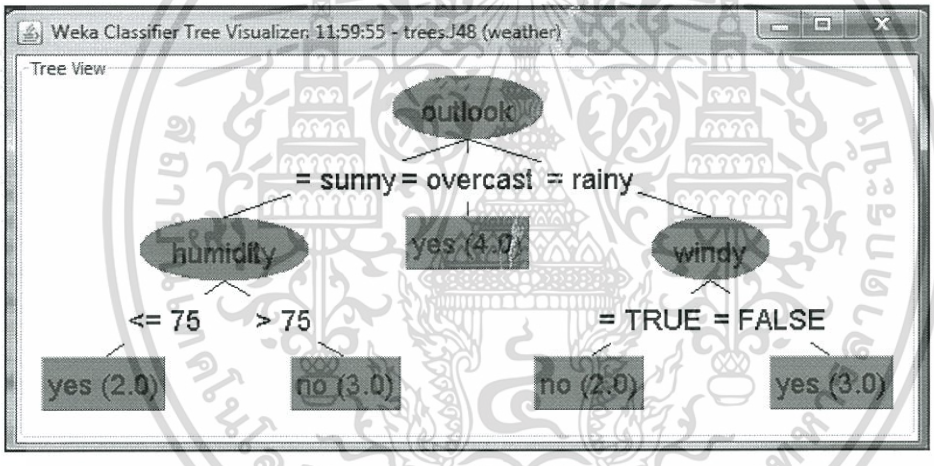


Fig. 5 A structure of decision tree model

2.4 CART Algorithm

The CART is a binary decision tree proposed by Breiman et al. [11], [13], [14]. The CART is constructed by feeding the attribute of feature vectors, and then a binary-branching tree from the root through iterative operations is built until it reaches a termination criterion. The two steps of CART are involved. First, the CART builds a tree structure by recursively partitioning training samples into different subclasses according to the selected test conditions until all samples are under the same subclass category. The tree structure is established. Second, the CART will prune the decision tree structure from the bottom of the tree until a stopping criterion.

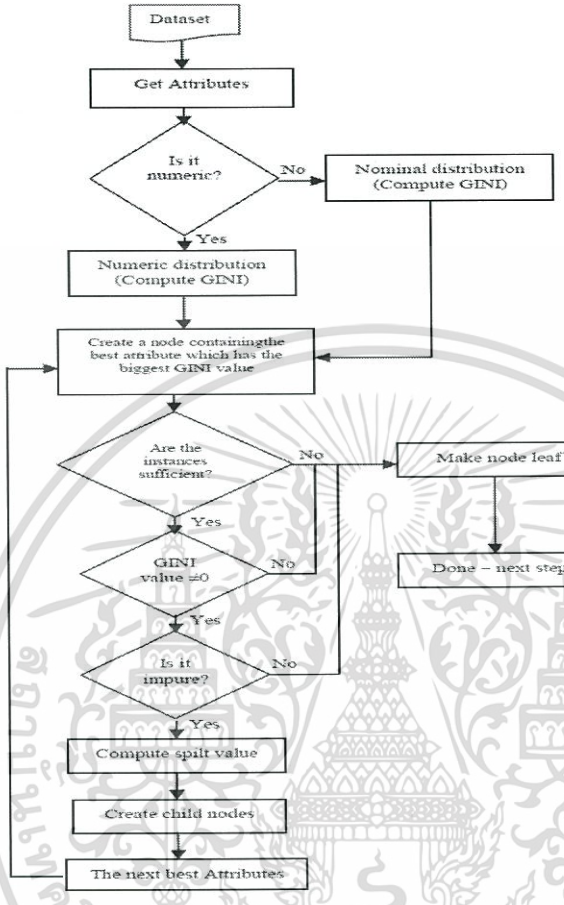


Fig. 6 A basic dataflow diagram of CART [13].

2.5 Neural Network Algorithm

An artificial neural network (ANN), often just called a "Neural network" (NN), is a mathematical model or computational model based on biological Neural networks. [8], [15], [16] The neural network is developed to recognize and associatively retrieve patterns, to solve combinatorial optimization problems, to filter noise from measurement data, and so on. In classification, the Neural network is used to build a predictive model by recognizing patterns that describe the group to which an item belongs by examining existing items or historical items that have been already classified and inferring a set of rules. The multi layer perceptron (MLP) and radial basis function (RBF) networks are the neural networks widely used in classification. The multi-layer perceptron (MLP), also called a feed-forward network, involves estimated weights between the inputs and a hidden layer where the hidden layer has a nonlinear activation function. (Sarle (1994)). In a typical MLP network, all the nodes from a layer are connected with every node from the previous and from the next layer. The Radial Basis Function (RBF) neural network was proposed by Broomhead and Lowe [15]-[17]. This neural network is very

different from neural networks with sigmoidal activation functions in that it utilizes basis functions in the hidden layer that are locally responsive to input stimulus. These hidden nodes are usually implemented using a Gaussian function as the nonlinearity. The key to a successful implementation of these networks is to find suitable centers for the Gaussian functions.

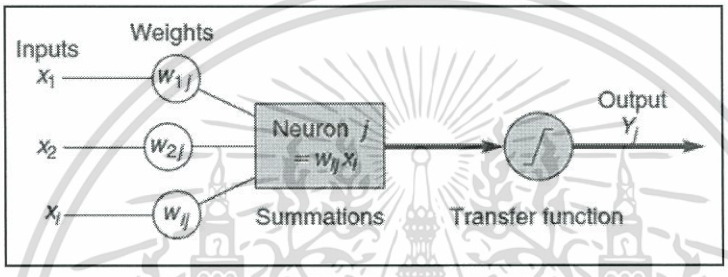


Fig. 7 The processing information of neural networks

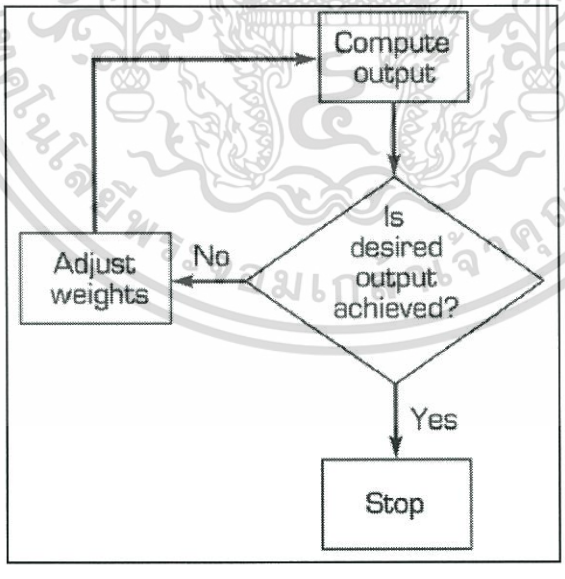


Fig. 8 A basic dataflow diagram of neural networks

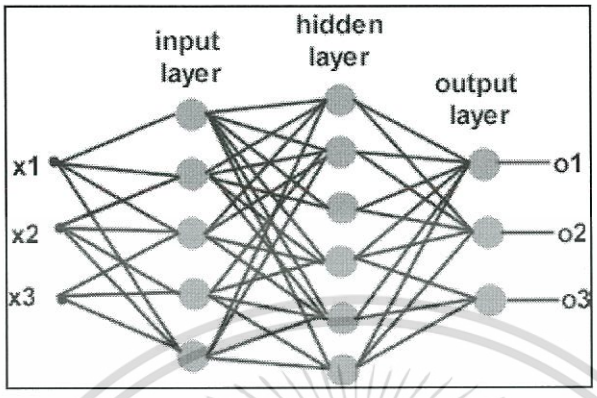


Fig. 9 The multi layer perceptron architecture

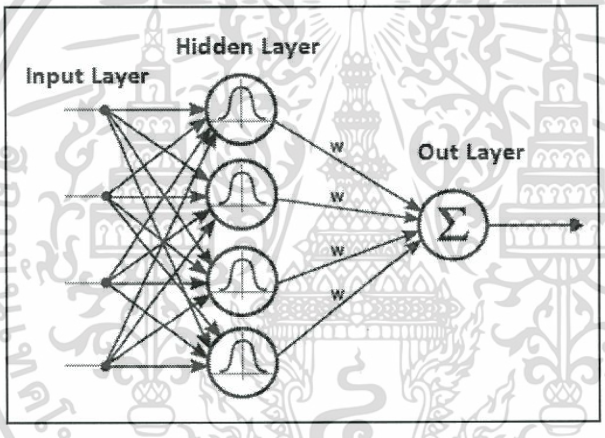


Fig. 10 The radial basis function architecture

2.6 Pearson's Chi-square

Pearson's chi-square [18], [19] is a statistical test commonly used to analyze categorical data between X, the feature under consideration with I categories, and Y target variable with J categories. The Pearson's chi-square test involves the difference between the observed and expected frequencies. Under the null hypothesis of independence, the expected frequencies are estimated by an equation (2).

$$\hat{N} = \frac{N_i \cdot N_j}{N} \tag{2}$$

Under the null hypothesis, Pearson's chi-square converges asymptotically to a chi-squared distribution x_d^2 with degree of freedom, where

$$d = (I-1)(J-1) \tag{3}$$

and the p value is equal with the probability that $x_d^2 > X^2$, where

$$X^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{(N_{ij} - \hat{N}_{ij})^2}{\hat{N}_{ij}} \quad (4)$$

The categorical variables were sorted first by p value in the ascending order, and if ties occurred they were sorted by chi-square in descending order. If ties still occurred, they were sorted by degree of freedom d in ascending order.

The following notation applies:

X is the predictor under consideration with I categories.

Y is target variable with J categories.

N is total number of cases.

N_{ij} is the number of cases with $X = i$ and $Y = j$.

2.7 Cosine Similarity

One very popular measure of similarity between two vectors of n dimensions is the Cosine similarity measure [20]-[23]. The Cosine similarity has its application in text mining and information retrieval. Given two vectors of attributes, $A = \{x_1,x_2,\ldots,x_n\}$ and $B = \{y_1,y_2,\ldots,y_n\}$, the Cosine similarity θ, is the measure of the angle between the two vectors and is defined as [22], [23]:

$$Sim(A,B) = Cos\theta = \frac{\overline{A}\times\overline{B}}{\left|\overline{A}\right|\left|\overline{B}\right|} = \frac{\sum_{i=1}^{n}(x_i \cdot y_i)}{\sqrt{\sum_{i=1}^{n} x_i^{\,2} \cdot \sum_{i=1}^{n} y_i^{\,2}}} \quad (5)$$

3 Proposed Approach

The proposed approach is used to select a robust feature subset to build a learning model for the classification task.

Note that the data which meets the demands of proposed methods must be numerical value. Therefore, the symbolic data should be transformed into numerical and make them under the same standardization.

An algorithm of this proposed approach is provided as follows:

Each attribute of n attributes without class attribute in training data set is represented as $A_1, A_2, A_3, \ldots, A_n$ respectively while m is represented as the number of instances in training data as shown in Fig. 11. The algorithm is as follows:

STEP1: Let L be the number of attributes. Therefore L is equal to n. Then, $C_{i,k}$ is computed by using the equation (6) , where $1 \leq i \leq L$ and $1 \leq k \leq L$. $C_{i,k}$, for any i, is an element in C_k as shown in Fig. 12. $C_{i,k}$ must have a value in rank of 0 and 1. $C_{i,k}$=1 where $i = k$. $C_{i,k}$ should be computed when k is less than i because of $C_{i,k} = C_{k,i}$. Fig. 13 looks like a lower triangular matrix. The number of elements in this matrix is equal to $(L(L\text{-}1))/2$.

STEP2: Then find the maximum value $C_{i,k}$ where $i \neq k$ in each C_k and then put i, the selected attribute index, in the Set R by keeping its frequency also.

STEP3: Next, remove redundant attribute index in Set R.

STEP4: Finally, each attribute index in Set R is promoted as a significant element of the robust feature subset.

$$C_{i,k} = \frac{\sum_{j=1}^{n} (x_{j,i} \cdot x_{j,k})}{\sqrt{\sum_{j=1}^{n} x_{j,i}^{2} \cdot \sum_{j=1}^{n} x_{j,k}^{2}}} \quad (6)$$

Instances	Attributes				
	A_1	A_2	A_3	...	A_n
	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,n}$
	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{2,n}$
	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,n}$
	⋮	⋮	⋮	⋮	⋮
	$X_{m,1}$	$X_{m,2}$	$X_{m,3}$	...	$X_{m,n}$

Fig. 11 Vectors of each attribute

an Ordinal Number of Attributes	an Ordinal Number of Attributes				
	C_1	C_2	C_3	...	C_n
	$c_{1,1}$	$c_{1,2}$	$c_{1,3}$	...	$c_{1,n}$
	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	...	$c_{2,n}$
	$c_{3,1}$	$c_{3,2}$	$c_{3,3}$	...	$c_{3,n}$
	⋮	⋮	⋮	⋮	⋮
	$c_{n,1}$	$c_{n,2}$	$c_{n,3}$	...	$c_{n,n}$

Fig. 12 Structure of any vector C_k

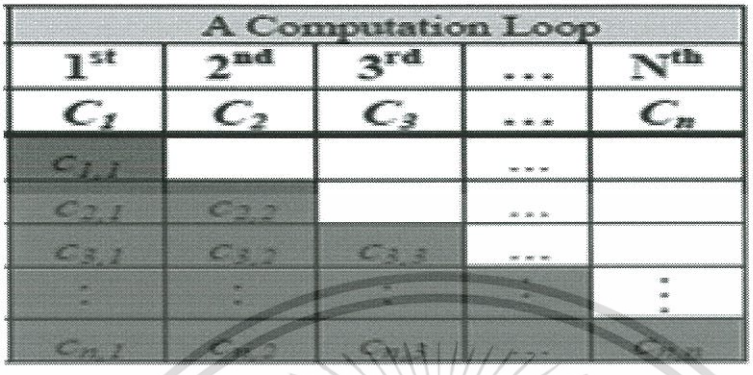


Fig. 13 A computation loop of any vector C_k

For example, below we show how to select feature subset selection of a data set having 42 attributes by using the proposed method as follows:

Note that according to the proposed approach in Section 3, the class attribute was not considered; thus, in this example, there are only 41 attributes computed by this proposed method.

From the proposed approach 1, each attribute of 41 attributes in a training set is represented as $A_1, A_2, A_3, \ldots, A_{41}$ respectively.

STEP1: Next let L be the number of attributes. Therefore L is equal to 41. Then, $C_{i,k}$ is computed by using the equation (6) , where $1 \leq i \leq L$ and $1 \leq k \leq L$. $C_{i,k}$, for any i, is an element in C_k. $C_{i,k}$ must have a value in rank of 0 and 1. $C_{i,k}=1$ where $i = k$. $C_{i,k}$ should be computed when i is less than k because of $C_{i,k} = C_{k,i}$.

STEP2: Then find the maximum value $C_{i,k}$ where $i \neq k$ in each C_k and then put i, the selected attribute index, in the Set R by keeping its frequency also.

STEP3: Next, remove redundant attribute index in Set R.

STEP4: Finally, each attribute index in Set R is promoted as a significant element of the robust feature subset.

A pseudo code of the proposed algorithm

```
BEGIN
/* compute each C_{i,k} using equation (4) */
1.   n = the number of attributes -1;
2.   m= the number of instances;
3.   L=n;
4.   R as array[L];
5.   FOR k=1 TO L
6.       FOR i=1 TO L
7.           FOR j=1 TO m
8.               t1=(x_{j,i}*x_{j,k})+t1;
9.               t2=(x_{j,i})^2+t2;
10.              t3=(x_{j,k})^2+t3;
             END LOOP
11.          C_{i,k}=t1/sqrt(t2*t3);
         END LOOP
     END LOOP
/* to find maximum value in each C_k to get Set R */
12.  FOR k=1 TO L
13.      FOR i=1 TO L
14.          IF i≠k THEN
15.              IF C_{i,k} = MAXIMUM(C_k) THEN
16.                  R_k=i;
                 END IF
             END IF
         END LOOP
     END LOOP
/* to get the robust subset feature from Set R */
17.  Remove the redundant attribute index in R;
18.  The robust subset feature = Set R;
END
```

Fig. 14 A pseudo code of the proposed algorithm

4 Experimental Result

In order to evaluate a final set of robust features gained from our proposed approach in Section 3 with reducing bias of experimental results, the ten data sets from the UCI repository [24] (Breast Cancer, Image Segmentation, MAGIC Gamma Telescope, Musk, Optical Recognition of Handwritten Digits, Page Blocks Classification, Pen-Based Recognition of Handwritten Digits, Red Wine, Statlog and Statlog Land sat Satellite) are used in this paper.

The details of these data sets show in Table 1. Furthermore, C5.0, CART and Neural Network algorithm described in Section 2.3, 2.4 and 2.5 respectively are

used to measure efficiencies of this proposed approach in terms of its accuracy and storage space. For environment of our experimental evaluation, we used the HP ProBook 6450b Laptop with Intel® Core™ i5 CPU 2.40 GHz, and 2 GB of RAM. The operating system was the Windows 7. SPSS Clementine 12 was used as evaluation application. Moreover, C5.0, CART and Neural Network modeling in the Clementine was set up as follows:

Note that SPSS Clementine is the data mining tool that is used to won the British government SMART innovation prize twice. SPSS Clementine not only supports the entire data mining flow composing of getting data, transferring data, modeling, evaluating and deploying but also contribute the accepted data mining standard—CRISP-DM (Cross-Industry Standard Process Data Mining) [25].

The C5.0 was set up following the Fig. 15. (Selecting "Use partitioned data", "Decision tree", "Cross-validate Number of folds: 10", "Simple" and "Accuracy").

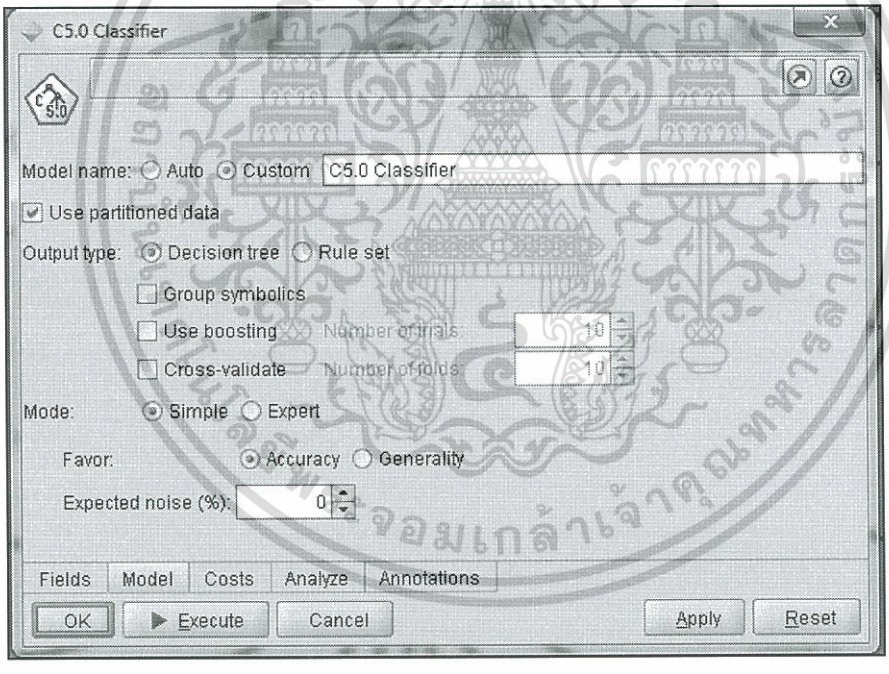


Fig. 15 Setting up C5.0 modelling in Clementine

Meanwhile, the CART was set up following the Fig. 16. (Selecting "Use partitioned data", "Generate Model" and "Maximum tree depth: 5"),

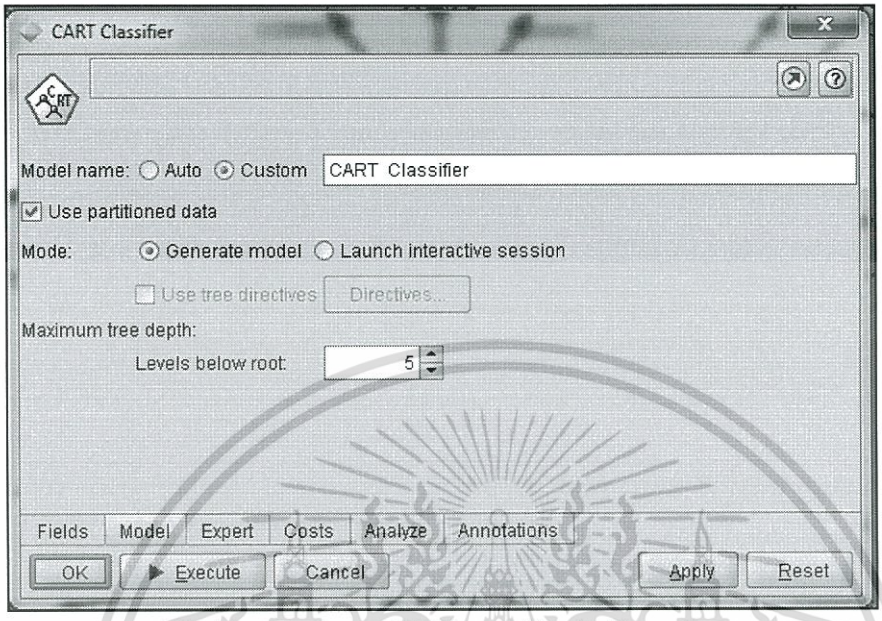


Fig. 16 Setting up CART modelling in Clementine

The Neural Network was set up following the Fig. 17 (Selecting "Use partitioned data", "Method: Quick", "Prevent overtraining Sample %: 50", "Stop on: Default" and "Optimize: Memory").

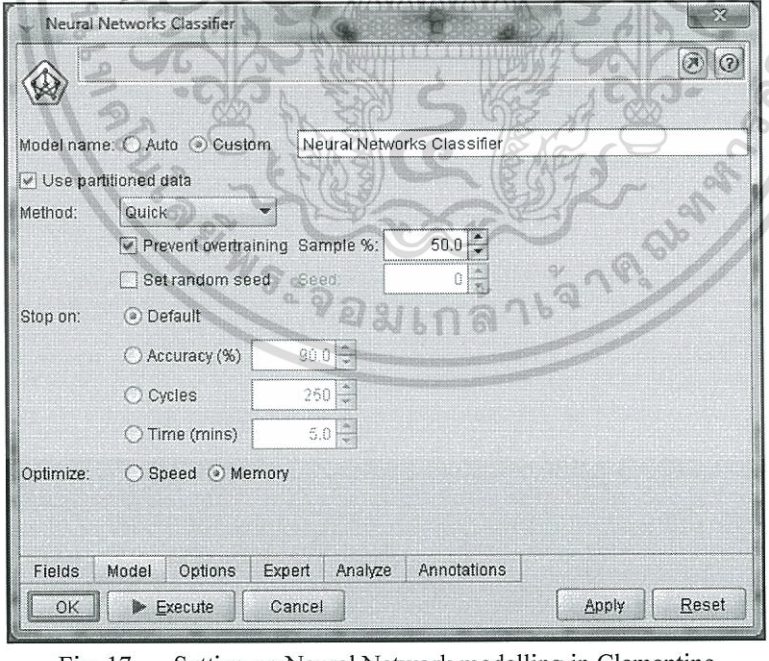


Fig. 17 Setting up Neural Network modelling in Clementine

However, to ensure our proposed approach is effective and practical, it is compared with a well-known algorithm. It is Pearson's chi-square, which was mentioned in Section 2.6.

TABLE 1
THE DETAILS OF DATA SETS FOR EVALUATING THE PROPOSED APPROACH

Data set	NO. Case	NO. Attribute	Detail
Breast Cancer	699	10	To classify breast cancer
Image Segmentation	2310	19	To classify 7 outdoor images
MAGIC Gamma Telescope	19020	11	To classify Gramma signal from images provided by the gamma telescope
Musk	476	168	To predict whether new molecules will be musks or non-musks
Optical Recognition of Handwritten Digits	5620	64	To Classify characters from the optical recognition hand written digits
Page Blocks Classification	5473	10	To Classify all the blocks of the page layout of a document
Pen-Based Recognition of Handwritten Digits	10992	16	To classify characters from the pen-based recognition of hand written digits
Red Wine	178	11	To classify the quality of wine
Statlog	58000	9	To predict the codes of Shuttle
Statlog Landsat Satellite	6435	36	To predict the multi-spectral values

4.1 Preparation Evaluation

In this paper, the performance of the proposed approach is evaluated with classification accuracy using ten-fold cross-validation. Cross-Validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one is used to train a model and the other is employed to validate the model. Moreover, each data set composed of symbolic or string data was transformed into numerical data; furthermore, each attribute of each data set in this paper was made under the same standardization. Afterwards, each data set was randomly divided into 60% for a training set and the rest (40%) for a test set.

4.2 Experimental Results

By using training sets in Section 4.2, the proposed approach and the Pearson's chi-square technique can provide the number of features of each data set as follows: Table 2.

Note that in this paper, CS represents the proposed approach while PS refers to the Pearson's chi-square method.

The Table 2 shows the proposed approach can reduce the number of features from the number of entire features in each data set impressively. However, when it is compared with the number of features given by Pearson's chi-square method, the number of features produced by both techniques is not different obviously; this proposed approach provides the number of features smaller than Pearson's chi-square method in every data sets. Therefore, in the real-world applications, the proposed approach is better than the Pearson's chi-square method because the smaller features are always advantageous in terms of computational cost of processing data.

TABLE 2
THE NUMBER OF SELECTED FEATURES OF EACH METHOD

Data set	NO. Entire Features	NO. Features provided by CS	NO. Features provided by PS
Breast Cancer	10	4	9
Image Segmentation	19	11	14
MAGIC Gamma Telescope	11	8	8
Musk	168	112	118
Optical Recognition of Handwritten Digits	64	42	48
Page Blocks Classification	10	8	10
Pen-Based Recognition of Handwritten Digits	16	8	16
Pima Indians Diabetes	8	5	7
Red Wine	11	7	9
Statlog	9	7	8
Statlog Landsat Satellite	36	23	36

Note that the number of features in Table 2 is without the class label or class attribute.

Note that the measurement in this paper of the experimental results is based on the detection rate or accuracy rate refers to the ratio between the number of correct detection, and the total number of cases or instances and the error detection rate or wrong rate means the ratio between the number of incorrect detection, and the total number of cases or instances.

1) Breast Cancer Data Set

For the results based on the Breast Cancer data set in Table 3, the C.50, CART and Neural Network algorithms based on a feature subset provided by the CS—the proposed approach, are quite better than the PS—the Pearson's chi-square method. However, for this data set, the C.50 gives more accurate than any others. Thus, C.50 algorithm using a feature subset based on the proposed approach is recommended for this data set.

TABLE 3
THE ACCURACY RATE AND ERROR RATE OF THE BREAST CANCER DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	96.63%	95.19%	94.23%	92.31%	92.31%	91.35%
Wrong	3.37%	4.81%	5.77%	7.69%	7.69%	8.65%

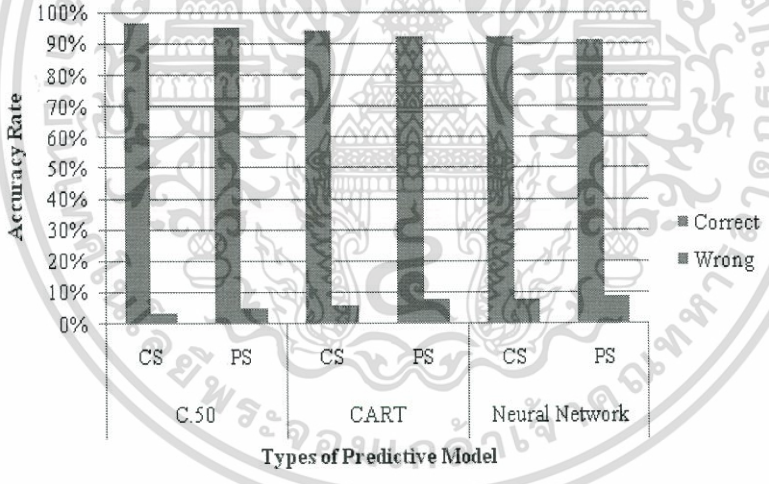


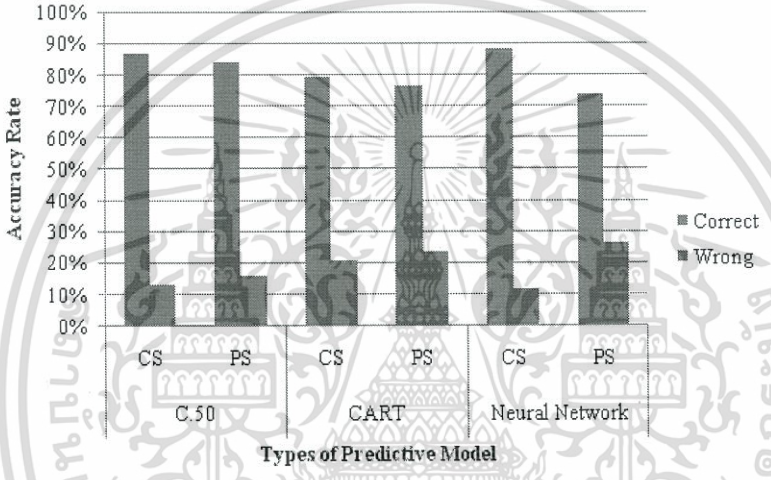
Fig. 18 The comparison of experimental results of Breast Cancer data set using the different

2) Image Segmentation Data Set

For the results based on Image Segmentation data set in Table 4, the Neural Network algorithm using a feature subset based on the proposed approach is better than the others because it provides the best accurate result.

TABLE 4
THE ACCURACY RATE AND ERROR RATE OF THE IMAGE SEGMENTATION DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	86.76%	83.82%	79.41%	76.47%	88.24%	73.53%
Wrong	13.24%	16.18%	20.59%	23.53%	11.76%	26.47%



3) MAGIC Gamma Telescope Data Set

In Table 5, the C.50, CART and Neural Network algorithms based on a feature subset provided by the proposed approach, are better than a feature subset given by the Pearson's chi-square method.

TABLE 5
THE ACCURACY RATE AND ERROR RATE OF THE MAGIC GAMMA TELESCOPE DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	86.71%	85.21%	82.57%	76.62%	85.48%	77.80%
Wrong	13.29%	14.79%	17.43%	23.38%	14.52%	22.20%

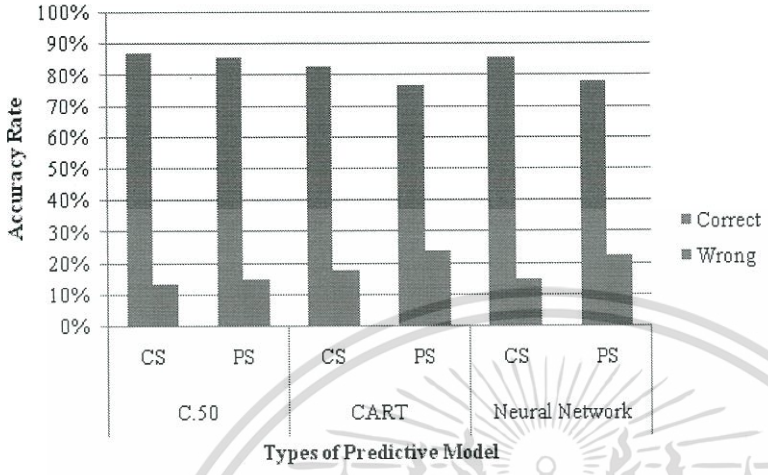


Fig. 20 The comparison of experimental results of MAGIC Gamma Telescope data set using the different feature subset of two models based on three classifiers.

4) Musk Data Set

In this data set, the results in Table 6 show the C.50 algorithm using a feature subset based on the CS is the same correct value as C.50 algorithm using a feature subset based on the PS while the other algorithms based on a feature subset provided by the proposed approach are better than a feature subset given by PS method.

TABLE 6
THE ACCURACY RATE AND ERROR RATE OF THE MUSK DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	97.97%	97.97%	79.73%	78.38%	81.76%	77.03%
Wrong	2.03%	2.03%	20.27%	21.62%	18.24%	22.97%

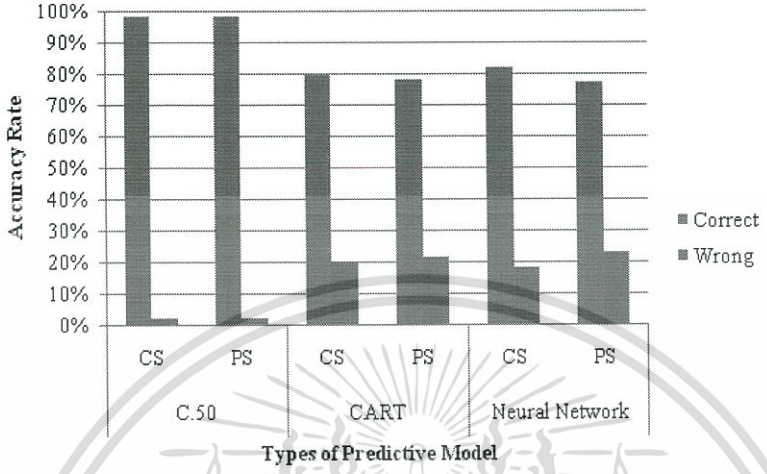


Fig. 21 The comparison of experimental results of Musk data set using the different feature subset of two models based on three classifiers.

5) Optical Recognition of Hand written Digits Data Set

For the results based on the Optical Recognition of Hand written Digits data set in Table 7, the C.50, CART and Neural Network algorithms based on a feature subset provided by the proposed approach, are better than the Pearson's chi-square method. Furthermore, the C.50 algorithm using a feature subset based on the proposed approach is recommended for this data set because it can give the best accurate value.

TABLE 7
THE ACCURACY RATE AND ERROR RATE OF THE OPTICAL RECOGNITION OF HAND WRITTEN DIGITS DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	94.83%	92.74%	68.64%	58.22%	79.32%	57.13%
Wrong	5.17%	7.26%	31.36%	41.78%	20.68%	42.87%

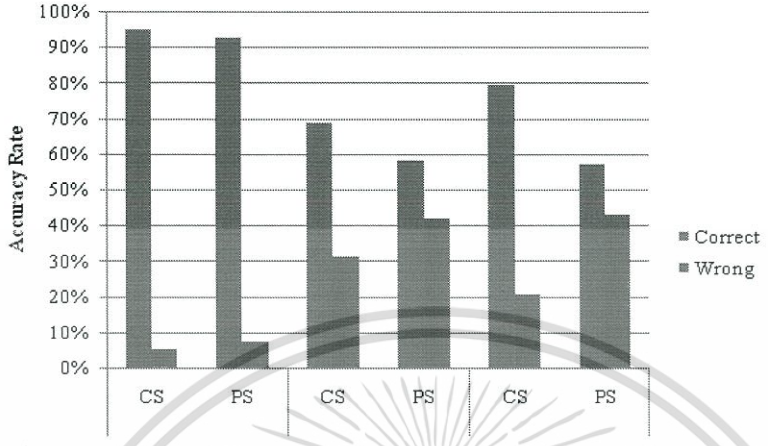


Fig. 22 The comparison of experimental results of Optical Recognition of Hand written Digits data set using the different feature subset of two models based on three classifiers.

Page Blocks Classification Data Set

For the results based on the Page Blocks Classification data set in Table 8, the C.50, CART and Neural Network algorithms based on a feature subset provided by the proposed approach, are still better than the Pearson's chi-square method. Besides, the C.50 algorithm also gives the best accurate value.

TABLE 8
THE ACCURACY RATE AND ERROR RATE OF THE PAGE BLOCKS CLASSIFICATION DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	98.06%	96.61%	95.09%	94.97%	95.58%	95.46%
Wrong	1.94%	3.39%	4.91%	5.03%	4.42%	4.54%

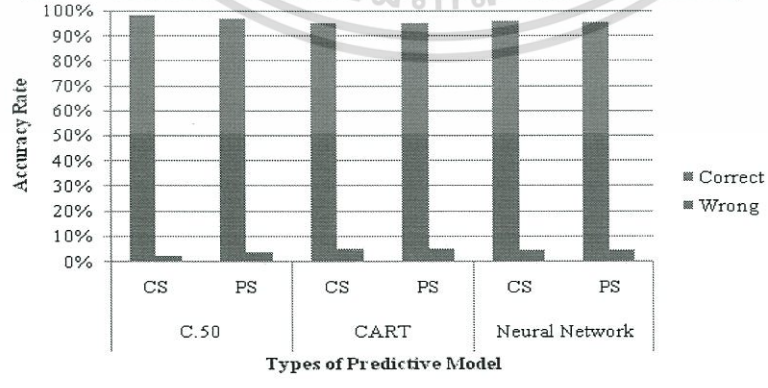


Fig. 23 The comparison of experimental results of Page Blocks Classification data set using the different feature subset of two models based on three classifiers.

6) Pen-Based Recognition of Hand Written Digits Data Set

For the results based on the Pen-Based Recognition of Hand Written Digits data set in Table 9, a feature subset provided by the proposed approach can improve the accuracy rate of the C.50 and CART algorithms but also the Neural Network algorithm.

TABLE 9
THE ACCURACY RATE AND ERROR RATE OF THE PEN-BASED RECOGNITION OF HAND WRITTEN DIGITS DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	96.60%	95.46%	79.18%	75.83%	82.75%	77.99%
Wrong	3.40%	4.54%	20.82%	24.17%	17.25%	22.01%

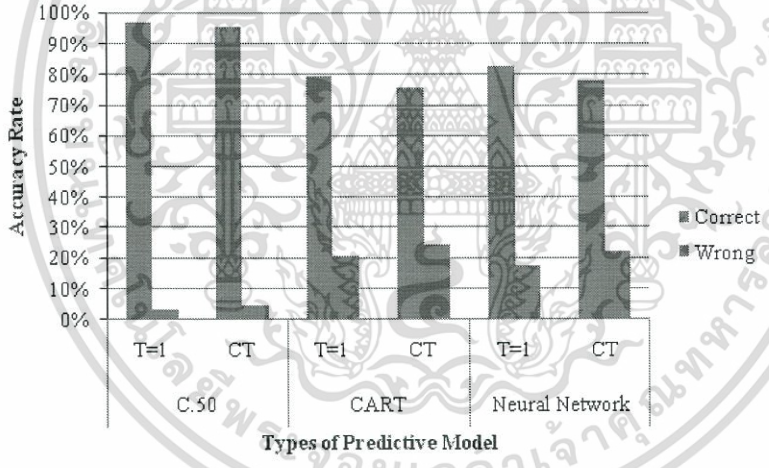


Fig. 24 The comparison of experimental results of Pen-Based Recognition of Hand Written Digits data set using the different feature subset of two models based on three classifiers.

7) Red Wine Data Set

For the results based on Red wine data set in Table 10, the C.50 algorithm using a feature subset based on the proposed approach is better than the others because it provides the best accurate rate. However, the CART and Neural Network algorithms based on the feature subset provided by the proposed approach is better than a feature subset given by the Pearson's chi-square method also.

TABLE 10
THE ACCURACY RATE AND ERROR RATE OF THE RED WINE DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	81.82%	81.62%	67.47%	59.39%	58.18%	55.35%
Wrong	18.18%	18.38%	32.53%	40.61%	41.82%	44.65%

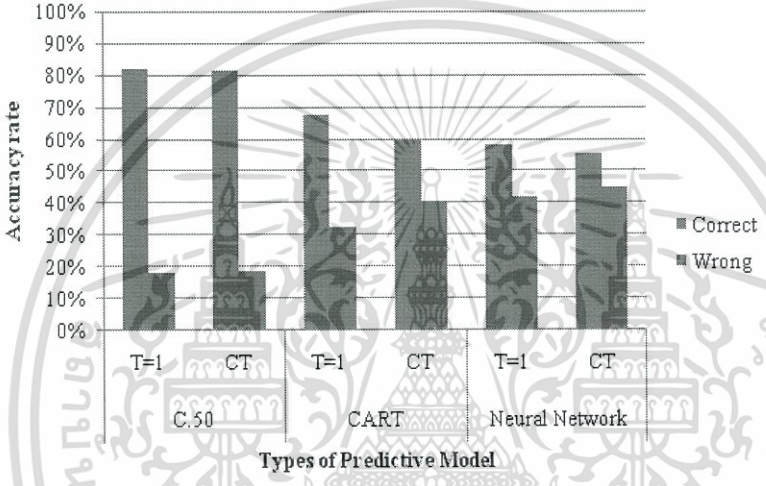


Fig. 25 The comparison of experimental results of Red Wine data set using the different feature subset of two models based on three classifiers.

8) Statlog Data Set

For the results based on the Statlog data set in Table 11, a feature subset provided by the proposed approach and a feature subset provided by the Pearson's chi-square method can improve the accuracy rate of the C.50 and CART and Neural Network algorithms alike.

TABLE 11
THE ACCURACY RATE AND ERROR RATE OF THE STATLOG DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	99.86%	99.82%	99.49%	99.49%	99.60%	99.60%
Wrong	0.14%	0.18%	0.51%	0.51%	0.40%	0.40%

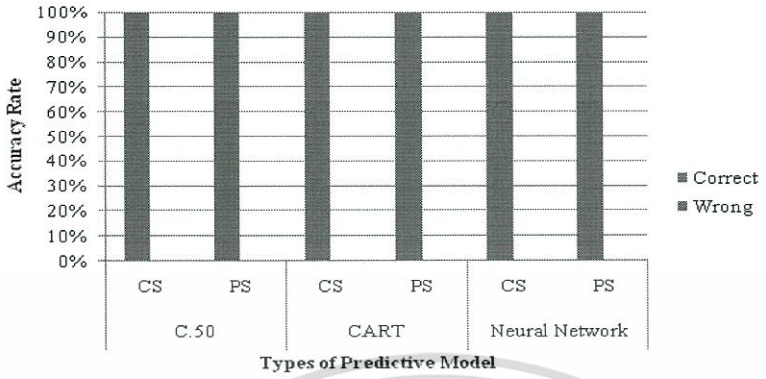


Fig. 26 The comparison of experimental results of Statlog data set using the different feature subset of two models based on three classifiers.

9) Stat log Land sat Satellite Data Set

For the results based on Stat log Land sat Satellite data set in Table 12, the C.50 algorithm using a feature subset based on the proposed approach is the best accurate rate. Nevertheless, the CART and Neural Network algorithms based on the feature subset provided by the proposed approach is better than a feature subset given by the Pearson's chi-square method.

TABLE 12
THE ACCURACY RATE AND ERROR RATE OF THE STAT LOG LAND SAT SATELLITE DATA SET

	C.50		CART		Neural Network	
	CS	PS	CS	PS	CS	PS
Correct	95.07%	94.77%	81.96%	81.66%	86.08%	84.46%
Wrong	4.93%	5.23%	18.04%	18.34%	13.92%	15.54%

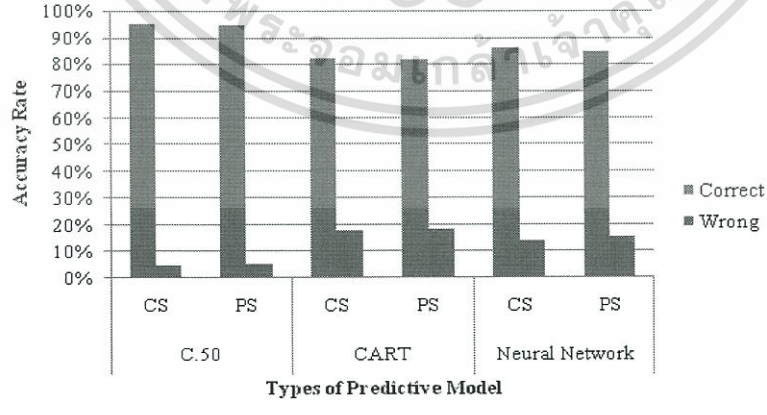


Fig. 27 The comparison of experimental results of Stat log Land sat Satellite data set using the different feature subset of two models based on three classifiers three classifiers.

5 Conclusions

To improve the performance of accuracy rate, the approach is proposed by applied the Cosine similarity method for selecting a robust feature subset. The proposed approach based on the Cosine similarity was mentioned thoroughly in Section 3. From the experimental results in Section 4.2, the proposed approach yielded a higher performance of the accuracy rate in every the benchmark data sets compared with Pearson's chi-square method widely used in Commercial applications such as SPSS Clementine, SAS, Minitab and so on. Moreover, Table 3 to Table 12 and Fig. 18 to Fig. 27 in Section 4.2 show the efficiency of the feature subset based on the proposed approach can improve the accuracy rate of widespread classification algorithms, for instance, C.50, CART and Neural Networks used in this paper. Furthermore, in Table 1, it shows the proposed approach is able to select a robust feature subset with the number of features that is smaller than the Pearson's chi-square method. The proposed approach is with the number of the smaller features which led to using smaller storage space and reducing computation time. Besides, with this proposed approach, the relevant features are produced automatically without manually setting up a threshold parameter. In addition, the proposed approach is not complicated techniques and is easier to understand because the proposed approach has few procedures with only common standard equation of the Cosine similarity and all procedures are simple and understandable (see in Fig. 14). Additionally, time processing is quite important as in real-world applications, any techniques extracting feature sets rapidly are always beneficial in terms of computational cost of processing data.

From the experimental results, it can be concluded the proposed approach based on the Cosine similarity method is a simple feature selection algorithm using smaller storage space, reducing computation time, gaining higher predictive performance and being compatible with the well-known classification algorithms. Moreover, there is no need to set up the threshold parameter.

Acknowledgment

The authors would like to thank the Office of the Higher Education Commission of Thailand for financial support throughout this research. The authors are also grateful to Mathematics and Computer lecturers for their valuable comments and suggestions to improve the quality of the paper. Moreover, the authors are thankful to King Mongkut's Institute of Technology Ladkrabang and Ubon Ratchathani University for partial supporting.

References

[1] A. Jain and D. Zongker, Feature selection: evaluation, application, and small sample performance, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19 (1997), 153-158.

[2] H. Liu, et al., Evolving feature selection, Intelligent Systems, IEEE, 20 (2005), 64-76.

[3] Y. Caballero, et al., Feature Selection Algorithms Using Rough Set Theory, in Intelligent Systems Design and Applications, ISDA 2007. Seventh International Conference on, 2007, 407-411.

[4] T. May, et al., Guiding feature subset selection with an interactive visualization, in Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on, 2011, 111-120.

[5] W. Duch, et al., Feature Ranking, Selection and Discretization, Int. Conf. on Artificial Neural Networks (ICANN) and Int. Conf. on Neural Information Processing (ICONIP), 2003, 251-254.

[6] K. Ron, Feature subset selection using the wrapper method: overfitting and dynamic search space topology, International Proceedings of the AAAI Fall symposium on relevance, 1994, 109-113.

[7] A. G. Karegowda, et al., Article:Feature Subset Selection using Cascaded GA & CFS: A Filter Approach in Supervised Learning, International Journal of Computer Applications, 23 (2011), 1-10.

[8] L. H. Witten and E. Frank, Eds., Data Mining: Practical Machine Learning Tools and Techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[9] T. Bujlow, et al., A method for classification of network traffic based on C5.0 Machine Learning Algorithm, in Computing, Networking and Communications (ICNC), 2012 International Conference on, 2012, 237-241.

[10] N. Zhixian, et al., Auto-recognizing DBMS Workload Based on C5.0 Algorithm, in Knowledge Discovery and Data Mining, WKDD 2009. Second International Workshop on, 2009, 777-780.

[11] Y. Gu and W. Guo, Decision Tree Method in Financial Analysis of Listed Logistics Companies, in Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on, 2010, 1101-1106.

[12] J. R. Quinlan, C4.5: programs for machine learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[13] P. R. J. Campbell, et al., Fuzzy CART: A novel Fuzzy Logic based Classification & Regression Trees Algorithm, in Innovations in Information Technology, IIT '09. International Conference on, 2009, 175-179.

[14] S. Gey and E. Nedelec, Model selection for CART regression trees, Information Theory, IEEE Transactions on, 51 (2005), 658-670.

[15] W. S. Sarle, Neural Networks and Statistical Models, Proceedings of the Nineteenth Annual SAS Users Group International Conference, 1994.

[16] Y. Singh and A. Chauhan, Neural networks in data mining, Journal of Theoretical and Applied information Technology, 2005, 37- 42.
[17] G. P. Zhang, Neural networks for classification: a survey, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 30 (2000), 451-462.
[18] A. Zanobini, The use of student, Chi-square and F distributions to quantify the uncertainty coverage interval in the case of different observed values, in Advanced Methods for Uncertainty Estimation in Measurement, AMUEM 2009. IEEE International Workshop on, 2009, 58-62.
[19] D. Liuling, et al., Using Modified CHI Square and Rough Set for Text Categorization with Many Redundant Features, in Computational Intelligence and Design, ISCID '08. International Symposium on, 2008, 182-185.
[20] A. Karnik, et al., Detecting Obfuscated Viruses Using Cosine Similarity Analysis, in Modelling & Simulation, AMS '07. First Asia International Conference on, 2007, 165-170.
[21] L. Muflikhah and B. Baharudin, Document Clustering Using Concept Space and Cosine Similarity Measurement, in Computer Technology and Development, ICCTD '09. International Conference on, 2009, 58-62.
[22] Y. Soe-Tsyr and S. Jerry, Ontology-based structured cosine similarity in document summarization: with applications to mobile audio-based knowledge management, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 35 (2005), 1028-1040.
[23] N. Swe Swe, Mining contents in Web page using cosine similarity, in Computer Research and Development (ICCRD), 2011 3rd International Conference on, 2011, 472-475.
[24] A. Asuncion and D. Newman. (8 March). UCI Machine Learning Repository. Available: http://www.ics.uci.edu/mlearn/MLRepository.html
[25] Z. Lin, et al., Application of Data Mining Classification Algorithms in Customer Membership Card Classification Model, in Information Management, Innovation Management and Industrial Engineering, ICIII '08. International Conference on, 2008, 211-215.

Received: September, 2012

BIOGRAPHY

PERSONNEL INFORMATION

Thai Name: นายอนิรุธ สืบสิงห์

English Name: Mr. Anirut Suebsing

Date of Birth: 12 September 1978

Permanent Address: 11/1 Burphanok Rd. Tombon Nai Maung, Aumper Maung, Ubon Ratchathani, 34000, Thailand.

Telephone: (+66) 089-6777-7007

E-mail: s9062952@kmitl.ac.th, anirut_s@msn.com

EDUCATION

2005 – at the present **Doctor of Philosophy in Computer Science.**
Department of Computer Science, Faculty of Science,
King Mongkut's Institute of Technology Ladkrabang,
Bangkok, 10520, Thailand.

2002 – 2005 **Master Degree in Information Technology.**
Department of Computer Science, Faculty of Science,
Ubon Ratchathani University,
Ubon Ratchathani,34120, Thailand.

1997 – 2001 **Bachelor Degree in Computer Science.**
Department of Mathematics and Computer Science,
Faculty of Science,
Rajabhat Ubonratchathani Institue,
Ubon Ratchathani, 3400, Thailand.

1993 – 1996 **Hischool in Science-Math Programme.**
Sripatum Hischool,
Ubon Ratchathani, 34000, Thailand.