NETWORK PROBLEMS DETECTION AND CLASSIFICATION BY ANALYZING SYSLOG DATA

FIDAA A. M. JARGHON

Supervisors:
Dr. Mohammad Shamrie Sainin
Assoc. Prof. Hatim Tahir

2016

Network Problems Detection and Classification by Analyzing Syslog data

By

Fidaa A. M. Jarghon

Supervisors:
Dr. Mohammad Shamrie Sainin
Assoc. Prof. Hatim Tahir

ABSTRAK

Rangkaian penyelesaian masalah adalah satu proses penting yang mempunyai bidang penyelidikan yang luas. Langkah pertama dalam prosedur penyelesaian masalah adalah mengumpul maklumat untuk mengenal pasti permasalahan. Mesej syslog yang dihantar oleh hampir semua peranti rangkaian mengandungi sejumlah besar data yang berkaitan dengan masalah rangkaian. Banyak kajian yang dijalankan sebelum ini didapati telah menggunakan menganalisis data syslog yang boleh membimbing untuk masalah rangkaian dan sebab-sebabnya. Mengesan masalah rangkaian akan menjadi lebih efektif jika masalah yang hendak dikesan telah dikelaskan dari segi lapisan rangkaian. Pengelasan data syslog perlu mengenal pasti mesej syslog yang menghuraikan masalah rangkaian untuk setiap lapisan, dan mengambil kira format yang berbeza dari pelbagai syslog untuk peranti vendor. Kajian ini menyediakan kaedah untuk mengelaskan mesej syslog yang menunjukkan masalah rangkaian dari segi lapisan rangkaian. Alat pengenalpastian data kaedah digunakan untuk pengelasan mesej syslog manakala penerangan bahagian atas mesej syslog telah digunakan untuk proses pengelasan. Apabila mesej syslog berkaitan telah dikenal pasti; ciri kemudiannya dipilih untuk melatih penjodoh bilangan. Enam algoritma pengelasan telah dipelajari iaitu LibSVM, SMO, KNN, Naive Bayes, J48, dan Random Forest. Satu set data sebenar yang diperoleh daripada peranti rangkaian Universiti Utara Malaysia (UUM) digunakan untuk peringkat ramalan. Keputusan merumuskan bahawa SVM menunjukkan prestasi terbaik semasa peringkat latihan dan ramalan. Kajian ini menyumbang pada bidang penyelesaian masalah rangkaian, dan pengelasan.

Keywords data teks: Pengelasan, SVM, Pengesanan Kerosakan..

ABSTRACT

Network troubleshooting is an important process which has a wide research field. The first step in troubleshooting procedures is to collect information in order to diagnose the problems. Syslog messages which are sent by almost all network devices contain a massive amount of data related to the network problems. It is found that in many studies conducted previously, analyzing syslog data which can be a guideline for network problems and their causes was used. Detecting network problems could be more efficient if the detected problems have been classified in terms of network layers. Classifying syslog data needs to identify the syslog messages that describe the network problems for each layer, taking into account the different formats of various syslog for vendors' devices. This study provides a method to classify syslog messages that indicates the network problem in terms of network layers. The method used data mining tool to classify the syslog messages while the description part of the syslog message was used for classification process. Related syslog messages were identified; features were then selected to train the classifiers. Six classification algorithms were learned; LibSVM, SMO, KNN, Naïve Bayes, J48, and Random Forest. A real data set which was obtained from the Universiti Utara Malaysia's (UUM) network devices is used for the prediction stage. Results indicate that SVM shows the best performance during the training and prediction stages. This study contributes to the field of network troubleshooting, and the field of text data classification.

Keywords: Classification, SVM, Fault Detection

TABLE OF CONTENTS

Title **Page**

LIST OF TABLES

Table Page

No table of figures entries found.

LIST OF FIGURES

CHAPTER One

INTRODUCTION

Most institutions and organizations, regardless of the business types, rely on networks to manage their business. Any failure or error which occurs in the network will negatively affect their achievements, productivity and services. Therefore, it is necessary to diagnose and detect the reasons behind network failures and problems in order to fix them and reduce similar occurrences in the future. Network troubleshooting , which begins by diagnosing the problems, is a complex process. The first step is to collect information [1].

Collecting information includes answering this question, "what are the potential errors that can lead to network problems and failures?" Kyas [2], has identified five categories of errors: operator error, mass storage problems, computer hardware problems, software problems, and network problems. Hudyma & Fels [3], added two new categories: failure due to denial of service attacks (Worms, Viruses, Trojan Horses and Malicious software), and failure due to disasters such as fire, flood, earthquakes, outages and the like. Network problems, which include hardware and software problems that are directly related to the network architecture [2],account for more than one-third of information technology (IT) failures.

Network architecture is organized as a series of layers or levels [4], Open Systems Interconnection (OSI) model, and Transmission Control Protocol/Internet Protocol (TCP/IP) model, which separates network functionality into modular layers that provide "a common language for network engineers and is usually used in troubleshooting networks."

Troubleshooting could be an efficient process if it relies on a systematic approach which minimizes confusion and shortened troubleshooting time. It is carried out using the Layered Model [5] as problems are normally described in terms of a specific model layer [1]. Network errors could be distributed into the network layers depending on OSI model or TCP\IP model (physical layer, data-link layer, network layer, transport layer, application layer). And knowing the layer that has problems in

its components can lead to efficient maintenance decisions. This can be done by analyzing network problems that happened during a period of time [2].

Syslog messages contain a massive amount of data related to network problems and this data is sent by almost all network equipment such as routers, switches, firewalls [6]. Analyzing the syslog data will help to identify network problems and their causes. Network problems can be classified according to their causing layers, depending on the layer elements problems. Through classification, the intended layer that causes network problems or failures, can be detected and therefore, treated immediately.

Detecting and classifying network problems through syslog are done by analyzing syslog messages in order to extract the data related to network problems; the next step is classification of the extracted data. The main issue here is that syslog data is not a normal data, in terms of the type and the volume, but an unstructured textual data, with a variety of formats and a big volume. These characteristics of syslog data make it difficult to analyse using traditional computing techniques and conventional database systems [7]. Analysis of Syslog data requires a special tool which involves data mining algorithms [8].

1.1 Motivations

The increase in telecommunication networks complexity means that managing networks have become more difficult, especially in detecting and classifying network problems which is crucial in making maintenance decision [1]. Syslog data, which are messages sent by network devices, contain massive volume of data related to network problems [9]. The progress in the field of text data mining algorithms and its tool has encouraged its use in the field of network management [10]. Today, the application of data mining algorithms to analyze and classify syslog data for network operation purposes such as detection and classification of network problems is widely used. The use of such techniques makes it easier to make an efficient maintenance decision for network management.

1.2 Problem Statement

Networks consist of many kinds of equipment which are not only different in types but also vendors and as a result, detecting network problems and diagnosing their causes for maintenance purposes are difficult [11]. Analyzing data using syslog network elements thus provides an efficient method of network problems detection and diagnosis. Currently, there are a number of network management devices, monitoring technology, and studies which are devoted to detect network problems from log messages [12]. Detecting the problems alone is insufficient to identify the source of the problem; it has to include the task of reading the detected syslog messages one by one for further understanding and diagnosing the responsible layer that causes network problems. The whole process involves cost, time and effort [11].

Classifying network problems in terms of network layers are done using the TCP/IP model. Having the knowledge of each layer of components and their potential problems are necessary in order to identify the responsible layer or device. In fact, it is very difficult to classify syslog messages due to the following reasons: the various types of logs which list messages with low or high severity [11]; the increasing number of network elements means that there is a massive volume of complex log data, and it is therefore, necessary to extract information accurately and efficiently in order to make correct maintenance decisions; and finally the log format, which depends on each vendor or service [13]. Thus, understanding the meaning of each log message requires deep domain knowledge of each format.

Classifying syslog data is done by analyzing syslog messages in order to identify related messages that describe the network problems of each layer. Data mining techniques provide many methods and algorithms for data analysis and classification, but the difficulty lies in how to determine the best classifier model. Knowledge of the classifier is, therefore, necessary in order to be able to make prediction and identify the features that represent each problem [14].

1.3 Research Questions

The following are the research questions for this study:

i. How to identify the syslog messages related to network problems for each network layer?
ii. How to classify syslog messages related to the network problems in terms of network layers?

1.4 Research Objectives

The main objective of the proposal is to investigate a method to classify network problems in terms of network layers using syslog data. In order to achieve this objective, the following research objectives have been formulated, which are:

i. To identify the syslog messages related to network problems for each network layer.
ii. To classify syslog messages related to the network problems in terms of network layers.

1.5 Significance of the Research

Syslog data provides important information about network problems and this is done by taking data which can give specific information from almost all network devices and elements. Syslog data is an important source due to its massive volume of information related to network [12]. Therefore, analyzing syslog data to detect network problems assists in network troubleshooting but detecting the problems alone is not sufficient to identify the source of the problem. It requires reading the detected syslog messages one by one for further understanding and diagnosing the responsible layer that causes the network problems. This process involves cost, time and effort. Classifying network problems in terms of network layers could be more efficient for maintenance decision, and network troubleshooting [13]. Applying data

mining techniques on syslog data could help to detect syslog messages that describe network problems, and classify them based on the related network layer.

1.6 Scope of the Research

This research focuses on how to detect and classify network problems from syslog data using data mining techniques. The following are the scope of this work:

i. This research focuses on developing a method to detect and classify network problems related to TCP/IP network layers (network access layer, Internet layer, transport layer, application layer).
ii. The intended messages to be extracted for analysis are the messages of network problems.
iii. Extracted syslog messages related to specified network problems.
iv. This study relied on Cisco syslog manual to identify syslog error messages.
v. Six Classification algorithms (Support Vector Machines (LibSVM , SMO), Naive Bayes (NB), K-Nearest Neighbour (k-NN), J48 Decision Tree and Random Forest) are applied on training dataset.
vi. Each message has been classified to one class (one layer) only.

1.7 Organization of the Study

Chapter one introduces network problems and troubleshooting, network layers modules, syslog data. It also includes problem statement, objectives, significance, and scope.

Chapter two explains the key terms of the study and reviews the literature of the study and the related works.

Chapter three explains the methodology phases that have been followed to achieve the objectives and the tools used .

Chapter four represents the results of experiments and validation.

Chapter five includes study summary, contribution, limitations, and future works.

CHAPTER Two

LITERATURE REVIEW

This chapter reviews the literature of the study, explains the key terms of the study, and presents the previous works related to the study. This chapter has 13 sections, summarized in the following paragraphs. Section 2.1 talks about the networks components, types, and architecture models, Open Systems Interconnection (OSI) model, and Internet TCP/IP model and their components. Section 2.2 explains the network layers, the components and functions of each layer (network access layer, Internet layer, transport layer, application layer). Section 2.3 discusses network problems and the categories of problems caused. Section 2.4 explains network troubleshooting approaches, in particular the layered model and general troubleshooting procedures. Section 2.5 describes symptoms and causes of network problems related to each layer. Section 2.6 discusses syslog data and the components of syslog messages. Section 2.7 explains unstructured data, its definition, characteristics, and sources; it also explains unstructured data mining techniques, and processing. Section 2.8 describes text data, its characteristics, classification methods and procedures. Section 2.9 describes machine learning techniques, illustrates number of text classification algorithms; support vector machine (SVM), K-nearest neighbour (KNN), decision tree, and Naïve Bayes Algorithm (NB). Section 2.10 discusses approaches to create feature vector for text classification, indicates the methods of vector term identification and the methods of feature weights representation, and Boolean weighting method for the traditional document representation. Section 2.11 discusses feature selection method and procedures for text data features selection, and illustrate document frequency thresholding (DF) as it is the used features selection method in this study. Section 2.12 revises the previous works that use DF method for reducing dimension. The last section, 2.13, revises the previous works of syslog data analysis.

2.1 Computer Networks

A computer network is the infrastructure that allows two or more computers (called nodes) to communicate with each other, and this is achieved by using the communication protocol [6]. In general, networks consist of two types of components: nodes and communication lines. A node is usually a computer with specific network software, and the communication lines are : copper wire cables, optical fibre, radio channels, and telephone lines [4].

Networks can be classified into different types or categories depending on the specified criteria, for example: geographical area covered, access restrictions and the communication model applied. Regardless of the network type, it should have an architecture [15].

Network architectures identify the concepts and techniques for designing and building systems of computers communication. There are different types of network architecture and most of them are organized as a series of layers or levels. The basic idea of a layered architecture is to divide the design into small parts, one of which is the Open Systems Interconnection (OSI) model [4].

2.1.1 Open Systems Interconnection (OSI)

The International Organization for Standardization (ISO) has developed OSI as a conceptual model for network architecture. The model consists of seven layers and each layer has a specific network functions and can be implemented independently. The layers 5, 6, and 7 deal with application issues and are implemented only in software; the rest of the layers handle data transport issues. Layer 3 and 4 are implemented only in software; the physical and data link layers are implemented in software and hardware. Seven layers of OSI include [4]: Physical layer, Data link layer, Network layer, Transport layer, Session layer, Presentation layer, Application layer.

2.1.2 Internet TCP/IP Model

The Internet model with Transmission Control Protocol/Internet Protocol (TCP/IP) consists of four levels of protocols that are related to the seven layers of the OSI architecture. The first layer provides functions that correspond to the first two layers of OSI model. The Internet layer is equivalent to the network layer. The three high layers in the OSI model are represented in TCP/IP by a single layer called the application layer (Figure 2.1). Four levels of TCP/IP include [16]: Network access layer, Internet layer, Transport layer, Application layer. This study uses TCP/IP for classifying network problems to its layers because it is used by Internet applications like email, World Wide Web, FTP, etc. It has only of four layers (network access layer, Internet layer, transport layer, application layer) [17], and this is more general and efficient than the OSI model, when classifying network problems to its layers. Since TCP/IP model consists of only four layers, compared to the OSI model which consists of seven layers, the TCP/IP model is found to be more efficient and easier to classify network problems to its layers.

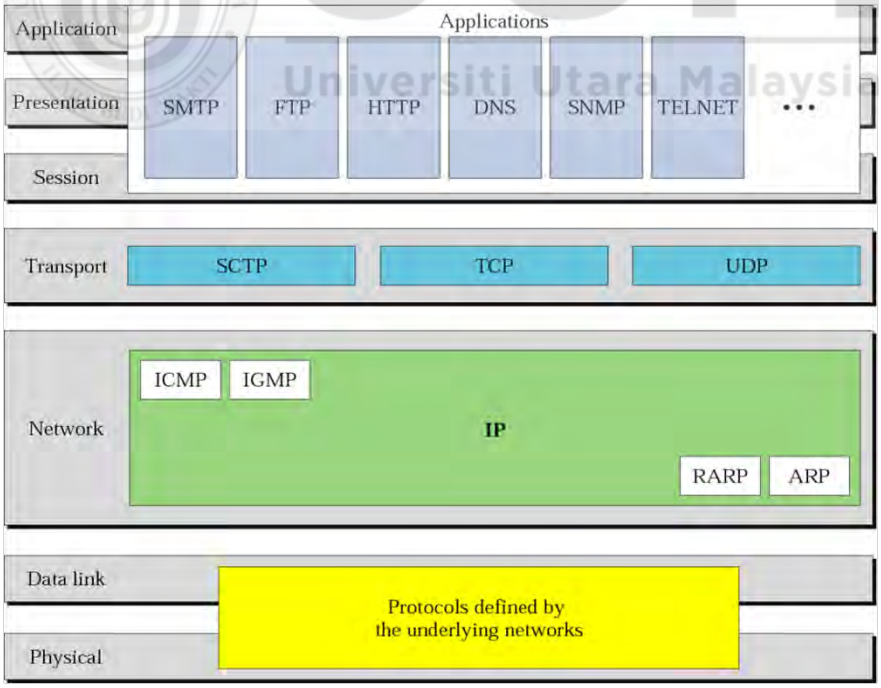


Figure 2.1: TCP/IP and OSI model [16]

2.2 Layers Components And Functions

This section presents information about the four layers components and functions. [16], [18], [19]

a. The Network Access Layer

The physical layer coordinates connector, interface specification and a physical medium. It deals with electrical, mechanical, functional, and procedural specifications to transmit a bit stream on a computer network.

The network access layer components include: cabling system components, adapters that connect media to physical interfaces, hub, repeater, and patch panel specifications, Network Interface Card (NIC), Ethernet and Token Ring switches, and bridges.

b. The Internet Layer

The Internet layer provides source-to-destination delivery (hop-to-hop routing) of a packet, possibly across multiple networks (links).

Internet layer components include: Routers, and Gateways.

c. The Transport Layer

The transport layer is responsible for end-to-end communication between end devices through a network.

Transport Layer components include Routers.

d. The Application Layer

The application layer provides the actual service to the user and it is responsible for establishing, managing and terminating sessions. It translates, encrypts and compresses data, and allows access to network resources.

Application layer components include Application Gateways.

2.3 Network problems

Computer networks are an integral part of the day-to-day operation of many organizations. The massive size and complexity of the networks and the variety of network equipment make them vulnerable to problems and failures. Network problems are classified into many categories based on various factors. Research conducted have shown [20] that planned network downtime accounts for at least 80% of all downtime while the unplanned downtime account for less than 20%. Pertet and Narasimhan [21] classify the causes of failures in Web applications into four categories namely: software failures, operator error, hardware and environmental failures, and security violations. The software failures include: Resource exhaustion, Computational/logic errors, System Overload, Recovery code, Failed software upgrade. The Human/Operator Errors include: Configuration errors, Procedural errors, and Miscellaneous accidents.

Kyas [2] has classified five categories of errors that cause system failure in data processing systems. They are: Operator Error, Mass Storage Problems, Computer Hardware Problems, Software Problems, Network Problems, and he pointed out that network problems account for more than one-third of IT failures, which include hardware and software problems that are directly related to the network, Hudyma and Fels [3], added two new categories: failures due to Denial of Service Attacks and failures due to disasters such as fire, flood, earthquakes, outages and the like.

2.4 Network Troubleshooting

Many network problems are easy to identify, for example a failure between routers, and hardware. This is done by using monitoring technology such as SNMP (Simple Net-work Management Protocol) or Ethernet-OAM (Operations, Administration, and Management) [22]. Some other network problems like software bugs and failures

that are not included in the rules are not easy to notice or identify because they may not cause a network failure, but they would definitely impact its performance [2]. Therefore, there is an urgent need to detect and diagnose the causes of these problems.

Network administrators and engineers realize that troubleshooting process takes most of their time, so it is important to use efficient techniques to reduce the time and effort [13] involved. In general, there are two approaches to troubleshooting [5]. First, the theorist or rocket scientist approach which depends on analyzing and reanalyzing the situation until the exact cause of the problem has been identified and corrected with surgical precision. Next is the impractical or caveman approach, which depends on swapping cards, cables, hardware, and software until miraculously, the network begins operating again. Both of these approaches are considered extremes; the solution is to adopt a systematic approach which minimizes confusion and eliminate time wasting by way of a trial and error method [5].

2.4.1 Layered Models for Troubleshooting

The OSI and TCP/IP models separate network functionality into modular layers and they provide a common language for network engineers. It is usually used in troubleshooting networks. Problems are normally described in terms of a specific model layer, for example if the symptoms suggest a problem in the physical connection, the network technician can focus on troubleshooting the circuit that operates at the physical layer [5].

2.4.2 General Troubleshooting Procedures

There are three interrelated and integrated stages of the troubleshooting process: gathering symptoms, isolating the problem, and correcting the problem [5].

a. **Gathering symptoms** is the first stage in the troubleshooting process. The aim is to collect symptoms from the network, end systems, and users. The symptoms have many different forms such as alerts from the network management system, console messages and user complaints.
b. **Isolating the problem** involves identifying a single problem, or a set of related problems. At this stage, the network administrator can identify the cause of the problem, determine the layer which has the problem. Next, isolate and fix it.
c. **Correcting the problem** is done after identifying the causes and isolating the problems. This helps the network administrator to correct the problem by implementing, testing, and documenting a solution.

For efficient network troubleshooting, it involves combining an efficient troubleshooting approach with troubleshooting procedures. The layered model approach would be efficient if the symptoms of network problems are gathered and the causes are identified for each network layer. The next section presents the symptoms and causes of the problems for each network layer as identified by CISCO [5].

2.5 Symptoms and Causes of Network Layers Problems

2.5.1 Symptoms and Causes of The Network Access Layer Problems

Problems associated with Network Access Layer occur when physical properties of network devices are substandard, which make data transfer slow.

a. Symptoms of Network Access Layer Problems Include:

i. **Performance lower than baseline**: if performance is unsatisfactory all the time, the problem is probably related to the physical layer.
ii. **Loss of connectivity**: if a cable or device fails, the most obvious symptom is loss of connectivity between devices.
iii. **Error indicators:** error messages reported on the device console indicate a physical layer problem.

iv. **High collision counts:** related to a bad cable to a single station on a hub.

v. **Network bottlenecks or congestion:** if an interface fails, routing protocols may redirect traffic to other routes that are not designed for extra capacity.

vi. **High CPU utilization rates:** when devices, such as a router, switch, or server, exceed their design limits.

vii. **No functionality or connectivity at network layer or above:** stopping the exchange of frames across a link.

viii. **Network is operating below baseline performance levels:** frames take an illogical path to their destination and some frames are dropped.

ix. **Excessive broadcasts:** poorly programmed or configured applications, large Layer 2 broadcast domains, and underlying network problems.

b. Causes of Network Access Layer Problems

i. **Power-related: This is** the most fundamental reason for network failure. For example, the operation of the fans.

ii. **Hardware faults:** faults in network interface cards (NIC) can be caused by network transmission errors.

iii. **Cabling faults:** disconnected cables, damaged cables, and improper cable types.

iv. **Attenuation**: occurs when a cable length exceeds the design limit for the media.

v. **Interface configuration errors:** asynchronous serial links instead of synchronous, and interface not turned on.

vi. **Noise:** local electromagnetic interference (EMI) is known as noise. There are four types of noise that are significant to networks:

- Impulse noise that is caused by voltage fluctuations or current spikes induced on the cabling.
- Random (white) noise that is generated by such as FM radio stations, police radio, and building security.
- Alien crosstalk, which is noise induced by other cables in the same pathway.

- Near end crosstalk (NEXT), which is noise originating from crosstalk from other adjacent cables or noise from nearby electric cables, devices with large electric motors or anything that includes a transmitter more powerful than a cell phone.

vii. **Exceeding designs limits**: components utilize at a higher average rate than it is configured to operate.

viii. **CPU overloads**: interfaces are overloaded with traffic.

ix. **Encapsulation errors:** encapsulation at one end of a WAN link is different of the encapsulation used at the other end.

x. **Address mapping errors:** mismatch of Layer 2 and Layer 3 addressing information, fail of the mapping of Layer 2 and Layer 3 information.

xi. **Framing errors:** the frame does not end on an 8-bit byte boundary.

xii. **STP failures or loops:** forwarding loops, Excessive flooding, Slow STP convergence.

2.5.2 Symptoms and Causes of Internet Layer Problems

i. The Symptoms of Internet layer problems are: network failure, and network performance below the baseline.

ii. The causes of network layer problems involve multiple layers or even the host computer itself.

2.5.3 Symptoms and Causes of Transport Layer Problems

These problems come from the router, particularly at the edge of the network, where security technologies are examining and modifying the traffic.

a) The symptoms of transport layer problems include: intermittent network problems, security problems, address translation problems, and problems with specific traffic types.

b) The causes of transport layer problems are related to the security technologies that implemented on the network. The most commonly

implemented security technologies are, Access Control Lists (ACL) and Network Address Translation (NAT).

i. Access Control Lists (ACL) Issues

The most common issues with ACLs are caused by misconfigurations, and this commonly occurs in eight areas: incorrect traffic direction, incorrect control element order, implicit deny any, addresses and wildcard masks, Transmission Control Protocol (**TCP**) and User Datagram Protocol (**UDP**) TCP/UDP selection, source and destination ports, use of the established keyword, and uncommon protocols.

ii. Network Address Translation (NAT) Issues.

A common problem with NAT technologies is interoperability with other network technologies. **Some common NAT issues are;** interoperability issues, incorrect static NAT, improperly configured NAT timers.

Some of the problematic NAT technologies are:

- The bootstrap protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) have difficulty operating over a router running either static or dynamic NAT.
- Domain Name Servers (DNS) or Windows Internet Name Service (WINS) server outside the NAT router does not have an accurate representation of the network inside the router.
- Simple Network Management Protocol (SNMP) management station on one side of a NAT router may not be able to contact SNMP agents on the other side of the NAT router.
- Tunneling and encryption protocols use specific UDP or TCP port, or use a protocol at the transport layer that NAT cannot process.

2.5.4 Symptoms and Causes of Application Layer Problems

Problems at application layer occur when the physical, data link, network, and transport layers are functional, but the resources are unreachable or unusable, or the data transfer and requests for network services from a single network service or application do not meet a user's normal expectations.

The symptoms of application layer problems are; no network services are available, user complaints about slow application performance, application error messages, console error messages, system logs file messages, and network management system alarms".

Troubleshooting the network problems based on the layered model approach need to gather symptoms and classify them in order to identify the causes and isolate the problem [5]. There are many types of sources for symptoms and causes gathering, as collecting information from the users, and collecting data from the network devices like syslog data [1].

Table 2.1: *Summary network layers and problems*

Network layer	Problems
Layer1	Loss of connectivity, cabling fault, high collision counts.
	Network bottlenecks or congestion, hardware fault.
Network access layer	High CPU utilization rates, attenuation.
	Address mapping error.

Layer2 Internet layer	Network failure, network performance below the baseline.
	Address translation problems.
Layer3 Transport layer	Domain name server (DNS) problems.
	DHCP difficulty operating.
	SNMP contact problems.
	Access control list (ACL) problems.
	Slow application performance.
Layer4 Application layer	No network service available.

Table 2.1: *Summary network layers and problems*

2.6 Syslog data

Syslog is a service that messages reports of system state information and errors are sent to a network manager, and it is considered as an essential component of any

network operating system [19]. Syslog protocol was originally written by Eric Allman [6], it allow many network devices including routers, switches, firewalls, application servers, and other network appliances to generate and send event notification messages across IP networks to the syslog server (event message collectors). If the connectivity between these devices and the syslog server is down, syslog messages stored locally by the devices. Each vendor of network devices have its own syslog format.

In general, syslog message carries information of facility, severity, hostname, timestamp, and text message [6].

i. Syslog Facility

A service provides classification of the sources that generate syslog messages, Cisco IOS devices have more than 500 facilities represented by integers. The most common facilities are: IP, Open Shorted Path First (OSPF), SYS Operating System, IP security (IPsec), Rout Switch Processor (RSP), Interface (IF) [19] .

ii. Syslog Severity

The facility that generates the syslog message also specifies the priority or severity level of the message by a single-digit integer. In general there are 8 levels of severity:

- Level 0 represent emergency messages (system is unusable).
- Level 1 represents alert messages (Action must be taken immediately).
- Level 2 represent critical conditions.
- Level 3 represent error messages (Error conditions).
- Level 4 represent Warning conditions.
- Level 5 represent notice messages (Normal but significant condition).
- Level 6 represent informational messages.
- Level 7 represent debug messages.

iii. Syslog hostname

A field consists of the host name as configured on the host itself or the IP address.

iv. Syslog timestamp

Is the local time of the device when the message was generated.

v. Syslog message

The text message that describe the event and contain details of the resource, port number, and network address.

Examples of syslog messages

- Cisco Syslog Message Format: %PIX|ASA-1-101002: (Primary) Bad failover cable.
- IBM Syslog Message Format: DFHAP1901 04/04/2014 17:03:25 CMAS01 SPI audit log is available.

Syslog servers contain a huge number of messages related to network problems, and these messages have been sent from all network devices that are from various vendors and with different syslog format, and in a textual format. These properties make syslog data identified as un structured data [23]–[26].

2.7 Unstructured Data Analysis

Unstructured data is data that are not already coded in terms of analytical categories, it can take many forms like word documents, spread sheets, email message, images, log files [27]. The usage of unstructured data has increased rapidly in the present days; therefor, it has become in strong need to analyse [10]. The nature of unstructured data, with lacking structure makes more challenging for mining than structured data [28]. Data Mining is knowledge detection and resolution process of databases [29], [30]. It is the process of semi automatically and analyzing large databases to find useful patterns [28]. So, data mining knowledge discovery in

databases is looking for patterns in data [31], for example, text mining looking for patterns in text. Text mining is a variation on a field called data mining, which is the process of analyzing text to extract information that is useful for particular purposes [32]. Text mining functionality is similar to data mining, but text mining can work with unstructured data such as PDF, and text files, or semistructured data sets such as emails, XML and HTML files and etc. So, text mining is a superior way for log data classification, since it is saved as text or in text files [33].

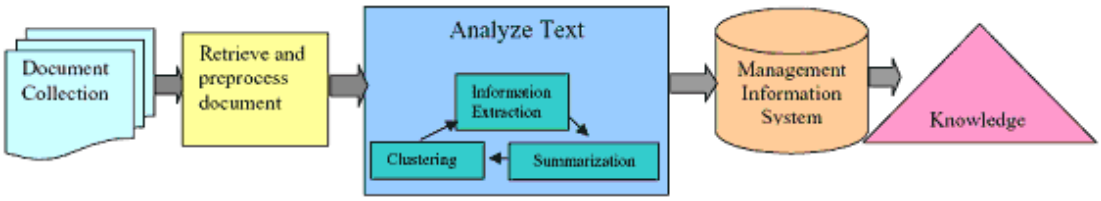


Figure 2.2: Text Mining Processes [33]

2.8 Text Data Classification

Text classification (TC) is a supervised learning approach where the task is to assign a given text document to one or more predefined categories from a set of pre-defined categories, based on its contents [34], [35]. The first step of text document classification is the transformation of documents into an appropriate representation for the classifier, and the second step is learning; the system learns to classify documents according to a ranking model where the classes are predetermined and training examples are known and correctly labeled in advance [36]. The first step aims to reduce the space of representation of documents. It usually includes pre-processing operations like removing stop words, lemmatization, selection and weighting of the descriptors. The second step involves two phases, a training phase and a prediction phase [8]. Training phase includes feature selection, class labels identification, and building the classifier using machine learning algorithm and classification rules. While prediction phase applies the classifier on the data sets. The main aspects in building the classifier are; creating feature vector for text, feature selection, and learning the classifier using text classification algorithm.

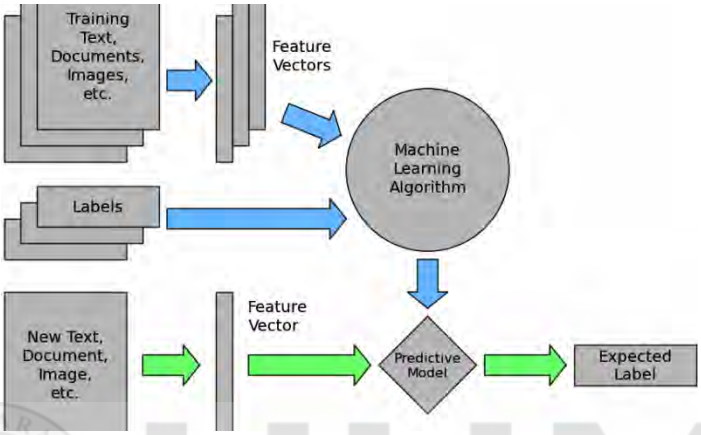


Figure 2.3: Text Classification Processes

2.9 Machine Learning Techniques

A lot of algorithms were developed for machine learning technique, supervised learning techniques are normally used for text classification, where the class labels are defined early [37], the most populer algorithms for text classification are [38]; Support Vector Machines (SVM) [14], Naive Bayes (NB), K-Nearest Neighbour (k-NN) [39], and Decision Tree algorithms [40].

2.9.1 Support Vector Machine (SVM)

Support vector machine is a classifier, which proposed by Vapnik [38]. It is a supervised learning technique from the field of machine learning applicable to both classification and regression, by applying linear classification techniques to non-linear data [41]. SVM is a machine learning technique, which is based on "structural

risk minimization principle" from the "computational learning theory". Introduced by Lapnik in 1979, the SVM splits the data from training set into two classes and making decision depending on the "Support Vectors" where the effective elements are selected from the training set [42].

kernel equations could be applied on SVM concepts, it is may be any function that transforms the linearly non-separable data in one domain into another domain where the instances become linearly separable. Kernel equations may be linear, quadratic, Gaussian, or anything else that achieves this particular purpose [38]. SVM is only applicable for binary classification tasks, it classifies a vector d to either –l or 1 [14].

$$d_{i=} \begin{cases} 1, & if \text{ the i document contains the selected features} \\ -1, & otherwise \end{cases} \quad (2.1)$$

Syslog data classification in terms of network layers, needs to classify syslog messages to four categories, therefore, there is a need to treat multi-class problems in SVM binary classification. There are two main methodes have been proposed for this purpos; one-against-one method [43], one-against-rest method [44].

2.9.1.1 One-Against-One

In this scheme, one SVM classifier, $SVM_{i,j}$, is constructed for every pair of classes (i,j). There are N(N +1)/2 SVM classifiers in total. As the study has four classes; there are 4(4 +1)/2 SVM classifiers; that's mean there are ten SVM classifiers. For training data from the ith and the jth classes [45],

$$\min_{w^{ij},b^{ij},\xi^{ij}} \quad \frac{1}{2}\left(w^{ij}\right)^T w^{ij} + C\sum_t \xi_t^{ij}$$

$$\left(w^{ij}\right)^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \ if y_t = i,$$

$$\left(w^{ij}\right)^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \ if y_t = j,$$

$$\xi_t^{ij} \geq 0. \qquad (2.2)$$

For prediction stage, the class for the test samples are determined by a voting scheme, if sign $\left(\left(w^{ij}\right)^T \phi(x_t) + b^{ij}\right)$ says *x* is in the *i*th class, then the vote for the *i*th class is added by one. Otherwise, the *j*th is increased by one. Then *x* is predicted in the class with the largest vote [45].

Sequential minimal optimization (SMO) is an algorithm that apply the concepts of one-against-one method.

```
Start
    Input:
        T: training set;
        F: feature set;
        K: size of feature space;
        D: size of feature subspace;
        C: number of classes;
        L: number of feature subspce;
        I: baseline leanear;
        ẋ: one test sample;
    Output:
        h(ẋ): which is the class label of the test sample ẋ
    process:
        for i= 1:C do
            label the samples of ith class as positive and the rest samples as negative;
            external L diverse training subset by feature subspaces;
            for j= 1:L do
                train imbalanced base clssifier I_{i,j} by training subset T_{i,j} ;
            end for
        end for
        for i= 1:C do
            for j= 1:L do
                use I_{i,j} to classify the test sample ẋ;
            end for
            calculate the value of counter i;
        end for
    output h(ẋ);
End
```

Algorithm 2.1: Pseudo code of the SMO algorithm implementation [45]

2.9.1.2 One-Against-Rest

It constructs *k* SVM models where *k* is the number of classes, this study has four classes (models). The *i*th SVM is trained with all of the examples in the *i*th class with positive labels, and all other examples with negative labels. Thus given *l* training data $(x_1,y_1),\ldots,(x_l,y_l)$, where $x_i \in R^n$, $i = 1,\ldots,l$ and $y_i \in \{1,\ldots,k\}$ is the class of x_i [45].

$$\min_{w^i,b^i,\xi^i} \quad \frac{1}{2}(w^i)^T w^i + C\sum_{j=1}\xi_j^i \;\; (w^i)^T$$

$$(w^i)^T \phi(x_j) + b^i \geq 1 - \xi_j^i, \qquad if\, y_i = i,$$

$$(w^i)^T \phi(x_j) + b^i \leq -1 + \xi_j^i, \qquad if\, y_i \neq i,$$

$$\xi_j^i \geq 0, \quad j = 1,\ldots,l \qquad (2.3)$$

where the training data x_i are mapped to higher dimensional space by the function ϕ and *C* is the penalty parameter.

Library for Support Vector Machines (LibSVM) is an algorithm that apply the concepts of one-against-All method.

2.9.2 K-Nearest Neighbor

The KNN is an algorithm, which classifies objects based on the distance between objects. Its simplicity, make it a widely employed technique for text classification. The KNN has a good performance even when multi-classification documents are used. However, under large training examples, the KNN requires much longer time to perform text classifications. KNN should select objects from training set by calculating the distance between objects from training examples [46].

In K-NN, examples are classified based on the class of their nearest neighbours,

that's mean the intuition underlying Nearest Neighbour Classification is quite straightforward. It is often useful to take more than one neighbour into account so the technique is more commonly referred to as k-Nearest Neighbour (k-NN) Classification where k nearest neighbours are used in determining the class. Since the training examples are needed at run-time, i.e. they need to be in memory at run-time, it is sometimes also called Memory-Based Classification. Because induction is delayed to run time, it is considered a Lazy Learning technique [47].

K-Nearest Neighbor classifier algorithm ranks the document's neighbors among the training document vectors, and uses the class labels of the k most similar neighbor to predict the class of the new document. The classes of these neighbors are weighted using the similarity of each neighbor to *X*, where similarity is be measured by Euclidean distance or the cosine value between two document vectors. The cosine similarity is defined as [48]:

$$sim\ (X, D_j) = \frac{\sum_{t_i \in (X \cap D_j)} x_i \times d_{ij}}{\|X\|_2 \times \|D_j\|_2} \tag{2.4}$$

Where *X* is the test document; D_j *is* the *jth* training document; t_i is a word shared by *X* and D_j; x_i is the weight of word t_i in *X*; d_{ij} is the weight of word t_i in document D_j; $\|X\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2 + \cdots}$ is the norm of *X*, and $\|D_j\|_2$ is the norm of D_j [48] .

BEGIN
Input: $D = \{(x_1, c_1), \ldots, (x_n, c_n)\}$ training dataset
$x = (x_1, \ldots, x_n)$ new instance (syslog message) to be classified
FOR each labeled instance (x_i, c_i) calculate d (x_i, x)
Order d (x_i, x) from lowest to highest, $(i = 1, \ldots, n)$
Select the K nearest instances to x: D_x^k
Assign to x the most frequent class in D_x^k
END

Algorithm 2.2:Pseudo code of the kNN algorithm implementation [47]

2.9.3 Decision Trees

Another technique used in text classification called Decision Tree, which is a classifier expressed as a recursive partition of the instance space. A Decision Tree consists of internal nodes. For each node on Decision Tree, set of terms is defined. Branches departing from these internal nodes are assigned with terms from text document while the leaves label the classes. The Decision Tree is constructed using the divide and conquer strategy where set of cases is associated with each node. Under this strategy, all training examples pose the same label. If a training example has a different label then a term will be selected from the pooled classes of documents, which carries the same values. This class is then placed on a separate sub-tree [46].

"Decision Trees" are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The main problem in DT classification algorithm is how to construct the optimal classifier. Generally, may DT classifier can be built from a set of features. In classification task when the size of search space is exponential, the accuracy of some trees are more precise than the others, and it is computationally infeasible to find the optimal tree. However, different algorithms have been developed to construct a reasonably accurate, "albeit suboptimal", "decision tree" in a practical time and efficiently. These algorithms usually use a greedy strategy that develops a "decision tree" by making a series of locally optimum decisions about which attribute to use for partitioning the data. For example; Hunt's algorithm, ID3, C4.5, CART, and SPRINT are greedy decision tree induction algorithms [49].

DTs are easy to interpret and understand. DT can be imagined, need a few data training and does not support missing values. The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree [50]. There are many of algorithms that apply decision tree concepts.

2.9.3.1 J48 Algorithm

Decision tree J48 implements Quinlan's C4.5 algorithm for generating a pruned or unpruned C4.5 tree. C4.5 is an extension of Quinlan's earlier ID3 algorithm. J48 builds decision trees from a set of labelled training data using the concept of information entropy. It uses the fact that each attribute of the data can be used to make a decision by splitting the data into smaller subsets. "Entropy" which is a measure of the data disorder, is calculated for $\vec{y}$ by [51]

$$Entropy\ (\vec{y}) = -\sum_{j=1}^{n} \frac{|y_i|}{|\vec{y}|} log(\frac{|y_i|}{|\vec{y}|})$$

$$Entropy\ (^{j}/_{\vec{y}}) = \frac{|y_i|}{|\vec{y}|} log(\frac{|y_i|}{|\vec{y}|})$$

And Gain is

START

"Check for any base cases;
for each attribute a do
find the feature that best divides the training data such as information gain rom splitting on a;
end for
let a_{best} *be the attribute with the highest normalized information gain;*
create a decision node that splits on a_{best}*;*
recourse on the sub lists obtained by splitting on a_{best}*, and add those nodes as children of node;*
sStop when the stopping condition is met;"

END

$$Gain\ (\vec{y}, j) = Entropy\ (\vec{y} - Entropy\ (^{j}/_{\vec{y}})) \quad (2.5)$$

Algorithm 2.3: Pseudo code of the J48 algorithm implementation [51]

2.9.3.2 RF Algorithm Implementation

Random Forests (RF) is an ensemble learning method for classification and regression [51]. The basic notion of the methodology is the construction of a group of decision trees. RF employs two sources of randomness in its operational procedures:

I. Each decision tree is grown on a different bootstrap sample drawn randomly from the training data.
II. At each node split during the construction of a decision tree, a random subset of *m* variables is selected from the original variable set and the best split based on these *m* variables is used.

For an unknown case, the predictions of the trees that are constructed by the RF are aggregated (majority voting for classification / averaging for regression). For a RF consisting of *N* trees, the following equation is used for predicting the class label l of a case *y* through majority voting [52]:

$$l(y) = argmax_c(\sum_{n=1}^{N} T_{h_n}(y) = c)$$

Where *I* the indicator function and h_n the *nth* tree of the RF.

```
Start
    To generate c classifiers:
    for i = 1 to c do
        Randomly sample the training data D with replacement to produce Di;
        Create a root node, Ni containing Di;
        Call BuildTree( Ni );
    end for

    Build tree (N):
    if N contains instances of only one class then
        return
    else
        Randomly select x% of the possible splitting features in N
        Select the feature F with the highest information gain to split on
        Create f child nodes of N , N1 ,..., N f , where F has f possible values ( F1 , ... , Ff )

         for i = 1 to f do
            Set the contents of Ni to Di , where Di is all instances in N that match Fi
            Call BuildTree( Ni )
         end for
    end if
End
```

Figure 2.7: Pseudo Code of RF Algorithm Implementation

2.9.4 Naïve Bayes Algorithm (NB)

Naïve Bayes classifier is a simple probabilistic classifier based on applying Bayes Theorem with strong independent assumptions. In this algorithm, "posterior probability" is calculated to each document belonging to different classes. The document is classified according to class which has the highest "posterior probability". This model is known as independent feature model because the presence of some feature will not effect the other features [54].

A “naive Bayes” classifier suggests that the presence (or absence) of a specific feature of a class is independent of the presence (or absence) of other feature. “Naive Bayes” classifiers are efficiently trained in a “supervised learning” setting, depending on the probability model. In many practical applications, parameter estimation for “Naive Bayes” models uses the method of maximum likelihood; in other words, one can work with the “Naive Bayes” model without believing in Bayesian probability” or using any “Bayesian” methods [49].

A “Bayes classifier” is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. Analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of naive Bayes classifiers. Still, a comprehensive comparison with other classification methods have shown that Bayes classification is outperformed by more current approaches, such as boosted trees or random forests [55]. In NB classifier, each document is viewed as a collection of words and the order of words is considered irrelevant. The probability of a class value *c* given a test document *d* is computed as [56]:

$$P(c/d) = \frac{P(c)\prod_{w\in d} P(^{w}/_{c})^{n_{wd}}}{P(d)}, \tag{2.7}$$

where n_{wd} is the number of times word *w* occurs in document *d*, *P(w/c)* is the probability of observing word *w* given class *c*, *P(c)* is the prior probability of class *c*, and *P(d)* is a constant that makes the probabilities for the different classes sum to one. *P(c)* is estimated by the proportion of training documents pertaining to class *c* and *P(w/c)* is estimated as [56]:

$$P(w/c) = \frac{1+\sum_{d\in D_c} n_{wd}}{k+\sum_{w'}\sum_{d\in D_c} n_{w'd}}, \tag{2.8}$$

where D_c is the collection of all training documents in class *c*, and *k* is the size of the vocabulary (i.e. the number of distinct words in all training documents).

Start

Begin training NB (C, D)

V, extract vocabulary (D)

N, count documents (D)

for each $c \in C$

do N_c *, count document in class (D, C)*

prior [c] , N_c / N

$text_c$ *, concatentate(concentrated) text of all documents in class (D, C)*

for each $t \in V$

do T_{ct} **,** *count tokens of term (*$text_c$ *, t)*

for each $t \in V$

do condprob [t][c], $\frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$

return V, prior, condprob

End training NB (C, D)

Begin applying NB (C, V, prior, condprob, d)

W, extract tokens from document (V, D)

for each $c \in C$

do score [c] , log prior [c]

for each $t \in W$

do score [c] += log condprob [t][c]

return arg max $_{c \in C}$ *score[c]*

End applying NB (C, V, prior, condprob, d)

End

Algorithm 2.5: Pseudo code of the NB algorithm implementation [56]

2.10 Approaches to Create Feature Vector for Text Classification

Document representation is one of the key components for determining the text classification effectiveness [57]. Traditional representation of documents, known as "Bag of Word" (BOW), considers every document as a vector in a very high dimensional space where each element of this vector represents one term if it appears in the document collection. This method of document representation is called a Vector Space Model (VSM), where each document is represented as a vector, and vector components represent certain feature weights [58]. The transformation of a document set D into the BOW representation enables the transformed set to be viewed as a matrix, where rows represent document vectors, and columns are terms [59]. There are several methods to identify the term of vector, and several methods to represent feature weights.

- **Methods of Vector Term Identification**

Part of Speech (POS) is a method that identifies the vector term, in terms of part of speech - a key term in any book about grammar, and even any dictionary, for that matter. Common examples of a word's part of speech include noun, verb, adjective, and so on [60]. N-gram is also a method, which is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application [49]. The traditional document representation is another method, which considers unique words as the components of vectors [59].

- **Methods of Feature Weights Representation**

There are several ways of determining the weight of word i in document k, such as Boolean weighting that represents the presence or absence of the word. Term frequency weighting that represents the frequency of the ith term in the nth document (the number of term presence in a document) [61]. Term Frequency Inverse Document Frequency (TF.IDF) is used to calculate the importance of a word in a collection of documents. The weight of the word increases depending on the number of times when the word appears in the document. This weight will decrease when the word appears many times in different documents [49].

The traditional document representation considers unique words as the components of vectors and they are known as "Bag of Word" (BOW). It considers every document as a vector in a very high dimensional space where each element of this vector represents one word which appears in the document collection. This method of document representation is called a Vector Space Model (VSM), where each document is represented as vector, and vector components represent certain feature weights [36]. The transformation of a document set D into the BOW representation enables the transformed set to be viewed as a matrix, where rows represent document vectors, and columns are words [37].

Let W be the dictionary – the set of all words that occur at least once in a collection

of documents D. The bag-of-words representation of document d_n is a vector of weights (w_{1n},. . . ,$w_{|W|n}$). In the simplest case, the weights $w_{in} \in \{0, 1\}$ denote the presence or absence of a particular term in a document [61].

2.11 Feature Selection Methods

Feature selection or feature extraction is a primitive task and a key step for text classification because it can reduce the dimensionality of feature space, decrease the computing complexity and improve the accuracy rate of classification [14]. A number of feature selection methods has been applied to text classification, including Document Frequency (DF), which is the number of documents a term occurs [36],[37]. Information Gain (IG) refers to the amount of information acquired for class prediction. This is achieved by noting the existence or not of a term in the sampled document [64]. Mutual Information (MI) is used to represent the correlation between two variables such as feature and class [65]. Chi2-test (CHI) measures the lack of independence between the term and the class [66].

2.11.1 Document Frequency Thresholding (DF)

Document Frequency (DF) is the number of documents in which a term occurs. It is the simplest technique for feature reduction, and has shown to behave well when compared to some other methods such as IG, and MI [63]. It is based on the assumption that infrequent terms are not reliable in text categorization and may degrade the performance. Each term from the training set represents a class from which the DF is calculated. Then all terms with DF values which are less than predefined threshold value are removed. The removal is due to the basic assumption that "rare words are, either not informative for class prediction, or not influential in global performance" [62].

Document frequency could be computed using the following formula.

$$tf_{t,d} = \sum_{x \epsilon d} ft(x) \quad \text{where} \quad ft(x) = \begin{cases} 1, & x = t \\ 0, & x \neq t \end{cases} \tag{2.9}$$

where $tf_{t,d}$ represents the frequency of any term t in document d.

2.12 Previous Works which Used Document Frequency for Reducing Dimension

For text classification, many works have tried to reduce the dimension of data by using DF feature selection. Yan Li & Chen, [67] proposed a method called "High Term Frequency and Weighted Document Frequency" (HTF-WDF) to enhance DF method. The HTF-WDF uses SVM to perform text classification. Chinese documents are selected to create dataset for training and testing the proposed method.

Term Frequency and Category Relevancy Factor (TFCRF), presented by Maleki [68], is a new method for feature weighting, specifically designed for text classification. The TFCRF uses weight of a feature as a function of its distribution within different documents. It also uses DF threshold method for feature selection, and SVM as classification algorithm. The proposed method produced good result.

Gang and Jiancang, [69] evaluated the performance of feature selection using SVM classification algorithm. The authors carried out experiments centered on the testing the performance of DF, Chi, DF+Chi feature selection. Document frequency was chosen as the best way in SVM algorithm.

Xia et al, [70] proposed "Text Categorization Method Based on Local Document Frequency" (TCBLDF) method for text classification. In order to reduce high dimensionality, DF feature selection was implemented before training algorithm. The proposed method got better results compared to SVM and Naive Bayes.

Yusof and Hui, [71] used "Artificial Neural Network" ANN and employed DF methods and "Class Frequency Document Frequency" CF-DF feature reduction methods, and applied "Bloom's taxonomy" to classify question items. The

experiments conducted showed that the proposed method enhanced the performance of classification in terms of time.

Harrag et al, [72] presented and compared the results achieved from Arabic text collection using "Dimension Reduction techniques" with "Back-Propagation Neural Network" (BPNN) algorithm. "Stemming", "Light-Stemming", DF, TF.IDF and "Latent Semantic Indexing" (LSI) methods were used to reduce the feature size. The results indicate that the proposed method achieved high performance in terms of Macro-Average F1 measure for Arabic text classification. Experiments on Arabic datasets show that the TF.IDF, DF and LSI approaches are favourable in terms of their efficiency.

Table 2.2: *Literature summary on text classification methods*

AUTHOR	FEATURES SELECTION TECHNIQUE	CLASSIFICATION ALGORITHM	DATA SET	OBJECTIVE	RESULTS
Yan Li & Chen, (2012)	HTF-WDF	SVM	Chinese text documents	To enhance DF method.	Weighted Document Frequency algorithm enhances the performance of text classification in term of accuracy, but was more time consuming for training classifier.
Maleki, (2010)	TFCRF & DF	SVM	Articles sourced from IEEE.	Feature weighting by considering the distribution of a feature in the document,	Significant improvement in the performance of SVM algorithm by using

				and class information.	TFCRF, but was more time consuming for training classifier.
Gang and Jiancang, (2009)	DF, Chi, DF+Chi	SVM	Document.	To research SVM and the used methods with SVM.	Document frequency was chosen as the best way in SVM algorithm, but was more time consuming for training classifier.
Xia et al, (2009)	DF	Employ a binary weighting method. (TCBLDF)	Document	To propose a fast and effective text categorization method named TCBLDF.	The dimension was reduced, but the use of binary weights is too limiting and proposes a framework in which partial matching is possible.

Yusof and Hui, (2010)	DF	ANN	"The Bloom‟s cognitive level".	To propose a classification model for the cognitive level of question items in examinations based on Bloom’s taxonomy.	The proposed method enhanced the performance of classification in terms of time.
Harrag et al, (2010)	Light-Stemming, DF, TF.IDF and (LSI).	(BPNN)	Arabic text collection.	To compare five Dimension Reduction Techniques in the context of the Arabic text classification problem.	The proposed method achieved high performance in terms of Macro-Average F1 measure for Arabic text classification.

Liparas et al, (2014)	N-gram, Visual Features	RF	News articles.	To classify news articles.	Using N-gram textual features alone led to much better accuracy results than using the visual features alone. However, the use of both N-gram textual features and visual features led to slightly better accuracy results.

2.13 Previous Works on Syslog Data Analysis

Currently, there are many applications being used to manage network, particularly in handling network problems. These applications utilise various types of data such as traffic measurement data, network configuration data, syslog data [11], and social network services (SNS) data [11], [73]. The important information in syslog data, and the big volume it collects , makes it a valuable tool to detect and diagnose network problems, and to address security issues. Several methods are involved in the process. Data mining techniques have a significant role in text data analysis, among which are to improve the ability of syslog messages classification and to enhance the field of network management.

There are many works that have studied network problems detection from log data using data mining techniques. Yamanishi & Maruyama [74], studied syslog behaviour to identify syslog messages that emerged when anomalous events occurred. They used a mixture of Hidden Markov Models to represent syslog behaviour, and an on-line discounting learning algorithm. Lim et al. [75], used a combination of data mining and statistical analysis techniques on the logs obtained from enterprise telephony system to identify the signature of a log file by determining the type of messages, the frequency of each message type, and the distribution of the message types over time, then clustering similar messages together.

Xu et al. [13], detected system runtime problems by mining console logs. They converted free-text console logs into numerical features which they then analysed using Principal Component Analysis (PCA), an unsupervised learning algorithm to detect operational problems. Qiu et al. [9], designed an automated tool syslog digest that transform massive volume of routers syslog messages into much smaller number of meaningful network events, then they identified the signature of syslog messages that captured network behaviour over time, and grouped them based on their nature and severities.

Fukuda [12], used syslog messages to detect unusual (anomalous) events in a network, by using a global weight, based on a global appearance of a message type in the all data set. For the same purpose, Martinez et al. [76], analysed all free form text fields hardware, software and users using search engine, information retrieve, with mining algorithm. They focused only on textual data to extract valuable intelligence for network analysis and troubleshooting.

Xu et al. [77], created features that capture various correlations among different types of log messages to detect anomaly in syslog behaviour, using Principal Component Analysis (PCA) learning algorithm. They validated their approach using Darkstar online game server and Hadoop file system to analyze console log. Kimura et al. [11], analysed two types of data, SNS data from tweeter and syslog messages, to detect and diagnose network failure. They used non-negative matrix factorization (NMF) machine learning algorithm to analyse syslog messages, and support vector machine to analyse tweeter messages.

The limitation with all the previous works is that they only tried to detect network problems but they didn't try to classify the problems for more efficient maintenance decisions and troubleshooting. This proposal presents a method to analyze syslog data to detect network problems and diagnose their causes, and classify these problems in terms of network layers.

Table 2.3: *Literature summary on syslog data analysis*

AUTHOR	ANALYSIS TECHNIQUE	DATA SET	OBJECTIVE	RESULTS
Yamanishi & Maruyama (2005)	Hidden Markov Models & On-Line Discounting Learning Algorithm.	Syslog data	To identify syslog messages that emerge when anomalous events occur	The methodology can be straightforwardly applied to analyse a wide range of event log files, including system calls, command lines, Web access logs, etc.
Lim et al, (2008)	Combination of data mining and statistical analysis techniques.	Logs obtained from enterprise telephony system.	To identify the signature of a log file by determining the type of messages, the frequency of each message type, and the distribution of the message types over time.	The analysis techniques led to the detection of several failure types and in some cases, predicted trends which lead towards failures.

Xu et al, (2010)	PCA	Console logs.	To detect system runtime problems.	Highly accurate detection results both with the offline and online algorithms.
Qiu et al, (2010)	Powerful data mining techniques.	Router syslogs data.	To design a Syslog Digest system that can automatically transform and compress such low-level minimally-structured syslog messages into meaningful and prioritized high-level network events.	Syslog Digest System without missing important incidents.
Fukuda, (2011)	Global weight that is based on a global appearance	Router log messages taken from a Japanese R&E network.	To detect unusual (anomalous) events in a network.	The use of two types of global weights could successfully highlight the anomalous time periods in the given time series.

Martinez et al, (2015)	Search engine, information retrieve, text analyzer and a mining algorithm.	All free form text fields; hardware & software.	To propose a text analytic based network anomaly detection approach	ADAMANT algorithm is independent of the number of documents or time interval, however has a big dependence of the number of windows.
Xu et al, (2009)	PCA, Darkstar online game server, Hadoop file system.	Textual console logs.	To mine the rich source of information (console logs) to automatically detect system runtime problems.	Numerous real problems were detected with high accuracy and few false positives.
Kimura et al, (2013)	NMF, SVM	SNS data from tweeter and syslog messages.	To diagnose and detect the causes of network failures.	Successful in detecting the cause of a network failure by automatically learning over 100 million logs, detect 1% of all tweets, in real time and with high accuracy.

CHAPTER Three

RESEARCH METHODOLOGY

This chapter presents the procedures and approaches that have been used to achieve the two objectives of the study. The study proposes a method, which involved four phases, to detect and classify network problems in terms of network layers. The phases are:

Phase one - identifying network problems and their causes for each network layer, and this has been done through the literature review.

Phase two - identifying syslog messages related to network problems for each layer, depending on phase two.

Phase three - applying classification technique on syslog data set to classify network problems in terms of network layers.

Phase four - validation results of classification method.

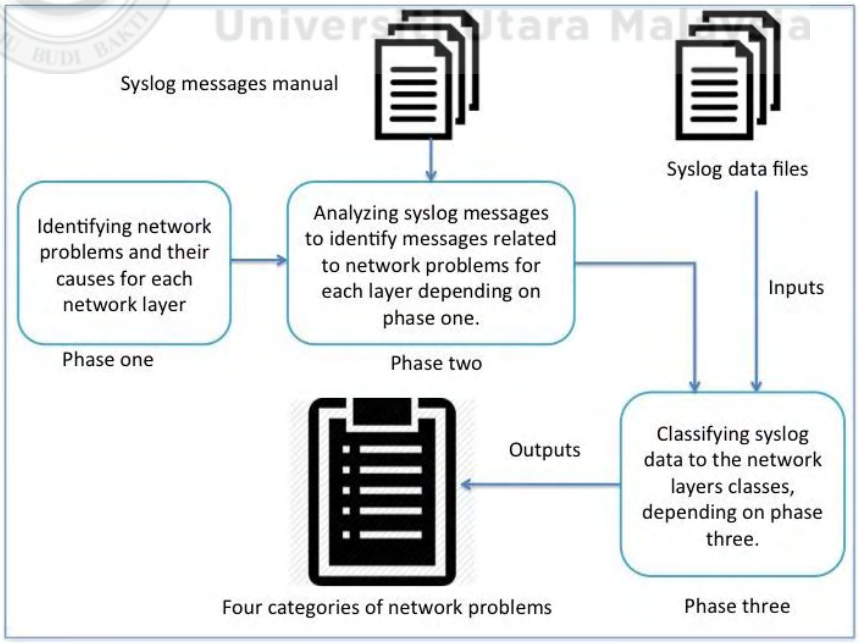


Figure 3. 1: Methodology Phases

3.1 Phase One: Network Problems Identifications

Phase one , which is the first part of the methodology, is based on the review of the relevant literature. The aim here is to identify the cause of problems for each network layer. There are four steps involved in the process which are:

1. Identifying network layers depending on TCP/IP model. This study chooses TCP/IP because it is used by Internet applications like email, World Wide Web, FTP, etc. It consists of only four layers (network access layer, Internet layer, transport layer, application layer) [17], and this is more general and efficient than OSI model, which consists of seven layers. As TCP/IP model consists of only four layers, network problems could point to more than one layer. This made TCP/IP model more efficient and easier to classify network problems to its layers.
2. Identifying components of each layer.
3. Identifying symptoms of the problems for each layer.
4. Identifying causes of the problems for each layer.

Network layers, components, problems causes and symptoms, were conducted in Chapter two. A summary of network layers and problems is conducted in the next section

3.2 Phase Two: Syslog Messages Identification

This second part explains the procedures of how to identify syslog messages which have network problems for each network layer, from syslog messages manual. Cisco syslog messages manual was used to identify related syslog messages and the result could be applied to other syslog data from different vendors, as all vendors describe network problems using almost same terms and vocabularies. In this phase, the related messages will be identified depending on the symptoms and causes of network problems related to the network access layer , identified in phase one. They include cabling faults, disconnected cables, damaged cables, and improper cable

types. The task of identifying related messages from the manual requires reading and searching the manual in order to extract them. Searching in the manual is done by using the symptoms and the causes of each problem as key words to identify related problems. Symptoms and causes of problems for each network layer have been identified in the literature review as identified by CISCO.

Loss of connectivity, performance lower than baseline, high collision counts, attenuation, bad cable, disconnected cables, damaged cables, improper cable types, cable length exceeds the design limit for the media, and cable fails are the symptoms and causes of cable problems that could be used as key words for searching in the manual, which include error message, its explanation, and the recommended action. The following are the extracted messages that describe cable problems from the Cisco manual:

%PIX|ASA-1-101001: (Primary) Failover cable OK.

%PIX|ASA-1-101002: (Primary) Bad failover cable.

%PIX|ASA-1-101003: (Primary) Failover cable not connected (this unit).

%PIX|ASA-1-101004: (Primary) Failover cable not connected (other unit).

%PIX|ASA-1-101005: (Primary) Error reading failover cable status.

These are examples of the messages that represent some network problems related to the network access layer, and they will be used in the next phase for classification purposes. The following table illustrates problems of each network layer, the key words used for searching in syslog manual, and examples of extracted syslog messages.

Table 2.2: *Summary of network problems and key words.*

Network layer	Problem	Key words
Layer1 Network access layer	Loss of connectivity, cabling fault, high collision counts.	Disconnected cable, damaged cable, improper cable, cable fault.
	Network bottlenecks or congestion, hardware fault.	Fault interface, interface fail, transmission error.
	High CPU utilization rates, attenuation.	Exceed design limit.
Layer2 Internet layer	Address mapping error.	Fail address mapping.
	Network failure, network performance below the baseline.	Network failure.
	Address translation problems.	Address translation, translation fail.
Layer3 Transport layer	Domain name server (DNS) problems.	DNS fail.
	DHCP difficulty operating.	DHCP configured fail.
	SNMP contact problems.	SNMP unable to open.
	Access control list (ACL) problems.	ACL error, ACL configuration.

Layer4 Application layer	Slow application performance.	Application fail, application stopped.
	No network service available.	FTP fail, HTTP fail.

3.3 Phase Three: Problems Classification

In this phase, syslog messages identified in phase two were processed to be used as training dataset, used to learn classification algorithms (SVM, K-NN, NB, and Decision Tree algorithms); these algorithms are the most popular text classification algorithms. Real dataset, obtained from Universiti Utara Malaysia (UUM) network devices, was used as a test dataset for prediction stage. UUM, a big public educational institution, possesses an extensive network with a large number of devices, and a massive volume of syslog data. Text classification process consists of five stages: data collection, data preprocessing, data representation, feature selection, and implementing classification algorithms [78].

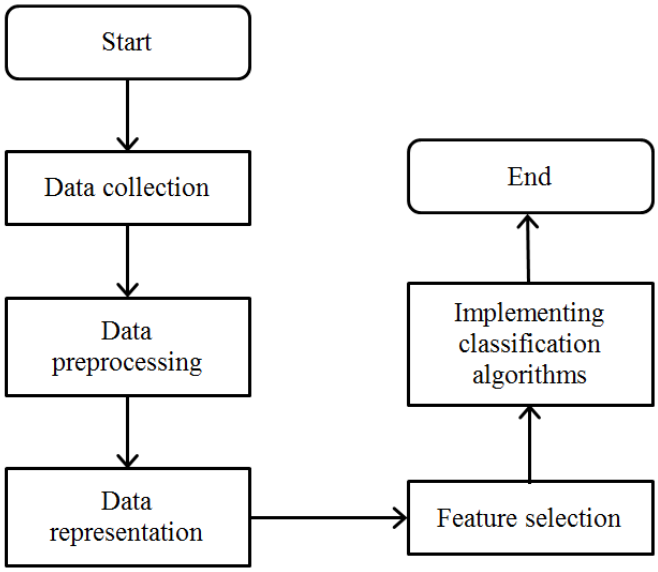


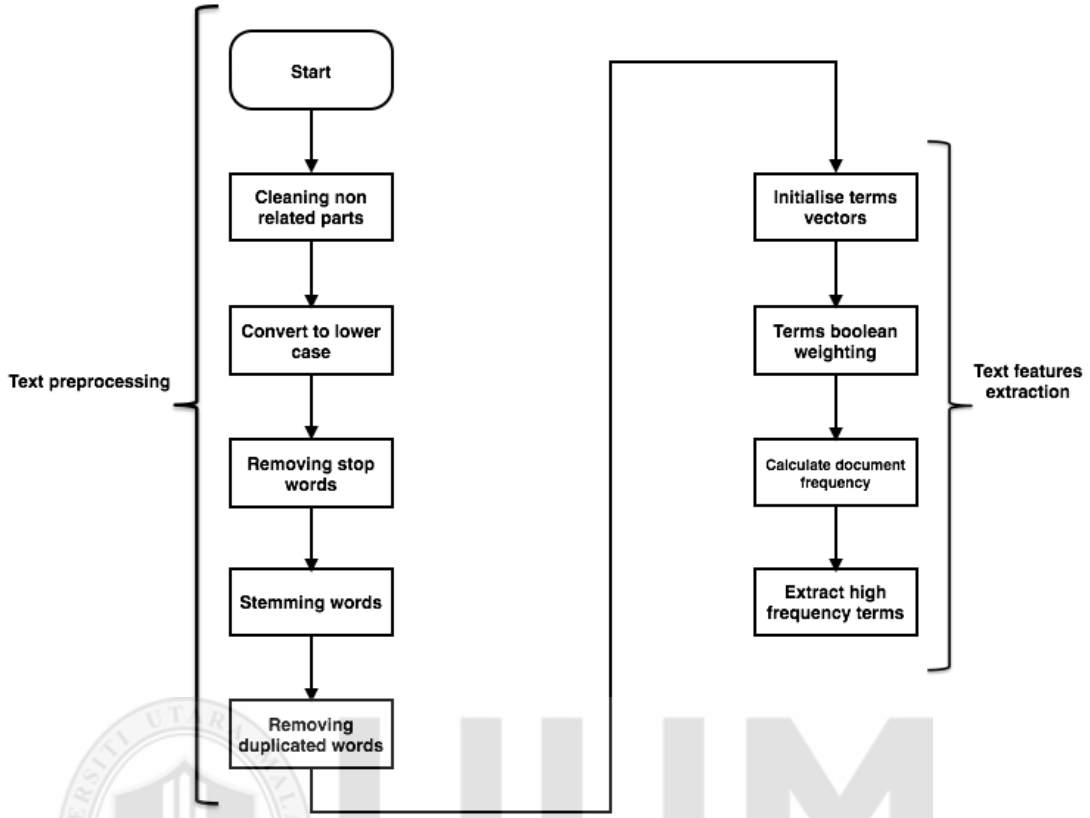
Figure 3.2: The flowchart of syslog data classification stages

3.3.1 Syslog Data Collection

Two datasets were used for classification process. The first came from training data identified from Cisco syslog manual and used in training stage to learn classification algorithms. The other dataset was taken from the computer centre of Universiti Utara Malaysia (UUM). It consisted of syslog data from network devices (firewalls, and switches) and used in prediction stage as testing dataset.

3.3.2 Syslog Data Preprocessing

The first step in text classification is to transform documents, which typically are string of characters, into a representation suitable for learning algorithm and the classification task [14]. Syslog data processing comprises four steps: cleaning non-related parts, removing stop words, stemming messages words, and removing duplicated words. Two datasets, training and testing, are processed using the same procedures.



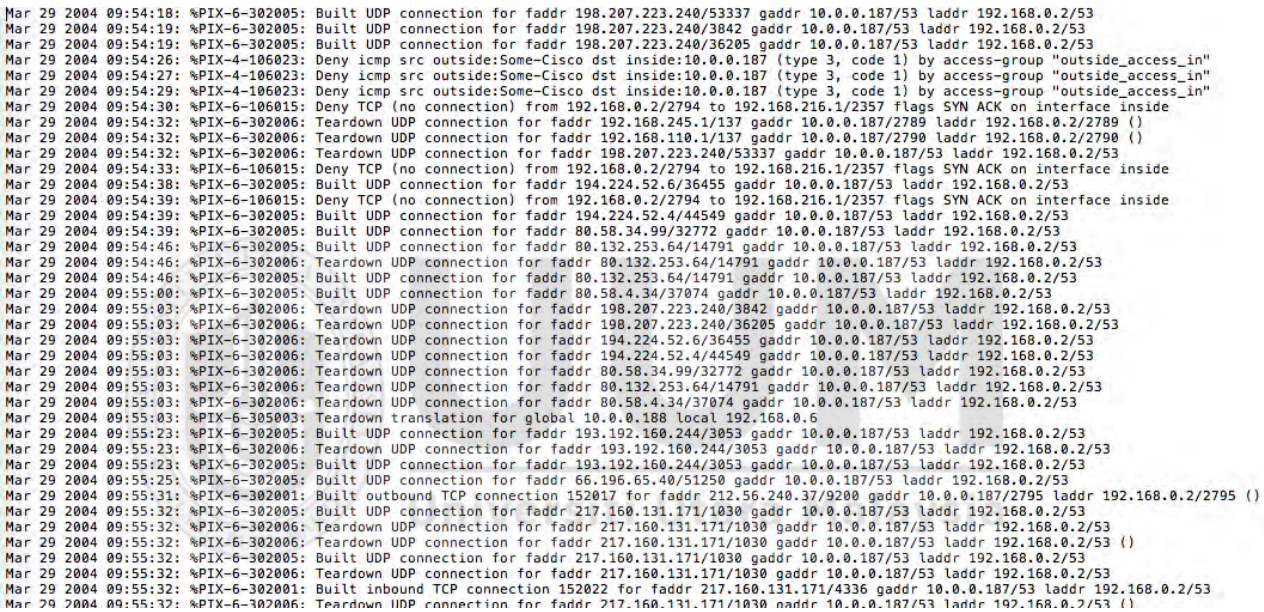

```
Start
    Execute text pre-processing
        Initialize document
        Clean noise parts
            Transform all case to lowercase
            Remove stop words using stop word removal
            Perform stemming words
        Remove duplicated words from each document
    End text pre-processing
    Start feature extraction
        Initialize term vector
        Perform term Boolean weighting
        For each term
                Calculate document frequency
        End for
        Extract high frequency terms

    End feature extraction

End
```

Figure 3.3: The flowchart of text pre-processing and feature extraction

Algorithm 3.1: The pseudo code of text pre-processing and term extraction

3.3.2.1 Cleaning Noise Parts

In general, syslog message consists of the following information: facility number, severity number, hostname, timestamp, and text message. Text message includes the related information to be classified. Therefore, the other information (facility number, severity number, hostname, timestamp) are non-related parts, and they are removed, before that. Each message is given a unique number as a reference. Then,



tags, punctuate marks, etc are removed from the text messages.

Figure 3.4: A screenshot of syslog data sample before removing non related parts

Figure 3.5: A screenshot of syslog data sample after removing non related parts

3.3.2.2 Removing stop words

Stop words or stop lists are lists of words that are removed prior to or after the processing of text according to their level of usefulness in a given context. Stop words removal is considered to be one of the important steps in text classification because it improves information retrieval and search by ignoring words that usually appear in every document and thus, are not helpful in classification process. Removing stop words reduces the index size, number of distinct words in the index, and therefore save space and time. Examples of some stop words in English include "the', "and", "a", "of" [79]. In this study, a list of stop-words was created using a set downloaded from the website "http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop" with some modifications by adding specific common words in syslog messages, and removing some words which point to a problem.

```
built connection faddr gaddr laddr
built connection faddr gaddr laddr
built connection faddr gaddr laddr
deny icmp cisco dst type code access access
deny icmp cisco dst type code access access
deny icmp cisco dst type code access access
deny tcp no connection flags syn ack interface
teardown connection faddr gaddr laddr
teardown connection faddr gaddr laddr
teardown connection faddr gaddr laddr
deny tcp no connection flags syn ack interface
built connection faddr gaddr laddr
deny tcp no connection flags syn ack interface
built connection faddr gaddr laddr
built connection faddr gaddr laddr
built connection faddr gaddr laddr
teardown connection faddr gaddr laddr
built connection faddr gaddr laddr
built connection faddr gaddr laddr
teardown connection faddr gaddr laddr
teardown connection faddr gaddr laddr
teardown connection faddr gaddr laddr
teardown connection faddr gaddr laddr
teardown connection faddr gaddr laddr
teardown connection faddr gaddr laddr
teardown connection faddr gaddr laddr
teardown translation global local
built connection faddr gaddr laddr
teardown connection faddr gaddr laddr
built connection faddr gaddr laddr
built connection faddr gaddr laddr
built outbound tcp connection faddr gaddr laddr
built connection faddr gaddr laddr
teardown connection faddr gaddr laddr
teardown connection faddr gaddr laddr
built connection faddr gaddr laddr
teardown connection faddr gaddr laddr
built inbound tcp connection faddr gaddr laddr
```

Figure 3.6: A Screenshot of Syslog Data Sample After Removing Useless Parts.

3.3.2.3 Stemming

Another important step in text classifications is stemming. Stemmers are basic elements in query systems, indexing, web search engines and information retrieval systems (IRS). It is the process of reducing words to their roots or stem in order to index the text and recognize them as the same word. For example, the words: fail, failure and failover would be recognized as one word - fail. In the field of text mining, stemming is used to group semantically related words to reduce the size of the dictionary (feature reduction) [80]. Porter Stemmer algorithm in java is used for removing suffix to generate word stem. “http://tartarus.org/martin/PorterStemmer/”.

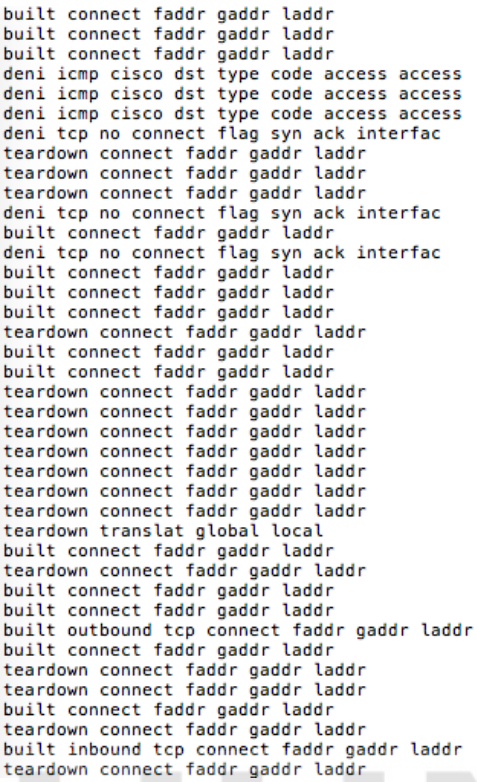


Figure 3.7: A Screenshot of Syslog Data Sample After Stemming

3.3.2.4 Removing Duplicated Words

Words space or the dictionary of text is an important issue in text classification because the small size of word space would be more efficient in text classification. As the document frequency is the responsible factor for determining the features, there would be no need for the duplicated words in the documents. Removing duplicated words from text documents decrease the word space of syslog files.

3.3.3 Syslog Data Representation

The most commonly used document representation is vector space model [78], where documents are represented by vectors of words. Usually, there is a collection of documents which is represented by a word-by-document matrix A, where each entry represents the occurrences of a word in a document, i.e. $A = (a_{td})$, where a is the weight of word t in document d [14]. There are several ways of determining the weight a of word i in document k, such as word frequency weighting, Boolean

weighting, *tf-idf* weighting, *tfc*-weighting, *itc*-weighting [81]. Boolean weighting has been used to determine the weight of words.

Firstly, words dictionary which contains all the words of relevant messages from the training sample are extracted from documents (messages). Vector space model (VSM) has been built to represent documents (d). A document d could be represented by the weight of each dictionary term: $V\sim(d) = (w(t_1, d), w(t_2, d), \ldots, w(t_n, d))$.

Figure 3.9: A Screenshot of Syslog Data Boolean Representation

3.3.4 Feature Selection

Document Frequency Thresholding (DF) method is used to extract the important features from documents. Features are high frequency terms (words) that describe the problem. Document frequency for a word is the number of documents in which the word occurs. Once the document frequency has been computed for each word, the words that document frequency less than the predetermined threshold are removed, as they are non-informative for category [14]. Java programming code is used to identify the best features that have high frequency among all documents, and generate ARFF files to be used in Weka data mining tool.

Training and testing files are generated in different numbers of features, to be used for classification algorithms. ARFF files for training and testing files are generated to be used in Weka data mining tool. The factor of features number is changed many times for each file in order to apply classification algorithms , and compare the results to specify the best number of selected features. ARFF files are generated once with all features, again with 1000 most frequency features, 550 most frequency features, 500 most frequency features, 450 features, and with 200 most frequency features. A java code is used to generate ARFF files with high frequency features, represented by Boolean weighting.

```
Start
    Initiate term dictionary
        Initialize syslog data
        Tokenize syslog instances
            Each term consists of one word
        Remove duplicated words
        Build vector space model for term dictionary
            Each term in one vector
    End term dictionary initiating
    Start Boolean weighting
        Initialize syslog data
        For each instance
            For each term vector
                If the term exist in the instance
                    Term weight = 1
                Else term weight = 0
                End if
            End for
        End for
    End Boolean weighting
    Start feature extraction
        Initialize vector space model with Boolean weighting
        For each term
            Calculate document frequency
        End for
        Identify the N highest frequency terms
        N is a variable represent the identified number of features
    End feature extraction
    Start ARFF generating
        Generate new ARFF file
        Copy the identified features
        Paste the features to the ARFF file
        Copy vector space model of the identified features
        Paste vector space model to the ARFF file
    End ARFF generating

End
```

Algorithm 3.2: The pseudo code of generating ARFF file process

ARFF is an attribute-relation file format, which is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files have two distinct sections. The first section is the Header information and the second is Data information. The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. The data contains instances or documents to be classified [82]; instances refer to the text messages (syslog messages). The used data

consists of two attributes: the document (syslog message) with string type and the class (layer1, layer2, layer3, layer4).

3.3.5 Implementing Text Classification Algorithms

The classification process was done using six text classification algorithms: SVM(SMO), LibSVM, NB, K-NN, J48 decision tree and Random Forest decision tree. The classification process consists of two stages: training stage and prediction stage.

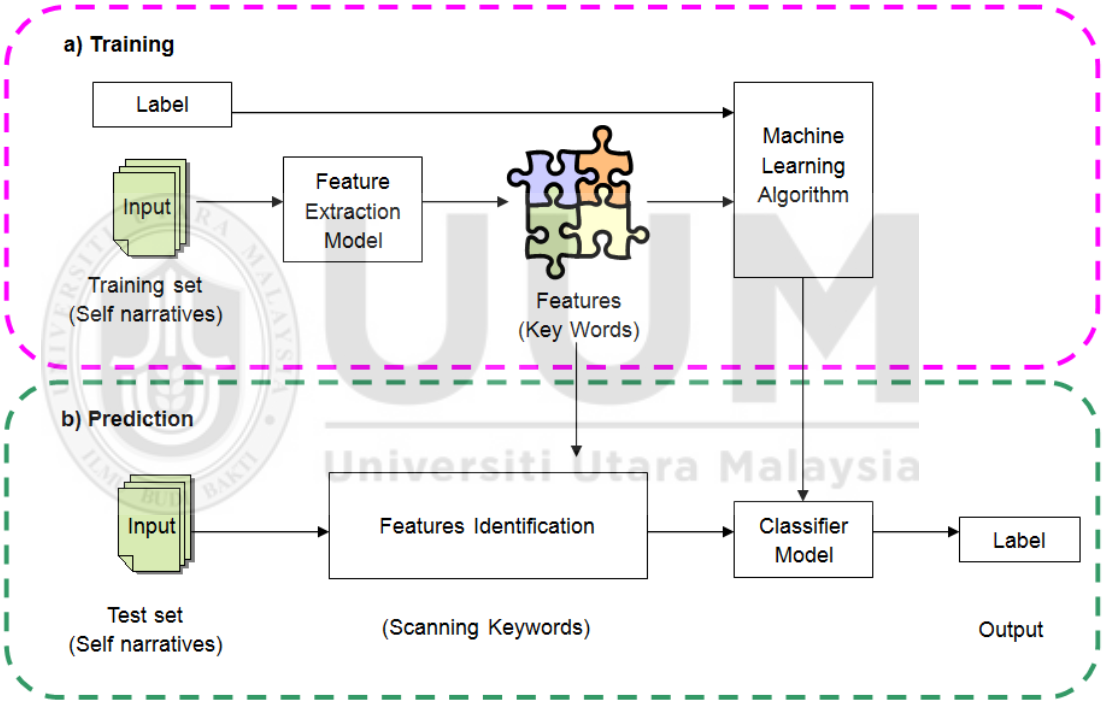


Figure 3. 10: Classification Phase [8]

3.3.5.1 Training stage

For the training stage, files of training dataset with different number of features were prepared. The training dataset came from the syslog messages extracted from Cisco syslog manual in phase two and processed (cleaned, removed stop words, stemmed, removed duplicated words). Training dataset was in ARFF extension, to be used in Weka data mining tool. Six classification algorithms were applied many times using

same dataset, but with different number of features. Results were compared to identify the best features, and the best classification model.

Firstly, six algorithms were applied to four training data files with different number of features. The first training file represents the data using all features (terms of vector space). The second training file represents the data using 1000 features that had the highest frequency. The Third training file represents the data using 500 features that had the highest frequency. The fourth training file represents the data using 200 features that had the highest frequency.

Results of six classification algorithms were compared in terms of accuracy rate in order to identify the best number of features to be used for representing data files. Accuracy rate indicates the ratio of correctly classified instances; the performance of classifier model would be better by scoring high accuracy rate. Accuracy rate was calculated using the following equation.

$$accuracy\ rate = n/N \tag{3.1}$$

Where n is the number of correctly classified instances, and N is the number of all classified instances.

The results showed that the performance of six classifiers scored the highest values by using training file that was represented with 500 features. They also showed that four classifiers: libSVM, RF. SMO, and J48 had recorded performance value higher than NB, and KNN.

These four algorithms: libSVM, RF. SMO, and J48, were relearned by another two training files with different features number. Relearning process used one file which represents the data using 550 features, and another file, using 450 features. Relearning process aims to make sure that 500 features is the best number of features to be used for representing datasets. Results of the four classification algorithms were compared in terms of accuracy rate, and the performance of four classifiers scored highest values by using training file that was represented with 500 features.

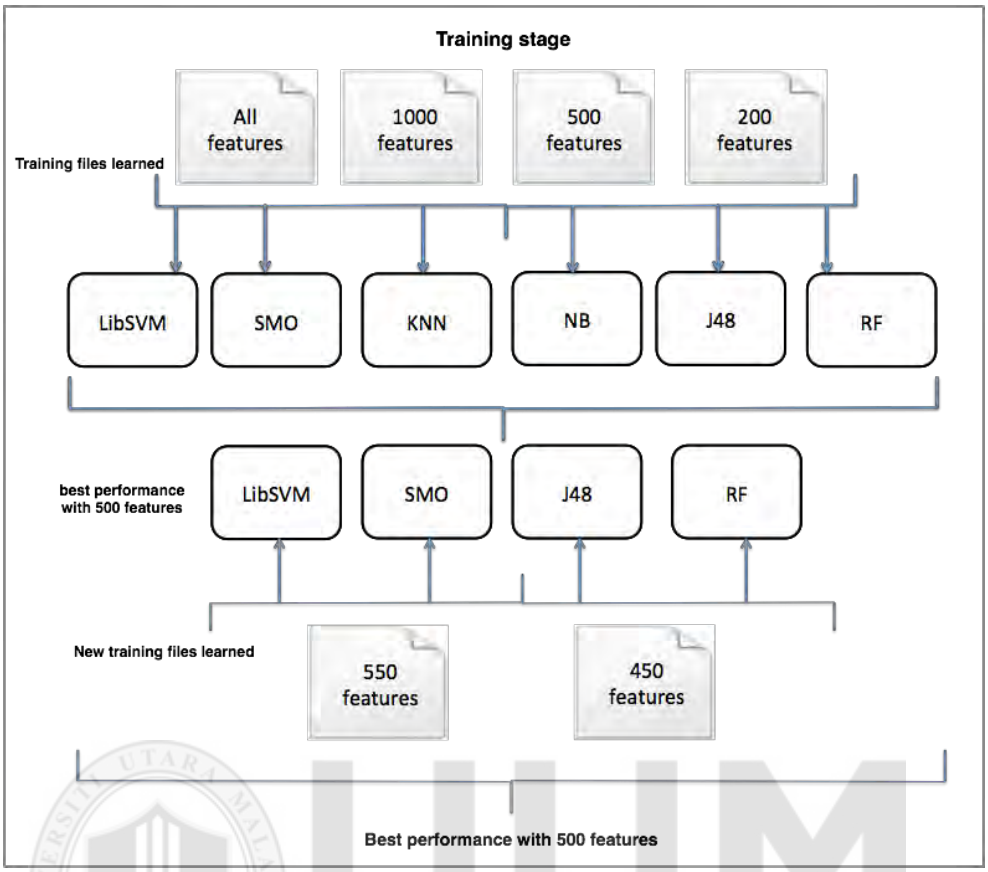


Figure 3. 11: Training Stage

3.3.5.2 Prediction stage

Based on the training stage, four classification algorithms, libSVM, RF. SMO, and J48, were used as classifiers to classify a testing dataset in the prediction stage as they showed good performance during the training stage. Testing dataset, obtained from UUM network devices, was processed and represented with 500 features in ARFF files; this 500 of features is the best number to be used as shown in the results of training stage. Classifiers results were compared with each other in terms of probability rate and it showed that LibSVM was the best model to be used for syslog data classification. Probability rate indicates to the proportion of accuracy that the classified instance relays to the specific class. Probability rate for each classified instance is shown in the table of classification results. It is calculated by algorithms of used tool, which is Weka.

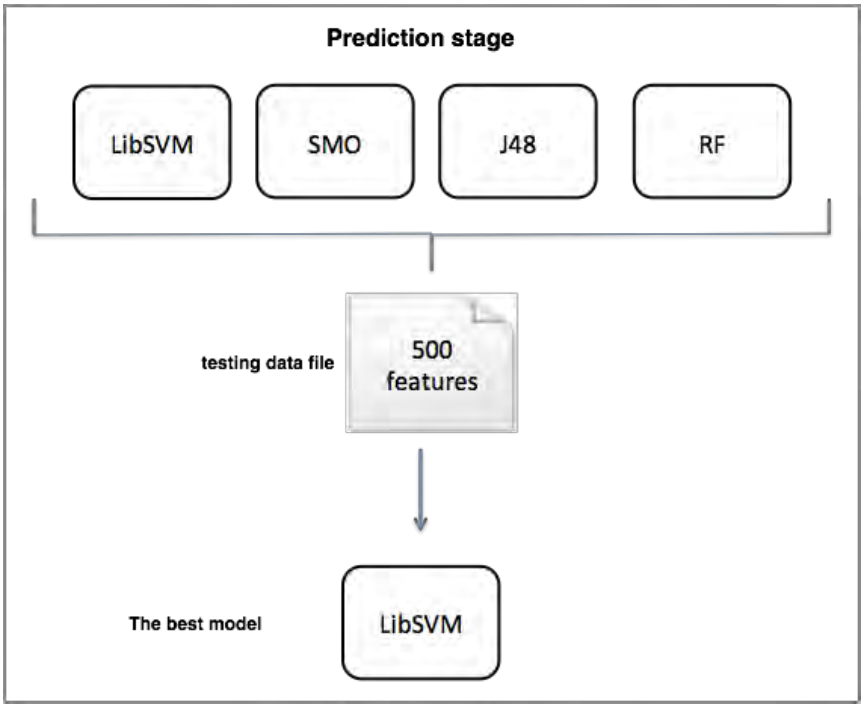


Figure 3. 12: Prediction Stage

3.4 Phase Four: Validation the Classification Method

Based on the training stage, four classification algorithms: libSVM, RF SMO, and J48 showed good performance. These four algorithms were used in prediction stage, and they were applied to the testing dataset.

In this phase, results of prediction stage were validated. Validation process was performed by comparing instances of each class to the corresponding training dataset. Procedures and results of validation phase are presented in chapter four.

3.5 Summary

This chapter presents the methodology of the study applied to the four phases: identifying network problems and their causes for each network layer, identifying syslog messages related to network problems for each layer, applying classification technique on syslog data sets, and the last one is validation of the results of classification method. The first phase clarifies the framework of the proposed

method, followed by steps of initializing training data set, and testing data set collected. This is followed by the preprocess steps in the data set to prepare it for classification process. Next is an explanation of feature extraction method, steps, and conducted experiments. This chapter then presents the explanation of building the classifier models, and the procedures of identifying the best model. It also shows the performance metrics that have been used which are: accuracy rate, and probability rate. Finally, the summary of this chapter is presented in the last section.

CHAPTER Four

RESULTS AND DISCUSSION

This chapter presents the results obtained from the classification algorithms used in the proposed method to classify syslog data. It includes five main sections organized in the following paragraphs. Section 4.1 explains results of achieving the first objective. Section 4.2 shows results of training stage, including the best performance. Section 4.3 shows results of prediction stage - classifying the testing datasets. Section 4.4 explains results of validation phase - validation results of each class. Section 4.5 presents comparison results of the algorithms . And Section 4.6 summarizes all the results.

4.1 Results of the first objective

The first objective aims to identify syslog messages that describe network problems related to each network layer. Network problems were identified for each layer and key words that described each problem were extracted to be used for searching in Cisco syslog manual. Table 3.1 explains the problems and key words for each network layer.

A total of 263 syslog message were extracted from the syslog manual. The following table shows examples of the identified syslog messages; one example for each problem.

Table 4.1: *Summary of network problems, key words, and message example.*

Network layer	Problem	Key words	Message example
Layer1	Loss of connectivity, cabling fault, high collision counts.	Disconnected cable, damaged cable, improper cable, cable fault.	101002: (Primary) Bad failover cable.
Network access layer	Network bottlenecks or congestion, hardware fault.	Fault interface, interface fail, transmission error.	105043: (Primary) Failover interface failed.
	High CPU utilization rates, attenuation.	Exceed design limit.	201009: TCP connection limit of

			number for host *IP_address* on *interface_name* exceeded.
	Address mapping error.	Fail address mapping.	737030: Unable to send *IP-address* to standby: address in use.
Layer2 Internet layer	Network failure, network performance below the baseline.	Network failure.	105032: LAN Failover interface is down.
Layer3	Address translation problems.	Address translation, translation fail.	202001: Out of address translation slots!
Transport layer	Domain name server (DNS) problems.	DNS fail.	331001: Dynamic DNS Update for '*fqdn_name*' <=> *ip_address*

		failed.
DHCP difficulty operating.	DHCP configured fail.	737004: DHCP configured, request failed for tunnel-group *'tunnel-group*.
SNMP contact problems.	SNMP unable to open.	212001: Unable to open SNMP channel (UDP port *port*) on interface *interface number*, error code = *code*
Access control list (ACL) problems.	ACL error, ACL configuration.	109020: Downloaded ACL has configuration error; ACE

Layer4 Application layer	Slow application performance.	Application fail, application stopped.	505012: Module in slot *slot*, application stopped *application*, version *version.*
	No network service available.	FTP fail, HTTP fail.	201005: FTP data connection failed for IP_address *IP_address*

4.2 Results of Training Stage

Training data set files with all features, 1000 features, 500 features and 200 features, were generated, firstly to be used for training stage to evaluate the performance of text classification algorithms. The results obtained were then compared to identify the best number of features and the best models. Training dataset which contains 263 instances (syslog message) was used in training stage.

Six classification algorithms were applied to the above training dataset files and performance rate –which is the accuracy rate- was expressed by the ratio of correctly classified instances, for each classifier. The results of six algorithms performance, using above training datasets are as follows:

Table 4.2: *Accuracy Rate of Six Classifiers Using Training Files Represented by Different Numbers of Features*.

Training File	SVM(Lib)	SVM(SMO)	Naïve Bayes	KNN	J48	Random Forest
All Features	73.80%	71.10%	41.10%	64.30%	64.30%	70.70%
1000 Features	74.10%	71.10%	52.90%	64.30%	64.30%	73.00%
500 Features	74.50%	71.10%	60.10%	64.30%	69.30%	75.70%
200 Features	65.4%	65.40%	55.50%	49.80%	58.20%	67.70%

As shown in Table 4.2, the performance values recorded from training file show that 500 features had the highest values, as the number of correctly classified instances were higher than the values of other files. This means that this file contains the best number of features to be used for prediction stage. It can be noted also that the accuracy rate of Random Forest, LibSVM, SMO, and J48 had the highest values, as

they classified instances correctly, better than other algorithms.

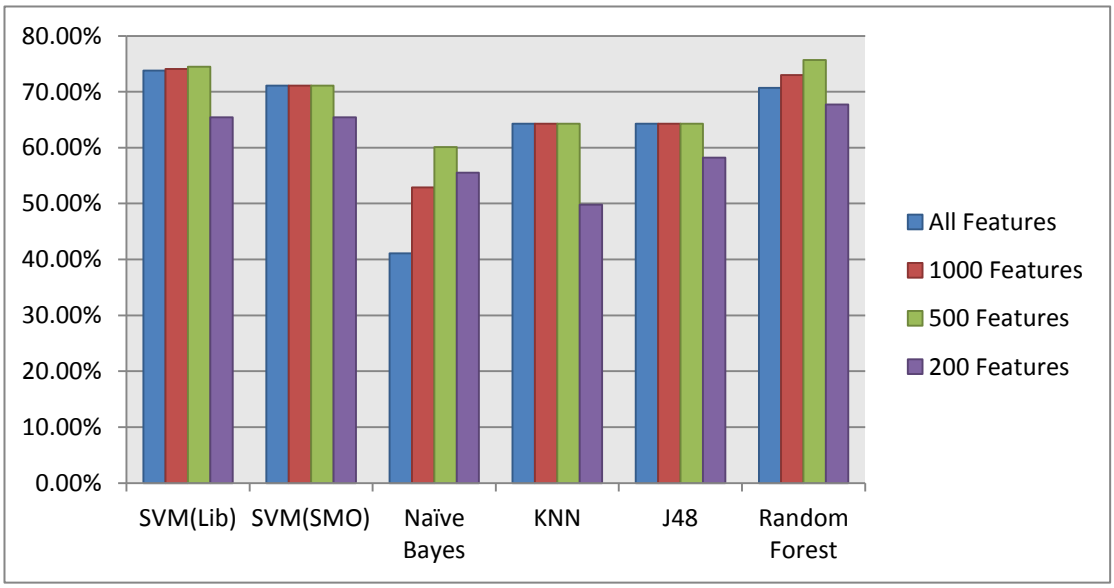


Figure 4.1: The comparison between LibSVM, SVM(SMO), NB,KNN, J48, RF algorithms in terms of accuracy rate.

Two training dataset files were generated, one with 450 features and the other with 550 features. They were used to make sure that the file of 500 features contained the best features to be used in prediction stage. Four classification algorithms: Random Forest, LibSVM, SMO, and J48 were applied to the new two training dataset, and the performance was recorded by calculating their accuracy rate. The results were compared with the results of previous training dataset with 500 features. Table 4.3 shows the accuracy rate of four classifiers using training files represented by different numbers of features.

Table 4.3: *Accuracy Rate of Four Classifiers Using Training Files Represented by Different Numbers of Features.*

Training File	SVM(Lib)	SVM(SMO)	Random Forest	J48
550 Features	71.10%	70.30%	71.10%	64.30%
500 Features	74.50%	71.10%	75.70%	64.30%
450 Features	71.10%	70.40%	71.10%	64.30%

As shown in Table 4.3, accuracy rate of algorithms show less values when using training dataset files of 550 features and 450 features. This means that training dataset file with 500 features contained the best features to be used for prediction stage.

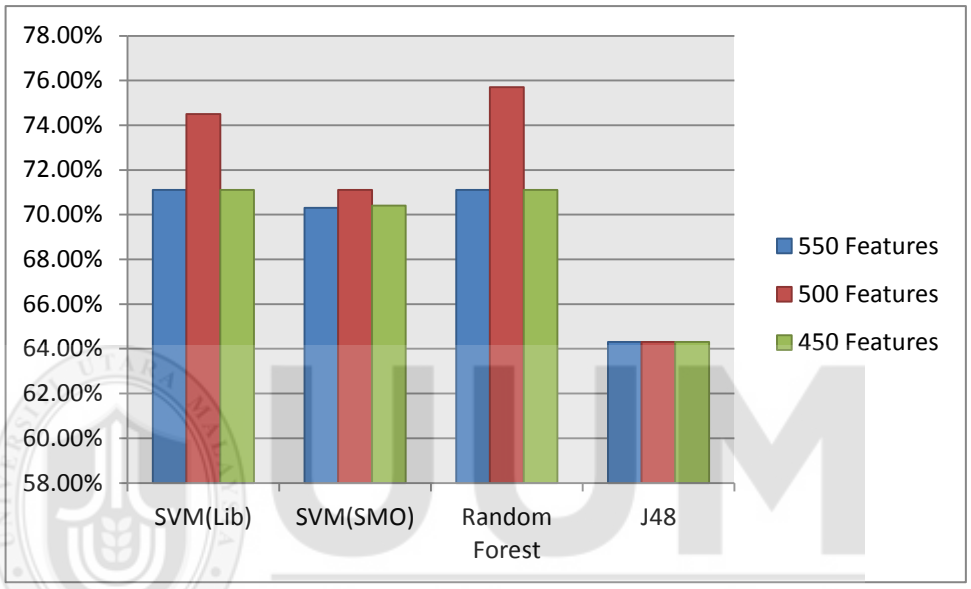


Figure 4.2: The comparison between dataset files in terms of algorithms accuracy rate

Experiments results of training stage showed that the features of number 500, had the best performance by applying Random Forest, LibSVM, SMO, and J48 classification algorithms..

4.3 Results of Prediction Stage

In the prediction stage, four classification algorithms: Random Forest, LibSVM, SMO, and J48 were applied to the testing dataset. Classification algorithms were chosen for good performance during the training stage.

A testing dataset was obtained from UUM network devices; it consists of 2610 instances (syslog message) from firewalls and switches devices involving a short

period of time (less than one minute). Testing dataset was preprocessed similar to training dataset (cleaned, removed stop words, stemmed, removed duplicated words). Testing data set file was generated with 500 features in ARFF format to be classified using Weka data mining tool. The results of the comparison between the four classification algorithms-Random Forest, LibSVM, SMO, and J48-involving testing dataset are presented in Table 4.4.

Table 4.4: *Prediction Stage Results for Four Classifiers*

Algorithm	layer1	layer2	layer3	layer4
LibSVM	172 (6.59%)	2383(91.30%)	55 (2.11%)	0 (0%)
SVM (SMO)	134 (5.13%)	2351 (90.08%)	126 (4.83%)	0 (0%)
J48	250 (9.58%)	2320 (88.89%)	40 (1.53%)	0 (0%)
RF	170 (6.51%)	2395 (91.76%)	43 (1.65%)	2 (0.08%)

Table 4.4 shows the number of instances, classified into each layer with the percentage of all testing dataset sample. Four algorithms Random Forest, LibSVM, SMO, and J48 gave convergent results; LibSVM, SMO, and J48 algorithms classified all instances into three classes (layer1, layer2, and layer3), while RF algorithm classified all instances into four classes (layer1, layer2, layer3, and layer4), layer4 got only two instances.

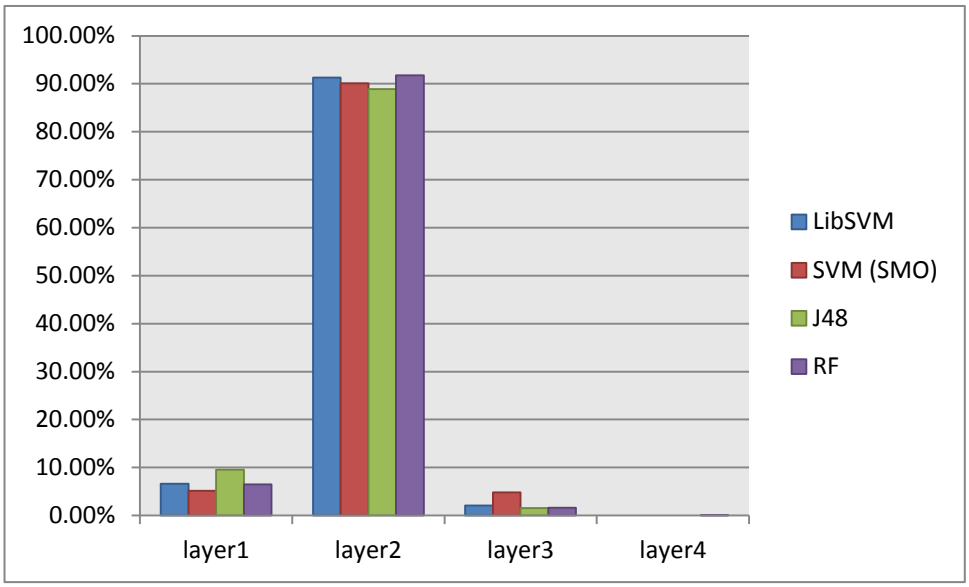


Figure 4.3: The Comparison Between Four Classifiers in Terms of Prediction Stage Results

Syslog data contains information of all network events with various types and severity levels and almost all syslog messages are either informational or problem messages. Since classification algorithms classify all instances of testing dataset into specific classes, the results need to be analyzed deeply in terms of the probability rate of each classified instance, as a way to identify informational instances and problems ones. Probability rate indicates to the proportion of accuracy that the classified instance relays to the specific class. The following table compares the range of probability rate of classified instances for results of Random Forest, LibSVM, SMO, and J48

Table 4.5: *Probability Range of Classified Instances for Used Classifiers.*

Algorithm	layer1	layer2	layer3	layer4
LibSVM	(72.20 - 32.80)%	(67.00 – 33.00)%	(89.90 - 36.20)%	0.00%
SVM (SMO)	(50.00 - 40.30)%	(55.00 - 40.3)%	(50.00 - 40.3)%	0.00%
J48	(60.50 - 45.30)%	(50.00 - 40.00)%	(50.30 - 40.30)	0.00%
RF	(79.00 - 26.00)%	(78.00 - 34.00)%	(58.00 - 36.00)%	38.00%

As shown in Table 4.5, the results of LibSVM algorithm show probability rates higher than others classifiers. LibSVM gave higher values for both maximum and minimum rate in each layer range, except for the first layer range whose maximum value is less than RF.

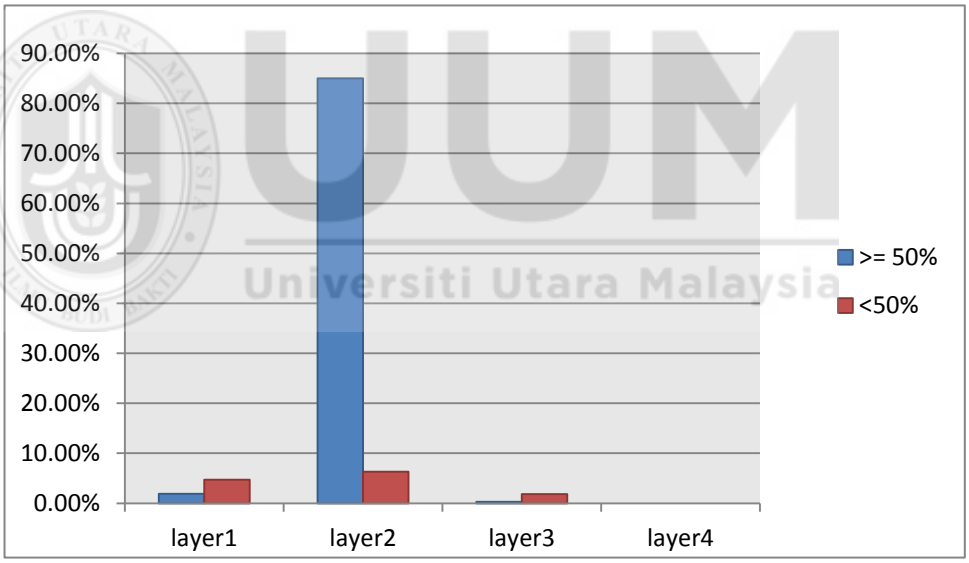
Depending on the previous results, the results of LibSVM algorithm is considered as the best. The result is divided into two parts: one for the lower probability to be validated as an informational messages, and the other for the higher probability to be

validated as problems messages. Each part is compared to the training dataset to validate the result. The following table shows the numbers of instances and their percentage with probability rate >= 50% and < 50% for each class.

Table 4.6: *Probability rate for classified instances using LibSVM*

probability	layer1	layer2	layer3	layer4
>= 50%	49 (1.88%)	2218 (84.98%)	8 (0.31%)	0.00%
<50%	123 (4.71%)	165 (6.32%)	47 (1.80%)	0.00%

As shown in Table 4.5, layer1, and layer3 got small numbers of instances with probability rate >= 50% , but layer2 had large numbers of instances with probability



rate >=50%; this is because of repeated messages for the same problem with one different word.

Figure 4.4: Classified Instances in Terms of Probability Rates for Each Class

4.4 Results of Validation Phase

In the validation phase, results of prediction stage were analyzed to make sure that each instance belongs to its class, and it refers to a problem in one layer. Validation process was performed by comparing instances of each class to the corresponding training dataset.

LibSVM classifier classified all instances of testing dataset into three layers- (layer1, layer2, layer3) - and as mentioned before, the classified instances for each layer were divided into two parts in terms of prediction probability. Instances with probability >= 50% were validated by comparing them to the corresponding training dataset. Instances with probability < 50% were validated as informational messages.

4.4.1 Layer1 Validation

The classifier had classified 49 instances with probability >= 50% , to layer1. These instances were compared to the instances belonging to class one in training dataset. Only three instances indicated network problem and the probability of them was > 70%. These three messages described the problem of " TCP connection to firewall server had been lost, restricted tunnels are now allowed full network access". Repeated three times.

By referring to syslog manual, this problem indicated that the TCP connection to the security appliance server was lost and this requires checking the server and network connections. This problem belongs to the first layer, which is network access layer, as mentioned in chapter two.

4.4.2 Layer2 Validation

The classifier had classified 2218 instances with probability >= 50% , to layer2. These instances were compared to the instances belonging to class two in training

dataset. From all classified instances, no instances had indicated any network problem. The probability of them was < 70%.

4.4.3 Layer3 Validation

The classifier had classified eight instances with probability >= 50%, to layer3. These instances were compared to the instances belonging to class three in training dataset. From all classified instances, one instance had indicated network problem. The probability of it was > 70%, this message described the problem of "No translation group found for protocol src".

By referring to the syslog manual, this problem was due to network address translation (NAT) not configured for the specified source and destination systems. This problem pointed to NAT issues that belong to the third layer, which is transport layer, as mentioned in chapter two.

4.4.4 Validation Of Instances With Low Probability

Instances that had probability validation less than 50% were compared to training dataset. There was no instance pointing to network problems; it described network events only. Result of validation process seems to be acceptable, as testing data sample was for short period of time, less than one minute.

4.5 Comparison of the Used Algorithms

This study used six classification algorithms to classify syslog data and the results of all algorithms were recorded in tables to be compared. Comparison results indicated that SVM had the best performance during the experiments of this study, as shown in Tables 4.2, 4.3, and 4.5.

This study used two methods for multi classification SVM, one-against-all, and one-against-one. The results of their experiments were compared, and the comparison

results indicated that one-against-all method showed better performance during the experiments of this study, as shown in Tables 4.2, 4.3, and 4.5.

4.6 Summary

The results indicate that Random Forest, LibSVM, SMO, and J48 algorithms showed good performance during training stage, and gave 75.70%, 74.50%, 71.10%, and 69.30% of correctly classified instances, respectively. LibSVM algorithm classified instances with probability rate higher than RF, SMO, and J48 algorithms. Probability rate of classified instances using LibSVM, was in the range of 89.90% - 32.80%. Validation results showed that probability rate of correctly classified instances was >70%. Comparison results of the used algorithms indicated that SVM algorithm got the best performance, and one-against-all method got better performance than one-against-one method during experiments of this study.

CHAPTER Five

CONCLUSION AND FUTURE WORK

This chapter begins with a summary of the thesis, presented in Section 5.1 Section 5.2. explains the contribution of the study. Section 5.3., limitations of the study and finally, Section 5.4 offers some suggestions and future directions.

5.1 Summary

Network troubleshooting is one of the main aspects of network management, and the first step in the process is to detect the problems. This thesis proposed a method to detect and classify network problems, in terms of network layers, by analyzing syslog data, and to achieve this goal, the thesis adopted a specific path. Firstly, by reviewing the relevant literature, network layers and identifying related problems. Secondly, syslog messages that represent network problems for each layer were identified, depending on the identified problems and their causes. These messages were extracted from Cisco syslog manual. Thirdly, the extracted messages were processed and used to learn the classifiers to categorize syslog data into four classes, which are TCP/IP layers. Six different classification algorithms were used in training phase, and the results were compared to identify the best classification model. Testing datasets were used to apply the method to the data obtained from UUM network devices. The results showed that LibSVM algorithm had the best performance. Probability rate of instances that were classified correctly was more than 70%.

5.2 Contribution of Study

Classifying syslog data in terms of TCP/IP layers is the main contribution of this

study, as it had succeeded in enhancing the processes of network troubleshooting, and impacted efficient maintenance decision. This study has also reinforced results of previous works, which compared text classification algorithms, and confirmed the results which considered SVM to be the best classification algorithm in the field of text data classification. This study made a comparison between one-against-one method, and one-against-all method, for multi classification (SVM), and the results indicated that one-against-all method performed better than one-against-one in this study.

5.3 Limitations

The study aims to classify syslog messages -that indicate to network problems- from syslog data into network layers. Syslog data contains a large number of messages. Identified syslog messages –that indicate to a problem- are related to the specified problems in this study only. The proposed method would be efficient if all problem messages have extracted, and used for training the model. This takes a long time, and a considerable effort.

The proposed method was applied to a small data, comparing to the real volume of syslog data, this couldn't show the real performance of the study.

5.4 Future Works

Recommendations for future research are outlined below:

i. In this study, syslog messages used in training stage were related to specific network problems. All syslog messages that indicated network problems were extracted and used as training dataset.
ii. This study used data mining tools for analysis and classification purposes. Big data applications would need to be applied for the same purposes, since syslog data is categorized as big data.
iii. This study used Cisco syslog format to extract problems messages for

training dataset. Other formats could be used for the same purpose and results could be compared.

REFERENCES

[1] J. D. Sloan, *"network management and troubleshooting" in Network Troubleshooting Tools*, 1st ed. USA: O'Reilly, 2001.

[2] O. Kyas, *Network Troubleshooting*. California: Agilent Technologies, 2001.

[3] R. Hudyma and D. I. Fels, "Causes of Failure in IT Telecomunications Networks," in *Proceedings of SCI*, 2004, pp. 35–38.

[4] P. C. Gupta, *Data Communications And Computer Networks*, Eastern ec. New Delhi: Prentice hall of india private limited, 2006.

[5] B. Vachon and R. Graziani, *Accessing the WAN CCNA Exploration Companion Guide*, 1st ed. USA: Cisco Press, 2008.

[6] A. Deveriya, *Network Administrators Survival Guide.*, 1st ed. USA: Cisco Press, 2005.

[7] J. Manyika, M. Chui, B. Brown, J. Bughin, and R. Dobbs, *Big data: The next frontier for innovation, competition, and productivity*, 1st ed. McKinsey Global Institute, 2011.

[8] Q. He and B. Veldkamp, *Classifying unstructured textual data using the Product Score Model: an alternative text mining algorithm*, 1st ed. Enschede, Netherlands: RCEC, Cito/University of Twente, 2012.

[9] T. Qiu, Z. Ge, D. Pei, J. Wang, and J. Xu, "What happened in my network: mining network events from router syslogs," in *Proceedings of the 10th ACM*, 2010, pp. 472–484.

[10] M. Roy, "Empirical Study of Different Classifiers for Sentiment Analysis," in *Data Mining and Knowledge Engineering 6.4*, 2014, pp. 160–164.

[11] T. Kimura, K. Takeshita, T. Toyono, M. Yokota, K. Nishimatsu, and T. Mori, "Network failure detection and diagnosis by analyzing Syslog and SNS data: Applying big data analysis to network operations," *NTT Tech. Rev.*, vol. 11, no. 11, 2013.

[12] K. Fukuda, "On the use of weighted syslog time series for anomaly detection," in *12th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2011, pp. 393–398.

[13] A. Fox, D. Patterson, and M. I. Jordan, "Invited Applications Paper Detecting Large-Scale System Problems by Mining Console Logs," in *27th International Confer- ence on Machine Learning*, 2010.

[14] S. Hekmat, *Communication networks*. Línea] http://www. pragsoft. com/books/CommNetwork. pdf, 2005.

[15] B. A. Forouzan and S. C. Fegan, *Data communications and networking*, 4th ed. NewYork: The McGraw-Hill Companies, Inc, 2007.

[16] P. Simoneau, *The OSI Model: understanding the seven layers of computer networks*. www.globalknowledge.com: Global Knowledge Training LLC, 2006.

[17] S. R. Wilkins, *Designing for Cisco internetwork solutions (Desgn) foundation learning guide*, 3rd ed. Indianapolis: Cisco Press, 2012.

[18] V. Karman, "Understanding downtime, A Vision solutions white paper," California, 2006.

[19] S. Pertet and P. Narasimhan, "Causes of failure in web applications," in *Research Showcase in CMU*, 2005, p. 48.

[20] K. Takeshita, M. Yokota, and K. Nishimatsu, "Early network failure detection system by analyzing twitter data," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 279–286.

[21] M. a. Mohamed, O. G. Altrafi, and M. O. Ismail, "Relational vs. NoSQL databases: A survey," *Int. J. Comput. Inf. Technol. (IJCIT)*, vol. 03, no. 03, pp. 598–601, 2014.

[22] S. Pospiech, S. Mielke, R. Mertens, K. Jagannath, and M. Stadler, "Exploration and analysis of undocumented processes using heterogeneous and unstructured musiness data," in *IEEE International Conference on Semantic Computing Exploration*, 2014, pp. 191–198.

[23] I. Neeman and B. H. Lovering, "Executing structured queries on text records of unstructured data," in *U.S. Patent No. 20,150,149,496*, 2015.

[24] S. Reissmann, D. Frisch, C. Pape, and S. Rieger, "Correlation and consolidation of distributed logging data in enterprise clouds," *Int. J. Adv. Internet Technol.*, vol. 7, no. 1, pp. 39 – 51, 2014.

[25] S. Geetha and G. Anandha Mala, "Effectual extraction of data relations from unstructured data," in *Th ird International Conference on Sustainable Energy and Intelligent System,VCTW, Tiruchengode, Tamilnadu, India*, 2012.

[26] A. Bacchelli, N. Bettenburg, and L. Guerrouj, "Workshop on mining unstructured data (MUD) ... Because 'Mining unstructured data is Like fishing in muddy waters'!," in *19th Working Conference on Reverse Engineering. IEEE*, 2012, pp. 5–6.

[27] F. S. Gharehchopogh, "Approach and review of user oriented interactive data mining," in *4th International Conference on Application of Information and Communication Technologies. IEEE*, 2010, pp. 1–4.

[28] F. S. Gharehchopogh, "Approach and developing data mining method for spatial applications," in *International Conference on Intelligent Sysyems and Data Processing (ICISD)*, 2011, pp. 342–344.

[29] L. Huo, Y. Fang, and H. Hu, "Dynamic service replica on distributed data mining grid," in *International Conference on Computer Science and Software Engineering*, 2008, pp. 390–393.

[30] F. S. Gharehchopogh and Z. A. Khalifelu, "Analysis and evaluation of unstructured data: text mining versus natural language processing," in *5th International Conference on Application of Information and Communication Technologies (AICT)*, 2011, pp. 1–4.

[31] V. Gupta and G. S. Lehal, "A survey of text mining techniques and applications," *J. Emerg. Technol. Web Intell.*, vol. 1, no. 1, pp. 60–76, 2009.

[32] H. H. Malik and V. S. Bhardwaj, "Automatic training data cleaning for text classification," in *11th IEEE International Conference on Data Mining Workshops*, 2011, pp. 442–449.

[33] K. Nithya, P. C. D. Kalaivaani, and R. Thangarajan, "An enhanced data mining model for text classification," in *nternational Conference on Computing, Communication and Applications (ICCCA)*, 2012, pp. 1–4.

[34] J.-C. Lamirel and P. Cuxac, "Improving textual data classification and discrimination using an ad-hoc metric: Application to a famous text discrimination challenge," in *4th IEEE International Symposium Concepts and Tools for knowledge Management (ISKO-Maghreb)*, 2014.

[35] B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for text-documents classification," *J. Adv. Inf. Technol.*, vol. 1, no. 1, pp. 4–20, 2010.

[36] L. Dan, L. Lihua, and Z. Zhaoxin, "Research of text categorization on WEKA," in *3rd IEEE International Conference on Intelligent System Design and Engineering Applications (ISDEA)*, 2013, pp. 1129–1131.

[37] G. Wei, X. Gao, and S. Wu, "study of text classification methods for data sets with huge features," in *2nd international conference on inustrial and information systems*, 2010, pp.

433–436.

[38] S. L. Bang, J. D. Yang, and H. J. Yang, “Hierarchical document categorization with k-NN and concept-based thesauri,” in *11th International Conference of String Processing and Information Retrieval (SPIRE). Padova. Italy*, 2006, pp. 387–406.

[39] J. W. Kim, B. H. Lee, M. J. Shaw, H. L. Chang, and M. Nelson, “Application of decision-tree induction techniques to personalized advertisements on internet storefronts.,” *Int. J. Electron. Commer.*, vol. 5, no. 3, pp. 45–62, 2001.

[40] M. R. Murty, J. V. R. Murthy, P. Reddy, and S. C. . Satapathy, “A Survey of cross-domain text categorization techniques,” in *1st IEEE International Conference on Recent Advances in Information Technology (RAIT)*, 2012.

[41] J. He, A.-H. Tan, and C.-L. Tan, “A comparative study on chinese text categorization methods,” 2000.

[42] T.-Y. Wang and H.-M. Chiang, “One-against-one fuzzy support vector machine text categorization classifier,” in *IEEE IEEM*, 2008, pp. 1519–1523.

[43] Y. Liu and Y. F. Zheng, “One-against-all multi-class SVM classification using reliability measures,” in *International Joint Conference on Neural Networks*, 2005, pp. 849–854.

[44] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines.,” *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–25, 2002.

[45] P. Pawar and S. H. Gawande, “A Comparative study on different types of approaches to text classification,” in *3rd International Conference on Machine Learning and Computing (ICMLC)*, 2011, pp. 423–426.

[46] P. Cunningham and S. J. Delany, “K -Nearest Neighbour classifiers,” 2007.

[47] Y. Liao and V. R. Vemuri, “Use of k-nearest neighbor classifier for intrusion detection,” *Comput. Secur.*, vol. 21, no. 5, pp. 439–448, 2002.

[48] S. Oleiwi, “Enhanced ntology-based text classification algorithm for structurally organized documents suha sahib oleiwi doctor of philosophy permission to use,” 2015.

[49] T. G. Dietterich, “Ensemble methods in machine learning.,” 2000.

[50] G. Kaur and A. Chhabra, “Improved J48 classification algorithm for the prediction of diabetes,” *Int. J. Comput. Appl.*, vol. 98, no. 22, pp. 13–17, 2014.

[51] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 5, pp. 1–35, 1999.

[52] D. . Liparas, Y. . HaCohen-Kerner, A. . Moumtzidou, S. . Vrochidis, and I. . Kompatsiaris, “News articles classification using random forests and weighted multimodal features,” in *In Multidisciplinary Information Retrieval, Springer International Publishing*, 2014, pp. 63–75.

[53] charu c. Aggarwal and cheng xiang Zhai, *Mining text data. Chapter six of the book XII, 524*. 2012.

[54] N. K. Korada, N. S. P. Kumar, and Y. V. N. H. Deekshitulu, “Implementation of Naive Bayesian Classifier and Ada-Boost Algorithm using Maize ExpertSystem,” *Int. J. Inf. Sci. Tech.*, vol. 2, no. 3, pp. 63–75, 2012.

[55] E. Frank and R. R. Bouckaert, “Naive bayes for text classification with unbalanced classes,” in *PKDD’06 Proceedings of the 10th European conference on Principle and Practice of Knowledge Discovery in Databases*, 2006, pp. 503–510.

[56] P. Achananuparp, Xiaohua Zhou, Xiaohua Hu, and Xiaodan Zhang, “Semantic representation in text classification using topic signature mapping,” in *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1034–

1040.

[57] B. Harish, D. Guru, and S. Manjunath, "Representation and classification of text documents: A brief review," *IJCA, Spec. Issue Recent Trends Image Process. Pattern Recognit.*, no. 2, pp. 110–119, 2010.

[58] S. V. Gaikwad, A. Chaugule, and P. Patil, "Text mining methods and techniques," *Int. J. Comput. Appl.*, vol. 85, no. 17, pp. 42–45, 2014.

[59] K. Celik and T. Gungor, "A comprehensive analysis of using semantic information in text categorization," in *Paper presented in IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2013, pp. 1–5.

[60] M. Radovanović and M. Ivanovi, "Text mining: Approaches and applications," *Novi Sad J. Math*, vol. 38, no. 3, pp. 227–234, 2008.

[61] Y. Chen, "New Feature Selection Methods Based on Context Similarity for Text Categorization," in *11th International Conference on Fuzzy Systems and Knowledge Discovery New*, 2014, pp. 598–604.

[62] Y. Xu and L. Chen, "Term-frequency based feature selection methods for text categorization," in *4th IEEE International Conference on Genetic and Evolutionary Computing*, 2010, pp. 280–283.

[63] F. Yigit and omer kaan Bayakan, "A new feature selection method for text categorization based on information gain and particle swarm optimization," in *3rd IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 2014, pp. 523–529.

[64] A. Shadvar and A. Erfanian, "Mutual information-based fisher discriminant analysis for feature extraction and recognition with applications to medical diagnosis.," in *32nd Annual International Conference of the IEEE EMBS*, 2010, pp. 5811–5814.

[65] S. Arani and S. Mozaffari, "Genetic-based feature selection for spam detection," in *21st IEEE Iranian Conference on Electrical Engineering (ICEE)*, 2013.

[66] H. Chen, "Partially supervised learning for radical opinion identification in hate group Web forums," in *IEEE International Conference of Intelligence and Security Informatics (ISI)*, 2012, pp. 96–101.

[67] M. Maleki, "Utilizing category relevancy factor for Ttxt categorization," in *IEEE 2nd International Conference on Software Engineering and Data Mining (SEDM)*, 2010, pp. 334–339.

[68] X. Gang and X. Jiancang, "Performance analysis of chinese webpage categorizing algorithm Based on support vector machines (SVM)," in *IEEE Fifth International Conference on Information Assurance and Security Performance*, 2009, pp. 231–235.

[69] F. Xia, T. Jicun, and L. Zhihui, "A text categorization method based on local document frequency," in *IEEE Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, 2009, pp. 468–471.

[70] N. Yusof and C. J. Hui, "Determination of Bloom ' s Cognitive Level of Question Items using Artificial Neural Network," in *10th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2010, pp. 866–870.

[71] F. Harrag, E. El-qawasmah, A. M. S. Al-salman, and S. Arabia, "Comparing dimension reduction techniques for Arabic text classification using BPNN algorithm," in *IEEE First International Conference on Integrated Intelligent Computing Comparing*, 2010, pp. 6–11.

[72] K. Shiomoto, "Technologies for traffic and network management data applications of big data analytics technologies for traffic and network management data gaining useful insights from big data of traffic and network management," *NTT Tech. Rev.*, vol. 11, no. 11, pp. 1–6, 2013.

[73] K. Yamanishi and Y. Maruyama, "Dynamic syslog mining for network failure monitoring," in *11th ACM SIGKDD International Conference on Knowledge Discovery in dData Mining*, 2005, pp. 499–508.

[74] C. Lim, N. Singh, and S. Yajnik, "A log mining approach to failure analysis of enterprise telephony systems," in *International Conference on Dependable Systems & Networks: Anchorage, Alaska*, 2008, pp. 398–403.

[75] E. Martinez, E. Fallon, S. Fallon, and M. Wang, "ADAMANT - an anomaly detection algorithm for mAintenance and network troubleshooting," in *1st IFIP/IEEE IM Workshop on Cognitive Network & Service Management (CogMan)*, 2015, pp. 1292–1297.

[76] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," *22nd ACM SIGOPS Symp. Oper. Syst. Princ. SOSP*, vol. 10, no. 7, p. 117, 2009.

[77] M. Liu and J. Yang, "An improvement of TFIDF weighting in text categorization," in *International Conference on Computer Technology and Science (ICCTS)*, 2012, vol. 47, pp. 44–47.

[78] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale bayesian logistic regression for text categorization," *Am. Stat. Assoc. Am. Soc. Qual.*, vol. 49, no. 3, pp. 291–304, 2007.

[79] V. Maurya, P. Pandey, and L. S. Maurya, "Effective information retrieval system," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 4, pp. 787–792, 2013.

[80] F. Ag, S. Rakshit, and C. V. R. Nagar, "Feature selection using bag-Of-visual-words representation," in *2nd IEEE International Advance Computing Conference (IACC)*, 2010, pp. 151–156.

[81] R. R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald, and D. Scuse, *WEKA Manual for Version 3-6-13*. University of Waikato, Hamilton, New Zealand, 2015.