A Thesis

On

Information Extraction Using Named Entity Recognition from Log Messages

For Partial Fulfillment of the Requirements for the Degree of Masters of Computer Information System Awarded by

Pokhara University

Submitted by

Prabhat Pokharel

16818

Supervised by

Dr. Basanta Joshi

Department of Graduate Studies
Nepal College of Information Technology
Balkumari, Lalitpur

29 July, 2018

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to Prof. Dr. Shashidhar Ram Joshi and Dr. Arun Timilsina for their valuable feedbacks. I would like to thank Dr. Basanta Joshi for supervising my thesis work. I would also like to thank Mr. Saroj Shakya, Program Coordinator of Master's Degree, for his constant focus on research activity, motivation, and guidance. I would like to thank Mr. Niranjan Khakurel, Principal NCIT, for guidance and encouragement in research works during the master's program. Finally, I would like to thanks my colleagues Roshan Pokhrel and Sandeep Sigdel for their help in the course of the thesis.

ABSTRACT

Extracting correct and useful information from log messages is useful for real-time analysis and detecting faults, anomalies and security threats. The semantics of the extracted information is needed for deeper analysis. Very little work has been done in the past for automated information extraction from log messages. Thus, in this research work, I proposed a model to extract information from the log messages. The approach was used to extract named entities by building a classifier model. The classifier model was trained using Security Event Logs from Windows OS and Exchange Mail Server Log Messages. And the results were compared with the existing frequent item-set approach. In comparison, it was discovered that the proposed approach outperformed the existing approach as the numbers of categories were increased.

TABLE OF CONTENTS

1 INTRODUCTION 9

1.1 PROBLEM DEFINITION 10

1.2 MOTIVATION 10

1.3 OBJECTIVE 12

1.4 SCOPE 12

2 RELATED LITERATURE 13

2.1 BACKGROUND THEORY 13

2.2 NATURAL LANGUAGE PROCESSING 13

2.3 FREQUENT ITEM-SET MINING 14

2.4 CLUSTERING 14

2.5 SIEM 14

2.5.1 NAMED ENTITY RECOGNIZATION 14

2.5.2 PART OF SPEECH TAGGING 15

2.5.3 SUPPORT VECTOR CLASSIFIER 15

2.5.4 NAÏVE BAYES CLASSIFIER 15

2.6 LITERATURE REVIEW 16

3 METHODOLOGY 18

3.1 PROPOSED MODEL 18

3.2 DATA COLLECTION 20

3.3 STEPS IN FEATURE GENERATION 21

3.4 VALIDATION 23

4 RESULTS AND DISCUSSION 23

4.1 EXPIREMENTAL SETUP 23

4.1.1 TRAINING PHASE 23

4.2 RESULTS AND OBSERVATIONS 24

4.2.1 TESTING PHASE 24
4.2.2 COMPARISON BETWEEN FREQUENT ITEM-SET AND PROPOSED APPROACH 25
4.2.3 COMPARING ACCURACY WITH COUNT OF LOGS 26
4.2.4 COMPUTATIONAL TIME 26
5 REFERENCES 28

LIST OF FIGURES

Figure 1 Log Collection without data lake (left) and with data lake (right) 10

Figure 2 Regular log pattern 11

Figure 3 Log pattern with exception 12

Figure 4 Support vector classifier 15

Figure 5 Proposed approach 18

Figure 6 Break down of message part 19

Figure 7 Category 1: Break down of message part 19

Figure 8 Category 2: Break down of message part 20

Figure 9 Category 3: Break down of message part 20

Figure 10 Block diagram for feature generation 23

Figure 11 Accuracy comparison between frequent item-set and proposed approach 26

Figure 12 Variation of accuracy with count of logs 26

LIST OF TABLES

TABLE I FEATURE CONTRIBUTION 19

TABLE II TAGGING OF TOKENS WITH LABELS 20

TABLE III TYPE 1 LOG SAMPLE..... 24

TABLE IV TYPE 2 LOG SAMPLE 24

TABLE V TYPE 3 LOG SAMPLE..... 24

TABLE VI PERFORMANCE MATRIX 25

TABLE VII SAMPLE OF TEST RESULTS ON SYNTETIC DATA 25

LIST OF ABBREVATIONS

NLP	Natural Language Processing
NER	Named Entity Recognition
AUC	Area Under Curve
SIEM	Secure Information and Event Management
SVM	Support Vector Machine
HMM	Hidden Markov Model
CRF	Conditional Random Fields
OCSVM	One Class Support Vector Machine
CFDR	Computer Failure Data Repository
DC	Domain Controller
POS	Part of Speech
OS	Operating System
ROC	Receiver Operating Characteristics
IDS	Intrusion Detection System
IPS	Intrusion Protection System
NIDS	Network Intrusion Detection System
IoT	Internet Of Things
SLCT	Simple Log Clustering Tool
KB	Knowledge Base

1 INTRODUCTION

A log [1] message is a computer-generated string with a significant amount of contextual information. This information is passed to the logging unit through direct calls and also obtained from the operating system as a part of the operating system. Log messages can be of various types, among them Syslog and CEF are some of the most popular ones. The Syslog messages have the following components:

Severity

Severity defines the criticality of an event. Severity has values ranging from 0 to 7, 0 being the most severe while 7 being the least. The severity values can change based on an application and its usage. Following is the list of severities and their values:

- 0 Emergency
- 1 Alert
- 2 Critical
- 3 Error
- 4 Warning
- 5 Notice
- 6 Informational
- 7 Debug
- 8 Facility

The facility defines the type of program logging the message. Like facility 0 defines kernel messages and 2 define the user-level message. In total there are 24 facility values.

A regular Syslog message can be divided into two sections. The Syslog header and the Syslog message part. The header portion contains values such as severity, facility, timestamp, host or IP addresses of the logging server. While the message portion contains the actual log event. The message portion can contain information such as user names, computer names, service names, host or client names, IP addresses, data usage, response time, port numbers, protocols, object, action, etc.

Log management uses logs from the end sensors and deals with issues on security, operations and regulatory compliance. Challenges with log management, particularly when it comes with big data and IoT, can be as follows:

Volume

A high volume of data requires a lot of human and machine power for computation and analysis. With the rise of concepts such as big data and IOT, the volume of data generated by the end sensor has been growing rapidly. This is thus a big challenge when it comes to log management.

Velocity

Velocity means the rate at which the data is being collected. With the usage of IOT and ipv6, log generation is very high which at the collection layer brings about a lot of velocity and this can be challenging in log management.

Variation

As different endpoints generate different kinds of logs. These logs can greatly vary when it comes to data semantics and data structure. This can also be a big challenge for log management.

Veracity

Not all the information contained in the log messages may be correct. This can be the case in IDS, IPS, NIDS, etc. Incorrect information can also be a big challenge in log management.

Information extraction is the process of extracting clear and fact-based information from a data source. Named Entity Recognition is an approach in NLP or Natural Language Processing by which the names in a given string are extracted and mapped to a set of entities. It can be described as a process of finding and classifying names in a given text.

1.1 PROBLEM DEFINITION

Security requirements have been high with the increased usage of computer networks and related application. The number of applications, sensors and end point devices to monitor the status of the infrastructure has been growing every day. These devices generate a huge amount of log messages every day. The log messages contain contextual information related to action that triggered that log event. This information can be both human readable and machine-readable. The volume and variation of these data types adds to the complexity how we understand and perceive these log messages. So, to solve this problem, we need to identify an approach to extract the information in the form of named entities. Regular expression has been used for the purpose of parsing the log messages and extracting the needed information. Not much work has been done to perform this using Machine Learning approach and particularly NLP.

1.2 MOTIVATION

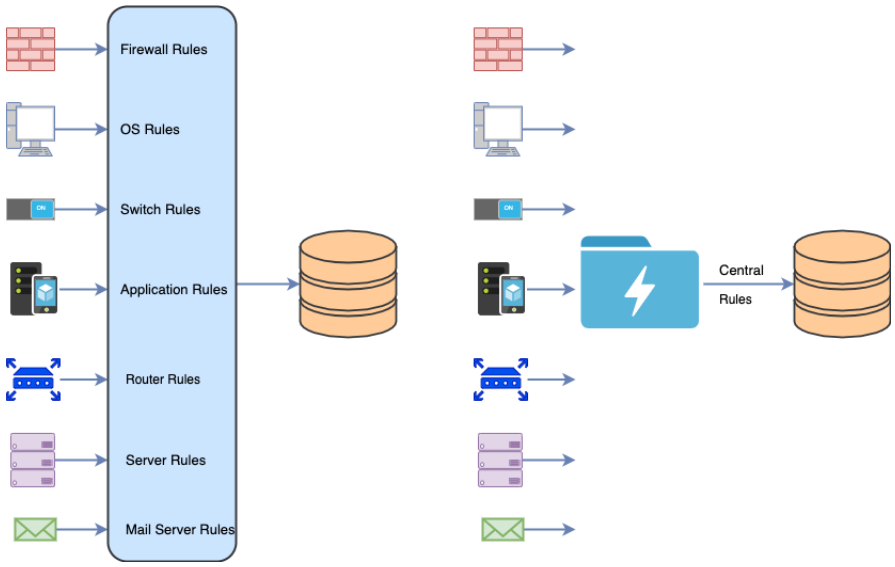

Figure 1 Log Collection without data lake (left) and with data lake (right)

Log management systems collect log data from many different sources and each of these sources contain multiple log categories. A log category contains similar log messages which can directly be mapped to a log signature. A log category is composed of a defined language pattern. However, this language pattern may not resemble the natural language. For example, "User <user> authentication successful" and "Authentication successful for <user>". These examples show that position of desired named entities may vary. Similar kinds of rules are observed in other log messages. Thus, these rules can be used to identify the constant parts and the variable parts in the log messages. A signature will represent the constants as in the original strings and variables by a fixed pattern. Here, Figure. 2 shows a regular log pattern. Most of the log signature generation techniques are based on the concept that within a pattern the varying strings are the entities and the remaining strings are the constants. However, this may not always be the case. There are some limitations to the assumptions made by most of the existing approaches. These are:

Variable entities do not change within a given dataset. In the example in Figure. 3 we see that failed and successful are represented as variables though they are actually the constants. In this case, as the user has same value it is interpreted as a constant. If the data set contains some rare log samples, with one log message per category, the constant tokens between different categories may be interpreted as variables.

In approaches such as parsing tree, it is assumed that log messages with different lengths can fall into different categories and thus generate different signatures. However, this cannot always be correct. The absence of a variable entity and change in number tokens used for a variable entity is widely observed behavior.

It is also observed that certain sequences in the log message are considered more significant in determining a category which again is not correct as most of the log message can have the same or similar header information. In the case of parsing tree sequence at the beginning of the log message is considered for identifying a category. This can cause the algorithm to fail in many cases. To solve these limitations, we adapt to use log clustering followed by NER. Clustering is used to identify log patterns. While NER is used to identify the constant terms and variable terms.

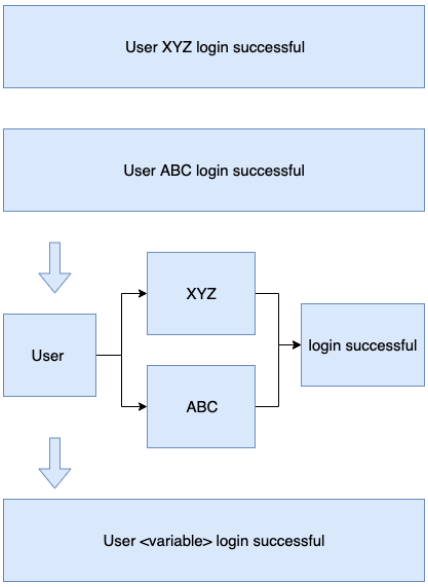

Figure 2 Regular log pattern

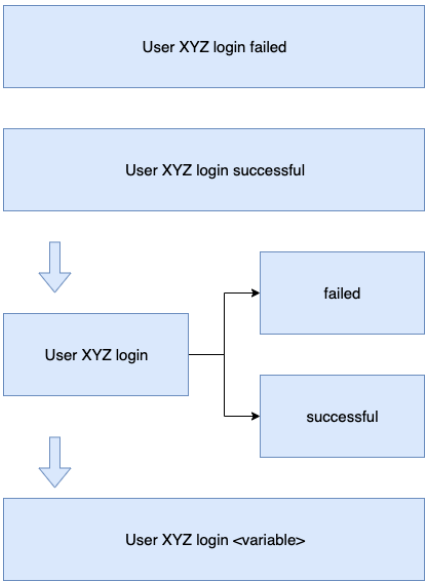

Figure 3 Log pattern with exception

1.3 OBJECTIVE

The objective of this thesis work is to verify that the language constructs in log messages can be used to build classifier models and thus used to extract information from log messages in the form of named entities. So, here we built classifier models to extract information from logs messages from Windows OS and Exchange Mail Server in the form of key-value pairs. This would further pave the way to automatically generating signatures in log management solutions. Furthermore, objective is to focus on extracting all the name type entities such as user, computer, host etc. Finally, we also require the outcome of the experiment to be compared with one of the existing and proven approach. Logcluster, which is based on frequent item-set mining has been used for this purpose.

1.4 SCOPE

Log messages based on their source can vary in the way their languages are formed. For this reason, the focus of the research work is to use dataset from one of the widely used and most operating system that is Windows OS. The other reason for using Windows OS logs is that it has more verbose logging, which in many cases resembles to the natural language. Furthermore, Exchange Mail Server logs is also used as a dataset because the nature of the logs are similar to that of Windows. The dataset was collected from two different industries. However, on top of these standard sources, synthetic log data prepared manually was initially used in the experiments to develop the initial model.

2 RELATED LITERATURE

2.1 BACKGROUND THEORY

2.2 NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) [2] is an approach in computer science, which deals with interactions between computers and human language. There are two components of NLP. Natural Language Understanding and Natural Language Generation.

There are five steps in Natural Language Understanding

Lexical Analysis

It involves breaking texts in chunks of paragraphs, lines, and words. This is also known as tokenization. Stop words that do not add contextual meaning are removed.

Parsing

It involves analyzing words in a sentence that gives meaningful information. The sentence that does not follow a standard grammar is rejected. It creates a context-free grammar that can be used to perform Semantic Analysis.

Semantic Analysis

It is the process of extracting meaning from the extracted text. Anything that is not meaningful is discarded by performing Semantic Analysis.

Discourse Integration

It defines the meaning by looking up to sentences before or after a given sentence.

Pragmatic Analysis

There are some good uses of named entity recognition such as identifying relationships between the entities, sentiment analysis, answering questions and most importantly information extraction is fundamentally extraction of correct named entities.

There are three types of approaches in named entity recognition.

Rule Based Approach

The Rule-Based approach uses a list of triggered words. Rules are defined based on regular expression patterns. This approach needs a human effort to create the knowledge base for a regex-based parser.

Statistical Approach

The automated machine learning based approach uses Support Vector Machines (SVM) [3] and other approaches like Hidden Markov Model (HMM), Maximum Entropy Models, Decision Trees, and Conditional Random Fields (CRF), etc. I plan to focus on Naïve Bayes and SVM.

Hybrid Approach

The Hybrid method is a combination of rule-based approach and statistical approach.

2.3 FREQUENT ITEM-SET MINING

Particularly in the case of log messages, Frequent Item-Set Mining [4] can be used for extracting variables and constants. These variables are actually defining the named objects. However, it might be difficult to actually confirm the type of named entity with this approach.

2.4 CLUSTERING

Clustering [5] is an unsupervised classification process in which a given set of objects are classified such that the objects within a group are more similar to each other compared to those outside the group. Log Clustering can be valuable in the high-level classification of log messages. However, this does not completely solve the problem of information extraction. Which can be achieved through named entity recognition.

2.5 SIEM

A **SIEM** [6] solution, also known as "Secure Information and Event Management" is a tool that collects log data from various sources, extracts the information contained in the collected data, stores and then uses the stored information for correlating, alerting and reporting as per the security, compliance, operations and business needs.

2.5.1 NAMED ENTITY RECOGNIZATION

There are three types of approaches in named entity recognition.

Rule Based Approach

The Rule Based approach uses list of triggered words.

Statistical Approach

The automated machine learning based approach uses Support Vector Machines (SVM) and other approaches like Hidden Markov Model (HMM), Maximum Entropy Models, Decision Trees, and Conditional Random Fields (CRF), etc. The selected approach focused on the use of Naïve Bayes classifier. And as there are only two outcomes under the defined scope it was a binomial classifier.

Hybrid Approach

The Hybrid method is a combination of both rule-based and statistical approach.

2.5.2 PART OF SPEECH TAGGING

Part of speech (POS) tagging, also known as grammatical tagging or word-category disambiguation, is the process of marking up a word in a text as corresponding to a particular part of speech. This is based on both its definition and its context i.e., its relationship with adjacent and related words in a phrase, sentence, or paragraph.

Once performed by hand, POS tagging is now done in the context of computational linguistics, using algorithms, which associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags. POS tagging algorithms fall into two distinctive groups: rule-based and stochastic. Regular expressions can also be used to perform POS tagging.

2.5.3 SUPPORT VECTOR CLASSIFIER

Support vector classifier is a supervised learning model that analyze data used for classification and regression analysis. For a given a set of data points, each labelled as belonging to one of the available categories, it builds a model by assigning the data points one category or the other. A good separation between the two classes is possible using a hyperplane with the largest distance to the nearest data point in the training sample. As the margin is larger, lower is the error of the classifier. If a support vector machine is used for classification tasks it is known as support vector classifier.

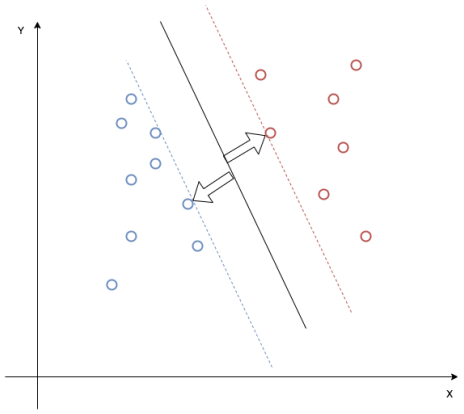

Figure 4 Support vector classifier

2.5.4 NAÏVE BAYES CLASSIFIER

A Naïve Bayes classifier is a classifier based on Bayes theorem. Bayesian [6] classifiers are statistical classifiers. These are used to predict class membership probabilities. For example, calculate the probability that a given tuple belongs to a particular class. It is based on Bayes' theorem. Bayesian classifier gives high accuracy with high speed when applied to large databases.

Naive Bayes is a classification algorithm, which can be applied to two or more classes. The classification method is easy to implement for both binary and categorical input values.

Bayes theorem is represented as.

P(A/B) = [P(B/A) P(A)]/P(B)

Bayes theorem can be used to find the probability of occurrence of A, such that B has occurred. All the predictions or the features should be independent. As the presence of one feature does not affect other it is called Naïve Bayes.

Bayes theorem can further be written as

P(y/X) = [P(X/y) P(y)]/P(X)

2.6 LITERATURE REVIEW

Risto Vaarandi et al [7] in their research work implemented the LogCluster algorithm to discover line patterns, anything that did not match the line patterns were treated as outliers. It was claimed that this approach performed better compared to the existing approaches. In this research work, they introduced the LogClusterC event log-mining tool and described a number of experiments to evaluate its performance against other publicly available log clustering tools. The experiments revealed that LogClusterC performed better compared to other algorithms and tools, and was able to efficiently mine large event logs.

Risto Vaarandi et al [8] in their research work explained about the use of LogClustering to mine log messages and discover anomalous events. They presented the LogCluster tool for mining line patterns and outlier events from textual event logs. They also described different scenarios for discovering security incidents and anomalous events. For more detailed information on its performance comparison was done with other log clustering algorithms.

Risto Vaarandi et al [9] in their research work implemented an algorithm to extract clusters in log messages using frequent item-set mining based on Perl.

Risto Vaarandi et al [10] in their research work for the first-time proposed approach to detect frequent patterns from log messages. They conducted experiments with SLCT and confirmed that the tool can be used to build log file profiles and detecting interesting patterns from the log file. SLCT was also instructed to identify outlier points; four passes over the data were made altogether during the experiments.

In all of the above approaches, the fundamental thing that was considered was the identification of objects as variables using frequent item-set detection. First, the algorithm calculated the count the number of appearances of each word in a set of logs. Then, it checked the words that appeared more frequently than a given threshold, which was derived empirically. Thus, it generated a template by replacing variables by wildcard. This algorithm was based on the assumption that constant words appear more frequently compared to variable words in the system log. For example, user names appear more frequently than the description in a given context. However, this assumption is not always correct. There can be cases where there are no variations in the variable fields. Log messages that are considered, as outliers might not actually be outliers. Similarly, this approach needed adjustments in parameters such as support, which is always not feasible. All of these findings suggested that there was some space for improvement.

Basanta Joshi [11] et al. in their research work identified an efficient approach for clustering of log messages. In the approach, they generated signatures which were used to define log cluster based on the percentage similarity of these signatures against the log messages. For an entirely new category of logs, which did not fall into the existing cluster or matched with an existing signature, a new signature was generated. Thus, this approach proved to be very intelligent for the purpose of log clustering.

Tobias Eka [12] et al. in their research work extracted named entities from Short Text Messages. Unlike most approaches, which implement NLP in a structured data set, in this research work named entities were extracted from SMS messages from Swedish text. Entities such as locations, names, dates, times, and telephone numbers were extracted so that these entities could be utilized by other applications running on the phone.

David Jaeger [13] et al. in their research work explained the use of hierarchical normalization for efficient normalization. They explained that hierarchical normalization will outperform flat normalization when it comes to parsing of big data. They organized normalization in multiple levels by using a hierarchical KB consisting of normalization rules. A performance gain of about 1000x with our presented approaches was achieved, on comparison with existing normalization solutions.

Tome Eftimov [14] et al. in their research work explained a rule-based approach for extraction of knowledge-based evidence from dietary recommendations. In this paper, they presented an approach for knowledge extraction of evidence-based dietary information. The approach used a rule-based NER that consisted of two phases. The first one involves the detection and determination of the entities mention, and the second one selected and extracted the entities.

Chenliang Li [15] et al. in their research work they created segments from tweets and then used those segments for Named Entity Recognition. They conducted experiments on two tweet data sets to show that tweet segmentation quality was significantly improved by learning both global and local contexts compared to using just global context.

Ertopçu [16] et al. in their research work discovered a new approach for Named Entity Recognition. They used the trained continuous representation of words to feed the classifiers as continuous features of words without any feature enhancement. The results showed that the continuous models for NER classification tasks performed as good as supervised and manually handcrafted discrete features.

3 METHODOLOGY

3.1 PROPOSED MODEL

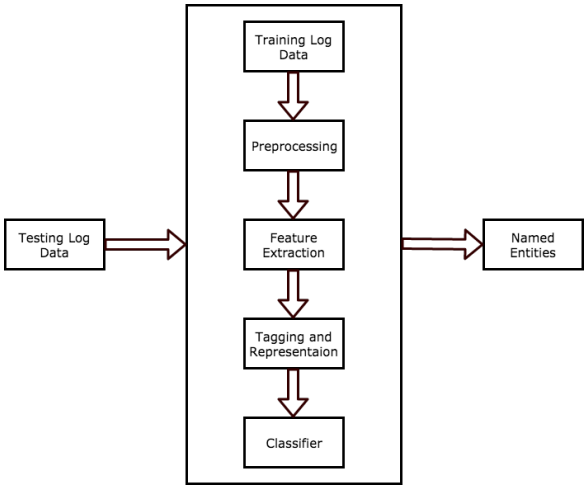

Figure 5 Proposed approach

The proposed model ingested the training data set in batches for which the names for the required entities had to be extracted. The batch data went through the following steps:

(1) Training Data

This was the first step in the process of extraction of named entities. This data contained labels. The labels were defined by passing the data through a SIEM [17] box to extract the standard named entities. Based on the extracted entities the data was labeled as True or False.

(2) Preprocessing

Preprocessing of the training data was done by removing all the unwanted words or tokens or any unwanted stop words.

(3) Feature Extraction

Following were the features that were defined for the classifier model:

- Previous POS
- Next POS
- POS
- Previous Word
- Next Word
- Word shape (Four types)

TABLE I FEATURE CONTRIBUTION

S. N	Contribution
1	prevpos = 'Time' True : False = 223.2 : 1.0
2	nextpos = 'Type' True : False = 223.2 : 1.0
3	prevword = 'datetime' True : False = 169.8 : 1.0
4	nextword = 'Microsoft-Windows-Security-Auditing' True : False = 156.7 : 1.0
5	shape = 'AAAAAAAAAAjggjxxA' True : False = 110.4 : 1.0
6	nextpos = u'VBN' True : False = 31.6 : 1.0
7	prevword = 'account' True : False = 24.0 : 1.0
8	nextword = 'created' True : False = 22.2 : 1.0
9	prevpos = 'Obj' True : False = 9.9 : 1.0
10	prevword = 'User' True : False = 7.6 : 1.0
11	pos = 'NN' True : False = 5.3 : 1.0
12	nextword = 'from' True : False = 4.7 : 1.0
13	nextpos = '<END>' True : False = 2.9 : 1.0
14	nextword = '<END>' True : False = 2.0 : 1.0
15	nextpos = u'IN' True : False = 1.9 : 1.0
16	prevpos = 'NN' True : False = 1.9 : 1.0
17	nextpos = 'NN' False : True = 1.7 : 1.0

Figure 6 Break down of message part

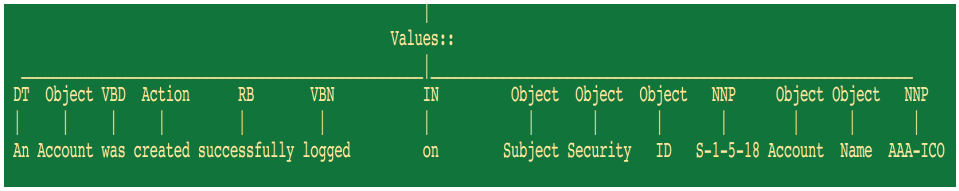

Figure 7 Category 1: Break down of message part

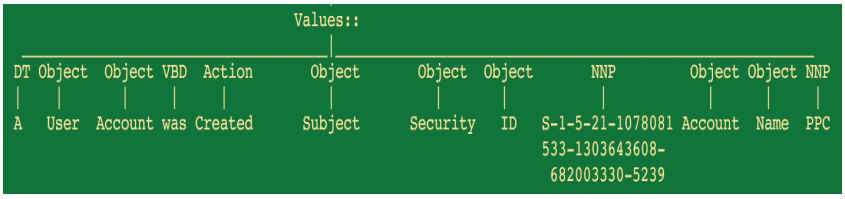

Figure 8 Category 2: Break down of message part

Figure 9 Category 3: Break down of message part

TABLE II TAGGING OF TOKENS WITH LABELS

S. N	Labelled Tokens
1	('<', 'SYM', False)
2	('14', 'CD', False)
3	('>', 'SYM', False)
4	('Apr', 'MON', False)
5	('21', 'CD', False)
6	('05:44:28', 'TIME', False)
7	('CABURDCW01.pp10.net', 'NN', True)
8	('Microsoft-Windows-Security-Auditing', 'TYPE', False)
9	('[', 'SYM', False)
10	('548', 'CD', False)
11	(']', 'SYM', False)
12	(':', u':', False)
13	('User', 'OBJ', False)
14	('account', u'OBJ', False)
15	('ghr', 'NN', True)
16	('created', u'Action', False)
17	('by', u'IN', False)

(4) Classifier Model

The classifier model was created based on Support Vector Classifier as well as Naïve Bayes Classifier, which used the above-mentioned features.

(5) Testing

Testing was done by performing 10-fold cross validation on the original dataset.

3.2 DATA COLLECTION

The data set for information extraction was taken from industry, as this kind of data was not commonly available for research. The collected data was anonymized. This collected data was passed through a trial version of a SIEM box in order to parse the information. This information was then used to construct labeled training data.

The research focused on following types of data sets:

Synthetic Data

This was used for initial research

Windows OS Event Logs

This was collected from the real standard industry infrastructure. The thesis work was entirely based on this data set.

Randomly split Windows OS Event Logs

This was done to check if the accuracy of the model changed or not after the length of the log was varied. In this dataset the length of the logs was randomly varied by splitting the standard Windows OS Events Logs between 2 to 10 folds.

Exchange Mail Logs

This was collected from the real standard industry infrastructure.

Example Categories of Windows OS Event Logs:

- User Kerberos Authentications
- Computer Kerberos Authentications
- NTLM Authentications
- Account Successful Login
- Account Failed Login
- Account Logoff
- Account Locked and unlocked
- User Account Management (created, deleted, disabled, moved, added and removed from group)
- Computer Account Management (groups created, deleted)
- Security Group Management (groups created, deleted)
- Distribution Group Management (groups created, deleted)
- Application Group Management (groups created, deleted)
- Other Group Management (groups created, deleted)
- Audit log cleared
- Object Auditing (request, access, delete, close)

3.3 STEPS IN FEATURE GENERATION

The actual feature generation process went through the following steps:

(1) Collect Data

Synthetic data was prepared manually while, industry standard Security Audit Event Logs were collected from Windows Operating System for 50 different log categories as specified above.

(2) Pass through SIEM

Data was passed through a freely available SIEM Box. The objective was to extract all the required named entities from the log messages.

(3) Preprocessing

All of the unwanted characters and stop words were removed from the collected data.

(4) Named Objects Identified

All of the extracted Named Entities were stored in a separate file.

(5) Cleaning Data

Output of data processing was clean data, which was used to define grammar and create labels using the identified named objects.

(6) Labeling Data

Named Objects Identified from SIEM Box were used to create labeled Data by comparing against the Clean Data.

(7) Create Tokens

Tokens of words, IP addresses and numbers were created from cleaned labeled data

(8) Define Shape

Four classes of shapes were defined based on the length and presence of certain characters in the extracted tokens.

(9) POS Tagging

Parts of speech tagging was done on the extracted tokens-based unigram bigram and trigram approach. Tagging was based on the grammar definition that was created by splitting a log message into header and message sections.

(10) Features Ready

Finally, the features were ready to be used in the classifier.

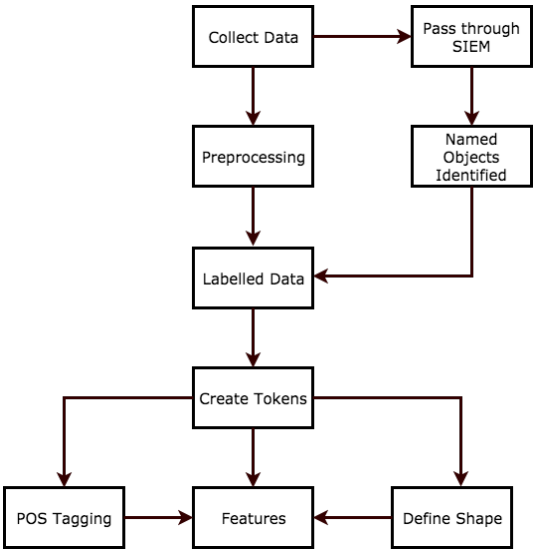

Figure 10 Block diagram for feature generation

3.4 VALIDATION

Validation of the model was performed using the following measures:

- Accuracy
- Precision
- Recall

4 RESULTS AND DISCUSSION

4.1 EXPIREMENTAL SETUP

Log definition for the given sources provided the standard semantics for the strings and sub-strings inside the log messages. The validation against the standard Log definition was used to calculate the performance metrics. Standard Log was built by using a standard SIEM solution and then it was used to create a labeled data. The standard outcome of the experiment was compared with the labeled data to validate the result. A variety of log samples were used in the experiment as explained below in Figure 9, Figure 10 and Figure 11.

4.1.1 TRAINING PHASE

Before the training phase, the Windows Security Event Logs were passed through a trial version of a commercial SIEM solution. This was used to parse the information and extract the entities such as host names, user names, computer names, etc. These values were then checked in the data set and labeled as true or false based on if they were available in the data set or not. Finally, this data was used as a labeled dataset to train a model. The classifier model was built using both Naïve Bayes and Support Vector classifier. Using the data set, three kind of experiments were conducted.

- By increasing the number of log categories.
- By randomly varying lengths of log messages.
- On entirely new data source.

TABLE III TYPE 1 LOG SAMPLE

Log Source	Windows OS
Categories	123
Avg. Tokens	125
Avg. Named Entities	4
Minimum Samples Per Category	1
Maximum Samples Per Category	1019

TABLE IV TYPE 2 LOG SAMPLE

Log Source	Randomly split Windows OS
Categories	123
Avg. Tokens	40
Avg. Named Entities	1
Split Ratio	1:2 to 1:10
Minimum Samples Per Category	5
Maximum Samples Per Category	1019

TABLE V TYPE 3 LOG SAMPLE

Log Source	Exchange Mail
Categories	21
Avg. Tokens	110
Avg. Named Entities	2
Minimum Samples Per Category	1
Maximum Samples Per Category	389

4.2 RESULTS AND OBSERVATIONS

The proposed approach was used for information extraction in the form of Named Entities from Windows Security Event Logs. The experiment was also extended to Exchange Mail Server Logs.

4.2.1 TESTING PHASE

Testing was done through cross validation on the original data. For this the data was randomly split into 10 folds out of which 9 folds were used for training while the remaining 1 fold was used for testing.

TABLE VI PERFORMANCE MATRIX

Log Source	Algorithm	Count	Accuracy	Precision	F Measure
Windows OS	SVM	110300	97.00%	80.10%	75.77%
Windows OS	NB	110300	86.50%	79.65%	85.05%
Randomly Split Windows OS	SVM	110300	96.30%	79.13%	84.90%
Randomly Split Windows OS	NB	110300	95.93%	78.88%	84.70%
Exchange Mail	SVM	90410	97.40%	80.61%	85.12%
Exchange Mail	NB	90410	97.17%	80.11%	84.49%

- In case of Windows, the optimal accuracy was observed to be 97% for Support Vector Classifier and 96.50% for Naïve Bayes.
- In case of randomly split logs from Windows, the optimal accuracy was observed to be 96.30% for Support Vector Classifier and 95.93% for Naïve Bayes
- In case of Exchange, the optimal accuracy was observed to be 97.40% for Support Vector Classifier and 87.10% for Naïve Bayes.

Lower accuracy was seen in case of Naïve Bayes based classification. While the SVM based classifier gave a slight edge in accuracy which was 97% and 97.40% in case of Windows and Exchange respectively. As the SVM classifier gave higher performance metrics all of the comparative results are based on SVM.

TABLE VII SAMPLE OF TEST RESULTS ON SYNTETIC DATA

< 14 > Apr 10 23:45:09 <u>WIN-PP-SYS1.LOCAL</u> Microsoft-Windows-Security-Auditing [748] : User <u>abc</u> logged in successfully from source 192.168.2.1
< 14 > Apr 11 11:43:31 <u> WIN-PP-SYS2.LOCAL </u> Microsoft-Windows-Security-Auditing [504] : User <u>def</u> logged in failed from source 192.168.2.11
< 14 > Apr 21 05:44:28 <u> WIN-PP-SYS3.LOCAL </u> Microsoft-Windows-Security-Auditing [548] : <u>Audit</u> <u>log</u> <u>cleared</u> by ghi
< 14 > Apr 21 05:44:28 <u> WIN-PP-SYS3.LOCAL </u> Microsoft-Windows-Security-Auditing [548] : *Administrator <u>exited</u> <u>process</u>* C : \Windows\System32\notepad.exe
< 14 > Apr 20 15:15:33 <u> WIN-INS-SYS1.LOCAL </u> Microsoft-Windows-Security-Auditing [504] : Authentication failed for user <u>xyz</u> from source 192.168.2.29
< 14 > Apr 10 23:45:09 <u> WIN-INS-SYS2.LOCAL </u> Microsoft-Windows-Security-Auditing [748] : User <u>uvw</u> logged in successfully from source 192.168.2.1

4.2.2 COMPARISON BETWEEN FREQUENT ITEM-SET AND PROPOSED APPROACH

The outcome of the proposed approach was compared with the outcome of the frequent item-set approach. The obtained result was obtained as shown in the figure below. In the outcome, we can see that the proposed approach outperformed the frequent item-set approach as the number of categories increased. The accuracy for frequent item-set approach reduced drastically as the number of categories was increased, while the one for the proposed approach had only a minor reduction in accuracy. Additionally, it was observed that the results from the frequent item-set approach had the slopes with a change in direction within a few variations of log categories. However, the average change in gradient was negative.

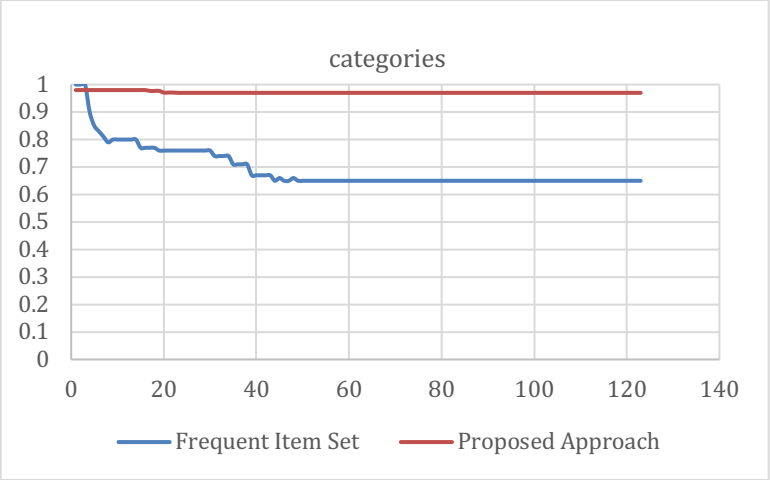

Figure 11 Accuracy comparison between frequent item-set and proposed approach

4.2.3 COMPARING ACCURACY WITH COUNT OF LOGS

For each of the experiments conducted, the accuracy remained fairly constant after a certain rise in samples per category. The figure below shows that the accuracy continues to rise for an increase in the number of log samples up to 10. However, after the 10, no significant rise in accuracy was observed. This result thus suggested that the training data with uniformly distributed samples with few samples of each category performed better compared to the randomly distributed dataset.

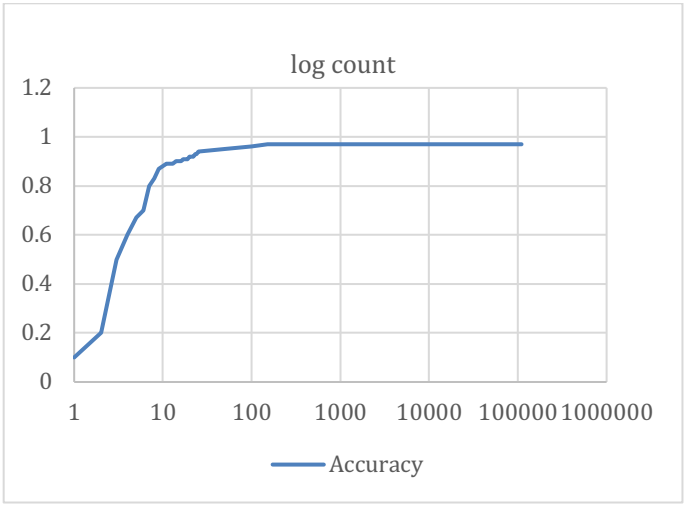

Figure 12 Variation of accuracy with count of logs

4.2.4 COMPUTATIONAL TIME

The average computation rate during the training phase was 9.23 log messages per second for log messages with average size of a log message being 0.78 KB. However, during the testing phase was the computational time was fairly constant with a value of 15 seconds in an average.

4.2.8 CONCLUSION

Classifier models have been used to automatically extract information from log messages. The approach used binomial classification based on Naïve Bayes and Support Vector. The Support Vector Classifier gave a slightly better result compared to the Naïve Based Classifier. The approach was used to conduct experiments on anonymized log dataset from two different industries and producing identical results. The classifier model that was trained on data set from the first organization performed with the same accuracy on a data set from the second organization.

Experiments were conducted on Windows OS and Exchange Mail Server logs. As these log sources resembled with each other in semantics and structure, the results showed only a slight variation in accuracy. The maximum obtained accuracy in Windows OS was 97% while that in Exchange Mail Server 97.40%.

There was only a slight hint of degradation in accuracy as the numbers of log categories increased and compared to the frequent item-set approach where the accuracy reduced drastically. In case of frequent item-set approach, the accuracy was 100% when there were only a few log categories in the available dataset. And, when the number of categories was increased to 50 and above, the accuracy dropped to about 65%. However, with the proposed approach the accuracy remained fairly constant

It was also discovered that the accuracy of the proposed model reached saturation at a count of 10 to 15 logs per category. Not much improvement in accuracy was seen after this threshold. However, the model performed poorly when the number of samples per category was less than 5 or 6.

4.2.9 FUTURE WORK

The thesis work paved the way for the use of NLP in log parsing and analysis. The most important area if enhancement in this area would be to form a hybrid approach for log signature generation which uses clustering and NER. Additionally, the current work can be extended in a number of other areas such as performance enhancement and usage of the multinomial classifier. Works can be performed in areas of real-time streaming log data. Further for practical and industrial implementation, a regex template generator can be built which can be used to extract the named entities without actually using the classifier model.

5 REFERENCES

[1] Chris Phillips, Kevin Schmidt and Anton Chuvakin. “Logging and Log Management”, 2012.

[2] Steven Bird, Ewan Klein and Edward Loper, “Natural Language Processing with Python”, 2019

[3] Meyer, David, Friedrich Leisch, and Kurt Hornik. "The support vector machine under test." Neurocomputing 55.1-2 (2003): 169-186.

[4] Han, Jiawei, Jian Pei, and Yiwen Yin. "Mining frequent patterns without candidate generation." ACM sigmod record. Vol. 29. No. 2. ACM, 2000.

[5] Jain, Anil K., and Richard C. Dubes. Algorithms for clustering data. Vol. 6. Englewood Cliffs: Prentice hall, 1988.

[6] Rish, Irina. "An empirical study of the Naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. 2001.

[7] Zhuge, Chen, and Risto Vaarandi. "Efficient Event Log Mining with LogClusterC." 2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids). IEEE, 2017.

[8] Vaarandi, Risto, Markus Kont, and Mauno Pihelgas. "Event log analysis with the LogCluster tool." MILCOM 2016-2016 IEEE Military Communications Conference. IEEE, 2016.

[9] Vaarandi, Risto, and Mauno Pihelgas. "LogCluster-A data clustering and pattern mining algorithm for event logs." 2015 11th International Conference on Network and Service Management (CNSM). IEEE, 2015.

[10] Vaarandi, Risto. "A data clustering algorithm for mining patterns from event logs." Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IPOM 2003) (IEEE Cat. No. 03EX764). IEEE, 2003.

[11] Joshi, Basanta, Umanga Bista, and Manoj Ghimire. "Intelligent clustering scheme for log data streams." International Conference on Intelligent Text Processing and Computational Linguistics. Springer, Berlin, Heidelberg, 2014.

[12] Tobias Eka * et al: “Named Entity Recognition for Short Text Messages”, Procedia - Social and Behavioral Sciences 27 (2011) 178 – 187

[13] David Jaeger et al: “Normalizing Security Events with a Hierarchical Knowledge Base”, 9th Workshop on Information Security Theory and Practice (WISTP), Aug 2015, Heraklion, Crete, Greece.

[14] TOME EFTIMOV ET AL: "A RULE-BASED NAMED-ENTITY RECOGNITION METHOD FOR KNOWLEDGE EXTRACTION OF EVIDENCE BASED DIETARY RECOMMENDATIONS", PLOS ONE | HTTPS://DOI.ORG/10.1371/JOURNAL.PONE.0179488 JUNE 23, 2017

[15] CHENLIANG LI ET AL: "TWEET SEGMENTATION AND ITS APPLICATION TO NAMED ENTITY RECOGNITION", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (VOLUME: 27, ISSUE: 2, FEBRUARY 1 2015)

[16] ERTOPÇU ET AL: "A NEW APPROACH FOR NAMED ENTITY RECOGNITION", COMPUTER SCIENCE AND ENGINEERING (UBMK), 2017

[17] ASIF EKBAL AND SIVAJI BANDYOPADHYAY: "NAMED ENTITY RECOGNITION USING SUPPORT VECTOR MACHINE: A LANGUAGE INDEPENDENT APPROACH", WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING VOL:4, NO:3, 2010

[18] H. ALANI, SANGHEE KIM, D.E. MILLARD: "AUTOMATIC ONTOLOGY-BASED KNOWLEDGE EXTRACTION FROM WEB DOCUMENTS", IEEE INTELLIGENT SYSTEMS (VOLUME: 18, ISSUE: 1, JAN-FEB 2003)

[19] DAVID CARASSO, "EXPLORING SPLUNK", 2012.