# CHAPTER 5
# EVALUATIONS, RESULTS AND DISCUSSION

## 5.1 BACKGROUND

This chapter discusses and presents results for evaluating detection and classification process using the proposed method. Then, the results are analyzed, discussed and compared to ensure the proposed model is usable and provide better results. Besides, this chapter also presents experiments to proof the concept of detection and classification using WEKA to help in understanding how the process happen using machine learning tools such as WEKA and the difference between these two processes (i.e. detection and classification). Another experiment is the proof of performance to report the performance of proposed algorithms and method in terms of accuracy and reliability in detection and classification of the SMS messages.

## 5.2 PROOF OF CONCEPT

Five experiments have been conducted to explore the concept and process of detection (i.e. classification) and classification (i.e. clustering) using WEKA. From these experiments, it gives more understanding on how to use WEKA and how it works toward SMS messages.

### 5.2.1 EXPERIMENT 1 (FORMULA IDENTIFICATION AND UNDERSTANDING)

This experiment is about the formula identification and understanding in WEKA for detection while for classification, results shows in term of number of instances in each cluster. The aim of this experiment is to further study about formulas available in WEKA for reporting detection results and how they are presented. It was found that WEKA shows results in term of accuracy, true positive rate, false positive rate, precision, recall, F-measure and also shows the ROC curve area to compare the performance of classifiers.

The formulas are discussed below.

i. Correctly classified instances /accuracy = number of instances that are correctly detected into defined group.
   a. (No. of correct prediction) / (Total No of predictions) * 100
ii. Incorrectly classified instances /Error rate = number of instances that are incorrectly detected into defined group.
   a. (No. of incorrect prediction) / (Total No of predictions) * 100
iii. Kappa = chance-corrected measure of agreement between the classifications and the true classes.
   a. Value of 1 : perfect agreement i.e. full on agreement upon the classification process by the rate.
   b. Value of 0 : the agreement not better than expected by chance.
   c. K > 0 : the classifier is doing better compared to chance thus indicating perfect agreement at k = 1.
   d. K = 0 : it denotes the change agreement.
   e. A kappa with the negative rating indicates worse agreement than that expected by chance.

f. Formula =

$K = (P_A - P_E) / (1 - P_E)$ where,

$P_A$ = Observed % age agreement

$P_E$ = the chance (hypothetical) % age agreement.

In the above case, for calculating k, where:

Total Instances + TP + FP + TN + FN.

$P_A$ = (aa+bb) / Total Instances

$P_E$ = (Pr (Predicted A) * Pr (Actual A)) + (Pr (Predicted B) * Pr (Actual B))

iv. Mean absolute error = a quantity used to measure how close forecasts or predictions are to the eventual outcomes.

v. Root mean squared error = a good measure of the model's accuracy.

vi. Root relative squared error = the average of the actual values.

vii. Relative absolute error = similar to the relative squared error.

viii. TP Rate (True Positive Rate) = the report of the positive instances classified as positive.

a. TPR = (TP) / (TP + FN)

ix. FP Rate (False Positive Rate) = the report of the negative classified instances as positive.

a. FPR = (FP) / (FP + TN)

x. Precision, P = (TP) / (TP + FP)

xi. Recall = same value with TPR

xii. F Measure = measure of a test's accuracy

a. (2 * R * P) / (R + P)

xiii. ROC (Receiver Operating Characteristics) = for analysing and illustrating the performance of various systems by using the four basic types / groups of classification:

- True Positive (TP) = correct positive prediction.
- False Positive (FP) = incorrect positive prediction.
- True Negative (TN) = correct negative prediction.
- False Negative (FN) = incorrect negative prediction.

a. The ROC curve is given by the TP Rate and FP Rate.
b. The area under the ROC curve (AUC) is a method of measuring the performance of the ROC curve.
c. The AUC getting large means that the classifier is getting better.
d. If AUC is above 0.5 = the classifier is working and prediction is perfect.
e. If AUC below 0.5 = the classifier is anti-learning (i.e. performance of classifiers slower) or the prediction is random.

xiv. Confusion Matrix = to visualize the performance of algorithm. Each column represents the instances in a predicted class while each row represents the instances in an actual class.

| | Predicted | |
|---|---|---|
| | a | b |
| a | TP | FN |
| b | FP | TN |

Actual Class

**Figure 5.1:** Confusion Matrix in WEKA

UCI dataset contains 5572 messages (i.e. 4825 ham and 747 spam) was used to classify ham and spam using four types of classification algorithms that are Naïve Bayes (NB), Support Vector Machine (SVM), k-Nearest Neighbour (k-NN) and Decision Tree (DT). These classification algorithms were chosen because they are familiar and mostly used by researchers in detection process. The results for each algorithm are shown below with calculation.

a. Naïve Bayes (NB)

**Table 5. 1:** Classification using NB

| | **Naïve Bayes** |
|---|---|
| Correctly Classified Instances / accuracy | 5306 @ 95.23% |
| Incorrectly Classified Instances / error | 266 @ 4.77% |
| Kappa statistic | 0.8046 |
| Mean absolute error | 0.0637 |
| Root mean squared error | 0.1947 |
| Relative absolute error | 27.4338% |
| Root relative squared error | 57.1502% |
| Time taken to build model | 0.83 seconds |

Correctly classified instances/ accuracy = (No. of correct prediction) / (Total No of predictions) * 100

= (5306) / (5572) * 100

= 95.23 %

Incorrectly classified instances/ Error rate = (No of wrong predictions) / (Total No of predictions) * 100

= (226) / (5572) * 100

= 4.77 %

Kappa statistic= $(P_A-P_E)/ (1-P_E)$

= (0.9523 – 0.7557) / (1-0.7557)

= 0.1966/ 0.2443

= 0.8047

$P_A$= (aa+bb) / Total Instances

= (5306) / 5572

= 0.9523

$P_E$= (Pr (Predicted A) * Pr (Actual A)) + (Pr (Predicted B) * Pr (Actual B))

= (4733/5572 * 4825/5572) + (839/5572 * 747/5572)

= (0.8494 * 0.8659) + (0.1506 * 0.1341)

= (0.7355) + (0.0202) = 0.7557

```
=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area  Class
                 0.963     0.116     0.982       0.963    0.972       0.974     ham
                 0.884     0.037     0.787       0.884    0.832       0.974     spam
Weighted Avg.    0.952     0.106     0.955       0.952    0.953       0.974
```

**Figure 5. 2:** Results for each class of messages in NB

TPR (ham) = (TP) / (TP + FN)
= (4646) / (4646 + 179)
= 0.963

TPR (spam) = (TN) / (FP + TN)
= (660) / (87 + 660)
= 0.884

FPR (ham) = (FP) / (FP + TN)
= (87) / (87 + 660)
= 0.116

TPR (spam) = (FN) / (TP + FN)
= (179) / (4646 + 179= 0.963
= 0.037

Precision, P (ham) = (TP) / (TP + FP)
= (4646) / (4646 + 87)
= 0.982

Precision, P (spam) = (TN) / (FN + TN)
= (660) / (179 + 660)
= 0.787

F-Measure (ham) = (2 * R * P) / (R + P)
= (2* 0.963*0.982) / (0.983 +0.963)
= 0.972

F-Measure (spam) = (2 * R * P) / (R + P)
= (2* 0.884*0.787) / (0.884 +0.787)
= 0.83

ROC area = 0.9739

= This result suggests that a value of ROC curve over 0.5 indicate that it is perfect and classifier is working as it supposed to work.

```
=== Confusion Matrix ===

    a    b   <-- classified as
 4646  179 |    a = ham
   87  660 |    b = spam
```

**Figure 5. 3:** Confusion Matrix for NB

**Table 5. 2:** Explanation results from Confusion Matrix for NB

| | |
|---|---|
| TP | 4646 messages are correctly predicted as ham messages. |
| FP | 87 messages are incorrectly predicted as ham messages. The actual messages are spam messages. |
| TN | 660 messages are correctly predicted as spam messages. |
| FN | 179 messages are incorrectly predicted as spam messages. The actual messages are ham messages. |

The number of messages that are correctly classified as spam and ham is 5306 messages with the accuracy is 95.23 % and the rest is incorrectly classified. For the kappa statistic, the value is greater than 0 (i.e. 0.8047) means the classifier is doing better. About 660 messages from 747 spam messages are correctly detected in spam group while 4646 messages from 4825 ham messages into ham group. Time taken require to build the model for detection is less than 1 seconds show that NB is faster in detection process.

b. Support Vector Machine (SVM)

**Table 5. 3:** Classification using SVM

| | **Support Vector Machine** |
|---|---|
| Correctly Classified Instances / accuracy | 5537 @ 99.37% |
| Incorrectly Classified Instances / error | 35 @ 0.63% |
| Kappa statistic | 0.9724 |
| Mean absolute error | 0.0063 |
| Root mean squared error | 0.0793 |
| Relative absolute error | 2.7043% |
| Root relative squared error | 23.2611% |
| Time taken to build model | 2.35 seconds |

Correctly classified instances/ accuracy = (No. of correct prediction) / (Total No of predictions) * 100

= (5537) / (5572) * 100

= 99.37 %

Incorrectly classified instances/ Error rate = (No of wrong predictions) / (Total No of predictions) * 100

= (35) / (5572) * 100

= 0.63 %

Kappa statistic $= (P_A - P_E) / (1 - P_E)$

$= (0.9937 - 0.7723) / (1 - 0.7723)$

$= 0.2214 / 0.2277$

$= 0.9724$

$P_A$ = (aa+bb) / Total Instances

= (5537) / 5572

= 0.9937

$P_E$= (Pr (Predicted A) * Pr (Actual A)) + (Pr (Predicted B) * Pr (Actual B))

= (4860/5572 * 4825/5572) + (712/5572 * 747/5572)

= (0.8722 * 0.8659) + (0.1278 * 0.1341)

= (0.7552) + (0.0171) = 0.7723

```
=== Detailed Accuracy By Class ===

                TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area  Class
                1.000     0.047     0.993       1.000    0.996       0.977     ham
                0.953     0.000     1.000       0.953    0.976       0.977     spam
Weighted Avg.   0.994     0.041     0.994       0.994    0.994       0.977
```

**Figure 5. 4:** Results for each class of messages in SVM

TPR (ham) = (TP) / (TP + FN)
= (4825) / (4825 + 0)
= 1.000

TPR (spam) = (TN) / (FP + TN)
= (712) / (35 + 712)
= 0.953

FPR (ham) = (FP) / (FP + TN)
= (35) / (35 + 712)
= 0.047

TPR (spam) = (FN) / (TP + FN)
= (0) / (4825 + 0)
= 0.000

Precision, P (ham) = (TP) / (TP + FP)
= (4825) / (4825 + 35)
= 0.993

Precision, P (spam) = (TN) / (FN + TN)
= (712) / (0 + 712)
= 1.000

F-Measure (ham) = (2 * R * P) / (R + P)
= (2* 1.000*0.993) / (1.000 +0.993)
= 0.996

F-Measure (spam) = (2 * R * P) / (R + P)
= (2* 0.953*1.000) / (0.953 +1.000)
= 0.976

ROC area = 0.9766

= The result shows that the value of ROC area is above 0.5 indicate that the classifiers is good in performance and working as it supposed to work.

```
=== Confusion Matrix ===

   a    b   <-- classified as
4825    0 |    a = ham
  35  712 |    b = spam
```

**Figure 5. 5:** Confusion Matrix for SVM

**Table 5. 4:** Explanation results from Confusion Matrix for SVM

| | |
|---|---|
| TP | 4825 messages are correctly predicted as ham messages. |
| FP | 35 messages are incorrectly predicted as ham messages. The actual messages are spam messages. |
| TN | 712 messages are correctly predicted as spam messages. |
| FN | No messages are incorrectly predicted as spam messages. |

The number of messages that are correctly classified as spam and ham is 5537 messages with the accuracy is 99.37 %. For the kappa statistic, the value is greater than 0 (i.e. 0.9724) means the classifier is doing better. About 712 messages from 747 spam messages are correctly detected in spam group while 4825 messages from 4825 ham messages into ham group. Time taken require to build the model for detection is more than 2 seconds show that the process using SVM require more time to proceed.

c. k-Nearest Neighbour (k-NN)

**Table 5. 5:** Classification using k-NN

| | **k-Nearest Neighbour** |
|---|---|
| Correctly Classified Instances / accuracy | 5564 @ 99.86% |
| Incorrectly Classified Instances / error | 8 @ 0.14% |
| Kappa statistic | 0.9938 |
| Mean absolute error | 0.0026 |
| Root mean squared error | 0.0349 |
| Relative absolute error | 1.1078% |
| Root relative squared error | 10.2444% |
| Time taken to build model | 0.01 seconds |

Correctly classified instances/ accuracy = (No. of correct prediction) / (Total No of predictions) * 100

= (5564) / (5572) * 100

= 99.86 %

Incorrectly classified instances/ Error rate = (No of wrong predictions) / (Total No of predictions) * 100

= (8) / (5572) * 100

= 0.14 %

Kappa statistic = $(P_A - P_E) / (1 - P_E)$

= (0.9986 - 0.7689) / (1-0.7689)

= 0.2297/ 0.2311

= 0.9938

$P_A$ = (aa+bb) / Total Instances

= (5564) / 5572

= 0.9986

$P_E$= (Pr (Predicted A) * Pr (Actual A)) + (Pr (Predicted B) * Pr (Actual B))

= (4833/5572 * 4825/5572) + (739/5572 * 747/5572)

= (0.8674 * 0.8659) + (0.1326 * 0.1341)

= (0.7511) + (0.0178)

= 0.7689

```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
               1.000    0.011    0.998      1.000   0.999      1.000     ham
               0.989    0.000    1.000      0.989   0.995      1.000     spam
Weighted Avg.  0.999    0.009    0.999      0.999   0.999      1.000
```

**Figure 5. 6:** Results for each class of messages in k-NN

TPR (ham) = (TP) / (TP + FN)
= (4825) / (4825 + 0)
= 1.000

TPR (spam) = (TN) / (FP + TN)
= (739) / (8 + 739)
= 0.989

FPR (ham) = (FP) / (FP + TN)
= (8) / (8 +739)
= 0.011

TPR (spam) = (FN) / (TP + FN)
= (0) / (4825 + 0)
= 0.000

Precision, P (ham) = (TP) / (TP + FP)
= (4825) / (4825 + 8)
= 0.998

Precision, P (spam) = (TN) / (FN + TN)
= (739) / (0 + 739)
= 1.000

F-Measure (ham) = (2 * R * P) / (R + P)
= (2* 1.000*0.998) / (1.000 +0.998)
= 0.999

F-Measure (spam) = (2 * R * P) / (R+P)
= (2* 0.989*1.000) / (0.989 +1.000)
= 0.995

ROC area = 0.9997

=The result shows that the value of ROC area is above 0.5 which is 0.9997 indicate that the classifiers is good in performance and working as it supposed to work.

```
=== Confusion Matrix ===

    a    b   <-- classified as
 4825    0 |    a = ham
    8  739 |    b = spam
```

**Figure 5. 7:** Confusion Matrix for k-NN

**Table 5. 6:** Explanation results for Confusion Matrix for k-NN

| | |
|---|---|
| TP | 4825 messages are correctly predicted as ham messages. |
| FP | 8 messages are incorrectly predicted as ham messages. The actual messages are spam messages. |
| TN | 739 messages are correctly predicted as spam messages. |
| FN | No messages are incorrectly predicted as spam messages. |

The number of messages that are correctly classified as spam and ham is 5564 messages with the accuracy is 99.86 %. For the kappa statistic, the value is greater than 0 (i.e. 0.9938) means the classifier is doing better. About 739 messages from 747 spam messages are correctly detected in spam group while 4825 messages from 4825 ham messages into ham group. Time taken require to build the model require only 0.01 seconds, faster in detection process.

d. Decision Tree (DT)

**Table 5. 7:** Classification using DT

| | **Decision Tree** |
|---|---|
| Correctly Classified Instances / accuracy | 5416 @ 97.20% |
| Incorrectly Classified Instances / error | 156 @ 2.80% |
| Kappa statistic | 0.8706 |
| Mean absolute error | 0.052 |
| Root mean squared error | 0.1613 |
| Relative absolute error | 22.4013% |
| Root relative squared error | 47.3398% |
| Time taken to build model | 32.79 seconds |

Correctly classified instances/ accuracy = (No. of correct prediction) / (Total No of predictions) * 100

= (5416) / (5572) * 100

= 97.20 %

Incorrectly classified instances/ Error rate = (No of wrong predictions) / (Total No of predictions) * 100

= (156)/ (5572) * 100

= 2.80 %

Kappa statistic= $(P_A-P_E)/ (1-P_E)$

= (0.9720– 0.7836) / (1-0.7836)

= 0.1884/ 0.2164

= 0.8706

$P_A$ = (aa+bb) / Total Instances

= (5416) / 5572

= 0.8706

$P_E$= (Pr (Predicted A) * Pr (Actual A)) + (Pr (Predicted B) * Pr (Actual B))

= (4945/5572 * 4825/5572) + (627/5572 * 747/5572)

= (0.8875 * 0.8659) + (0.1125 * 0.1341)

= (0.7685) + (0.0151)

= 0.7836

```
=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  ROC Area  Class
                0.996    0.185    0.972      0.996    0.984      0.939     ham
                0.815    0.004    0.971      0.815    0.886      0.939     spam
Weighted Avg.   0.972    0.160    0.972      0.972    0.971      0.939
```

**Figure 5. 8:** Results for each class of messages in DT

TPR (ham) = (TP) / (TP + FN)
= (4807) / (4807 + 18)
= 0.996

TPR (spam) = (TN) / (FP + TN)
= (609) / (138 + 609)
= 0.815

FPR (ham) = (FP) / (FP + TN)
= (138) / (138 +609)
= 0.185

TPR (spam) = (FN) / (TP + FN)
= (18) / (4807 + 18)
= 0.004

Precision, P (ham) = (TP) / (TP + FP)
= (4807) / (4807 + 138)
= 0.972

Precision, P (spam) = (TN) / (FN + TN)
= (609) / (18 + 609)
= 0.971

F-Measure (ham) = (2 * R * P) / (R + P)
= (2* 0.996*0.972) / (0.996 + 0.972)
= 0.984

F-Measure (spam) = (2 * R * P) / (R + P)
= (2* 0.815*0.971) / (0.815 +0.971)
= 0.886

ROC area = 0.9385

= The result shows that the value of ROC area is above 0.5 which is 0.9385 indicate that the classifiers is good in performance and working as it supposed to work.

```
=== Confusion Matrix ===

    a    b   <-- classified as
 4807   18 |    a = ham
  138  609 |    b = spam
```

**Figure 5. 9:** Confusion Matrix for DT

**Table 5. 8:** Explanation results for Confusion Matrix for DT

| | |
|---|---|
| TP | 4807 messages are correctly predicted as ham messages. |
| FP | 138 messages are incorrectly predicted as ham messages. The actual messages are spam messages. |
| TN | 609 messages are correctly predicted as spam messages. |
| FN | 18 messages are incorrectly predicted as spam messages. The actual messages are ham messages. |

The number of messages that are correctly classified as spam and ham is 5416 messages with the accuracy is 97.20 %. For the kappa statistic, the value is greater than 0 (i.e. 0.8706) means the classifier is doing better. About 609 messages from 747 spam messages are correctly detected in spam group while 4807 messages from 4825 ham messages into ham group. Time taken require to build the model is longer which is 32.79 second.

From this experiment, it helps us in understanding what the results will be presented after detecting process in WEKA and how to calculate them using formula provided. Besides, this experiment assist us in making the comparison of performance between classifiers in WEKA in term of accuracy, time taken and statistics of result. Different types of classifiers algorithms used will give different results due to concept and theory of

algorithms. In other way, this experiment gives an idea on what formula and results that should be shown for our proposed algorithm and method in detection SMS messages.

### 5.2.2 EXPERIMENT 2 (DETECTION USING LABEL AND UNLABELED MESSAGES)

This experiment was carried out to identify the capability of WEKA in detecting SMS messages using label and unlabeled messages. Label message means that each message in dataset has been labeled into ham and spam while unlabeled message is the dataset contains only text messages without knowing whether the message is a spam or ham. Usually labeled messages are used for training and unlabeled messages for testing but in this experiment, we use both types of messages in training to see their capability in detection. UCI dataset was used and only 300 messages were chosen for this experiment (i.e. 200 ham messages and 100 spam messages). Same as experiment 1, five different classifiers algorithms were used that are Naïve Bayes (NB), Support Vector Machine (SVM), k-Nearest Neighbour (k-NN) and Decision Tree (DT) using three different test options to classify the messages. Several steps involves in this experiment:-

1. Dataset used is UCI dataset (i.e. contains 5572 messages but only uses 300 messages).
2. Process of data cleaning is needed in order to filter the format of text messages from String to Nominal due to some classifiers algorithms cannot read the content of dataset that in the form of String.

3. The dataset are run in WEKA using three different test option for detecting (i.e. use training set, cross-validation and percentage split.
4. Results are shown in term of time taken, accuracy and number of messages that are detected into spam and ham.

a) Label messages

Figure 5.10 shows example of label messages. This dataset was cleaned by filtering from String into Nominal and was tested using three test options. Results for each option are discussed as follow.

**ham**,'Ok lar... Joking wif u oni...'
**spam**,'Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C\'s apply 08452810075over18\'s'

**Figure 5. 10** : SMS messages with label spam and ham0

i. Use training set

One of the test options in WEKA for classification process is "Use training set". This option is for training the dataset (i.e. to build the model and for learning) and the results as indicated in Table 5.9.

**Table 5. 9:** Use Training Set

| Test Options | Classification Algorithms | Time taken to build model (s) | Correctly Classified Instances (%) | TP | FP | TN | FN |
|---|---|---|---|---|---|---|---|
| Use Training Set | Naïve Bayes (NB) | 0 seconds | 100% | 200 | 0 | 100 | 0 |
| | Support Vector Machine (SVM) | 0.04 seconds | 100% | 200 | 0 | 100 | 0 |
| | k-Nearest Neighbour (k-NN) | 0 seconds | 100% | 200 | 0 | 100 | 0 |
| | Decision Tree (DT) | 0 seconds | 66.67% | 200 | 100 | 0 | 0 |

From the Table 5.9, it can be found that NB, SVM and k-NN obtained 100% for correctly classified instances, means they are good in performance as compared to DT. Their value of True Positive (TP) is 200 (i.e. about 200 SMS messages that is correctly classified as ham messages) and 100 for True Negative (FN) (i.e. about 100 SMS messages that are correctly classified as spam messages). For the DT algorithm, the percentage of accuracy is 66.67% and it was found that 100 messages incorrectly classified as ham. Overall, NB, k-NN and DT are faster in build the model than SVM.

ii. Cross-validation

Cross-validation is the repeating process of classification into several times by dividing the dataset into 10 parts so that it can reduce the estimation of results. The default of cross-validation is 10, means that the dataset is divided into 10 pieces, 9 pieces use for training and the last piece for testing. Result is reported in Table 5.10.

**Table 5. 10:** Supplied Test Set

| Test Options | Classification Algorithms | Time taken to build model (s) | Correctly Classified Instances (%) | TP | FP | TN | FN |
|---|---|---|---|---|---|---|---|
| Supplied test set | Naïve Bayes (NB) | 0 seconds | 67.33% | 200 | 98 | 0 | 2 |
| | Support Vector Machine (SVM) | 0.04 seconds | 67.33% | 200 | 98 | 0 | 2 |
| | k-Nearest Neighbour (k-NN) | 0 seconds | 67.33% | 200 | 98 | 0 | 2 |
| | Decision Tree (DT) | 0 seconds | 66.67% | 200 | 100 | 0 | 0 |

Results from Table 5.10 shows the percentage of correctly classified instances for all four classifiers is in between 66% - 67.33% and DT is less in performance compares the others. Meanwhile, for the time taken to build the model, all classifiers require 0 seconds, different with SVM that need 0.04 seconds. About 200 messages are correctly classified as ham messages for all algorithms.

iii. Percentage split

Percentage split is used to split the messages into training and testing. The default percentage for splitting is 66% for training and 34% for testing (for this experiment, 198 messages for training and 102 for testing). Table 5.11 shows result using percentage split.

**Table 5. 11:** Percentage Split

| Test Options | Classification Algorithms | Time taken to build model (s) | Correctly Classified Instances (%) | TP | FP | TN | FN |
|---|---|---|---|---|---|---|---|
| Percentage split | Naïve Bayes (NB) | 0 seconds | 62.75% | 63 | 38 | 1 | 0 |
| | Support Vector Machine (SVM) | 0.04 seconds | 62.75% | 63 | 38 | 1 | 0 |
| | k-Nearest Neighbour (k-NN) | 0 seconds | 62.75% | 63 | 38 | 1 | 0 |
| | Decision Tree (DT) | 0 seconds | 61.76% | 63 | 39 | 0 | 0 |

By using percentage split, it already divided the dataset into testing and training dataset and result will show for the testing dataset. From Table 5.11, results show that the accuracy of these four classifiers algorithms is between 63%-65%. For NB, SVM and k-NN, only one message that is correctly classified as spam, 38 messages that are incorrectly classified as ham and 63 messages that are correctly classified as ham. So, it can be suggested that 63 messages are ham and 39 messages are spam messages. Time taken to build the model is same for NB, k-NN and DT which is 0 seconds, faster than SVM.

From this experiment, label messages were tested using three different test options for detection process. Each test option has its own function and will give different results although using same dataset. Therefore making comparison results between test options used is not compatible as it is a requirement to continue the detection process in WEKA. Between these three options, the performance of detection using "use training set" is better in term of time and accuracy compare others.

b) Unlabeled messages

For this experiment, label in each message was removed as shown in Figure 5.11.

?,'Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...'
?,'Ok lar... Joking wif u oni...'
?,'Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C\'s apply 08452810075over18\'s'
?,'U dun say so early hor... U c already then say...'
?,'Nah I don\'t think he goes to usf, he lives around here though'

**Figure 5. 11**: Unlabeled SMS messages

However, WEKA cannot classify the messages due to unidentified numbers of spam and ham messages in dataset as shown in Figure 5.12 and Figure 5.13.

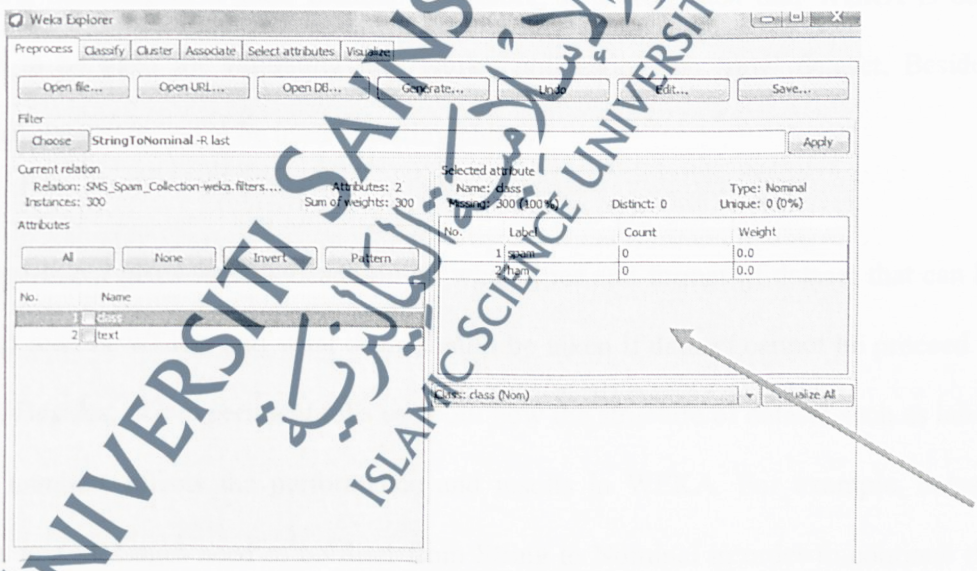


**Figure 5. 12:** Interface after entering dataset in WEKA

Figure 5.12 shows the numbers of spam and ham messages are zero because the label has been removed. WEKA cannot identify the number of messages based on label provided.

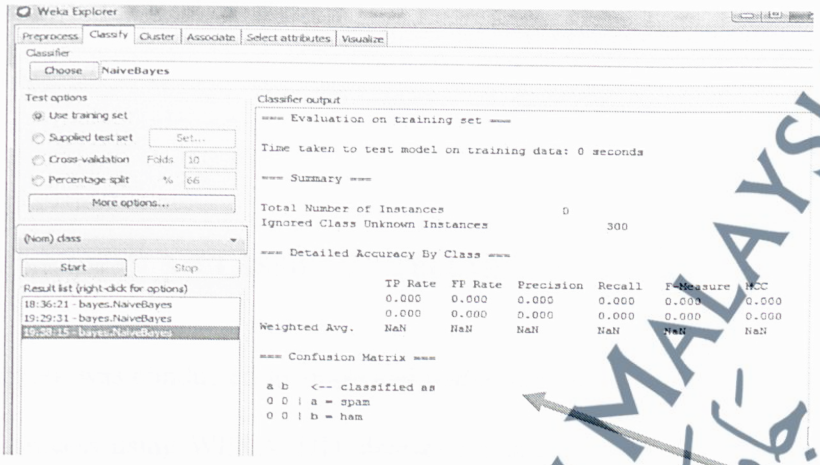


**Figure 5. 13:** Classifier output

Figure 5.13 shows the results for TP rate and FP rate with the accuracy is zero. From this experiment, it can be suggest that by removing the label in dataset will cause WEKA not producing detecting in a good manner. Therefore, it is suggested that WEKA is only suitable to be used for validating the 'work', not testing the 'raw' dataset. Besides, unlabeled dataset only can be used in testing phase.

This experiment gives us understanding on what types and format of dataset that can be used and read in WEKA and what steps should be taken if dataset cannot be proceed in WEKA. Besides, this experiment tells us about how the structure of dataset such as label, text or numbers affects the performance and results in WEKA. For example, dataset contains text messages need to be filter from String to Nominal in order to continue the next step for detection phase. However, we cannot make the comparison of results between both types of dataset from this experiment as WEKA cannot run unlabeled dataset. So we can conclude that unlabeled messages only can be used for testing phase and not suitable in training process. This experiment gives us suggestion about using

various structure of dataset (such as clean and unclean dataset) to see the different results in our research.

### 5.2.3 EXPERIMENT 3 (CLASSIFICATION CONCEPT USING WEKA)

This experiment was conducted to study and understand the classification (also known as clustering) process using WEKA. DIT dataset contains 1352 spam messages was tested with three familiar clustering algorithms; k-Means, Hierarchical and Cobweb. Dataset used is different with previous experiment (i.e. experiment 1 and 2) because it contains only spam messages as we want to do clustering process using spam messages only. In addition, ten groups of spam from this dataset have been identified manually to help in clustering process. Using our own view and understanding the content and meaning of each messages in dataset, we conclude that there are 10 group of spam available from the dataset. Results for clustering are discussed using two different cluster modes. Steps involve in this experiment as follow:-

1. Dataset used is DIT dataset (contains 1352 spam messages).
2. Process of data cleaning is needed in order to filter the format of text messages from String to Nominal due to some clusterers algorithms cannot read the content of dataset that in the form of String.
3. The dataset are run in WEKA using two different cluster mode for clustering (i.e. use training set and classes to clusters evaluation).
4. Results shown in term of time taken, correctly clustered instances and numbers of instances in each cluster.

i. Use Training Set

Use training set is one of the cluster modes in clustering. Table 5.12 shows the output for clustering using use training set. The number of cluster is set into 10 clusters, as manually identified the group.

**Table 5. 12:** Output of clustering using Use Training Set

| Cluster Mode | Clusterer Algorithm | Time taken to build model (s) |
|---|---|---|
| Use Training Set | k-Means | 0.02 s |
| | Hierarchical | 11.57 s |
| | Cobweb | 399.59 s |

From the Table 5.12, it can be found that k-Means produces good performance because it requires only 0.02 seconds to build the model, faster than Hierarchical with 11.57 seconds. Cobweb is the last in performance because times require is higher which is 399.59 seconds. Outputs for each cluster are different depend on formula in each algorithm. Figures 5.14 to Figure 5.16 show the different number of clustered instances for each clusterer algorithm. Clustered instances means that number of spam messages in each cluster.

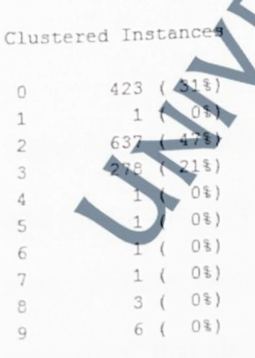


**Figure 5. 14:** k-Means

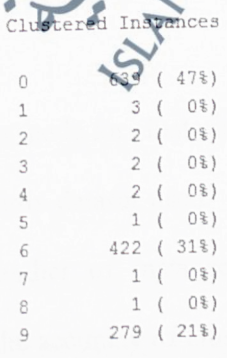


**Figure 5. 15:** Hierarchical

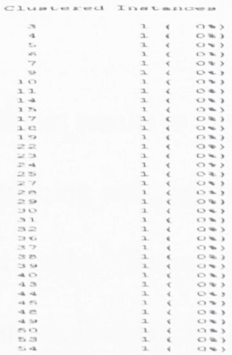

**Figure 5. 16:** Cobweb

ii. Classes to clusters evaluation

Classes to clusters evaluation is another cluster mode for clustering using WEKA. It first ignores the class attribute and generates the clustering, then during the test phase it assigns classes to the clusters based on the majority value of the class attribute within each clusters. Similar with previous experiment, the number of cluster is set into 10 and results are shown in Table 5.13.

**Table 5. 13:** Output of clustering using Classes to Clusters Evaluation

| Cluster Mode | Clusterer Algorithm | Time taken to build model (s) | Correctly Clustered Instances (%) |
|---|---|---|---|
| Classes to clusters evaluation | k-Means | 0.01 s | 78.48 % |
| | Hierarchical | 11.7 s | 78.40% |
| | Cobweb | 385.92s | 47.41% |

Table 5.13 indicates that the highest number of correctly clustered instances is k-Means with the percentage is 78.48% and require minimum time to build the model, 0.01 seconds. Cobweb is the last choice for clustering because it require maximum time for the process which is 385.92 seconds and value for accuracy is lower that other two clusterers algorithms.

Using both approach of cluster mode for clustering gives different results as their function and formula is different towards dataset used. For "Use training set", results show in term of time taken and number of instances, different with "Classes to clusters evaluation" that also having the accuracy of clustering instances. Besides, the number of spam messages in each cluster is same for both approaches

This experiment helps us investigate about the process of clustering (or also called as classification) using two different approach (i.e. using use training set and Classes to clusters evaluation as cluster mode) and identify what the results shown using three different clustering algorithms. It help us improve our proposed algorithms for clustering spam messages in term of correctly clustered by understanding the concept and process of each clustering algorithms in WEKA.

### 5.2.4 EXPERIMENT 4 (PERFORMANCE OF ALGORITHM FOR DETECTION AND CLASSIFICATION)

This experiment was conducted for detection and classification process using UCI dataset. 5572 messages contain ham and spam messages were used. This experiment aims to identify the performance of algorithms in WEKA for detection and classification.

a) Detection

Detection is the process to identify the message into spam or ham. Four types of classification algorithms are used; Naïve Bayes (NB), Support vector machine (SVM), k-Nearest Neighbour (k-NN) and Decision Tree (DT) to test their performance in this experiment. Dataset was filtered from String to Nominal since some algorithms may not be able to read the dataset because of structure of the messages in string. Table 5.14 shows the performance of four classification algorithms using SMS messages dataset for

use training set and Table 5.15 using cross-validation. The process for this experiment as follow:-

1. UCI dataset contains 5572 dataset was used (i.e. 4825 ham and 747 spam).
2. Pre-processing by filtering the dataset from String to Nominal.
3. Running dataset using two test options.
4. Results shows in term of time taken, accuracy and number of ham and spam messages.

There is different between this experiment with previous experiment (i.e. experiment 1 and 2). Experiment 1 is just to show formula calculation for output of detection while experiment 2 is to see how detection process using two different dataset (i.e. label and unlabelled messages). After understanding the both concept, this experiment aim to see the performance of algorithms in WEKA.

**Table 5. 14:** Performance of classification algorithms in WEKA using Use Training Set

| Test Options | Classification Algorithms | Time Taken to build model (s) | Correctly Classified Instances (%) | TP | FP | TN | FN |
|---|---|---|---|---|---|---|---|
| Use training set | Naïve Bayes (NB) | 0.02 | 86.93 | 4825 | 728 | 19 | 0 |
| | Support Vector Machine (SVM) | 1.332996 | 100 | 4825 | 0 | 747 | 0 |
| | k-Nearest Neighbour (k-NN) | 0.84 | 100 | 4825 | 0 | 747 | 0 |
| | Decision Tree (DT) | 0.09 | 86.59 | 4825 | 747 | 0 | 0 |

Table 5.14 shows the performance of four classification algorithms using use training set as test option in WEKA. In term of time taken to build the model, NB is faster than other algorithms while SVM requires longer time. For accuracy, SVM and k-NN are the

highest accuracy in classification with the percentage of correctly classified instances is 100%. The different percentage of accuracy between both NB and DT is only 0.34 %. SVM and k-NN have correctly classified 4825 messages as ham and 747 messages as spam, same with the original messages from dataset. Overall, k-NN is the best classifier algorithm as it has highest accuracy, correctly classified into ham and spam and required less than 1 second to build model.

**Table 5. 15:** Performance of classification algorithms in WEKA using Cross-Validation

| Test Options | Classification Algorithms | Time Taken to build model (s) | Correctly Classified Instances (%) | TP | FP | TN | FN |
|---|---|---|---|---|---|---|---|
| Cross-validation | Naïve Bayes (NB) | 0 | 86.63 | 4825 | 745 | 2 | 0 |
| | Support Vector Machine (SVM) | 1331.64 | 89.45 | 4825 | 588 | 159 | 0 |
| | k-Nearest Neighbour (k-NN) | 0.01 | 89.45 | 4825 | 588 | 159 | 0 |
| | Decision Tree (DT) | 0.27 | 86.60 | 4825 | 747 | 0 | 0 |

From Table 5.15, result indicates that NB is faster in time processing compare to k-NN with different time is 0.01 seconds while SVM require longer time. In term of correctly classified instances, SVM and k-NN have same percentage of accuracy and the highest as compare to NB and DT. All these four algorithms manage to detect 4825 ham messages into correct class of ham, similar with original class. However for spam messages, each of them have different number due to the performance and function of each algorithm. Overall, k-NN is better in detecting spam messages using cross-validation as test option because it require little time and highest accuracy compare to others.

From experiment of classification SMS messages, it can be conclude that k-NN is the best classifiers in term of performance and time for both approach (i.e. use training set and cross-validation). The different results in each algorithm depend on their theory and concept, hence they give various output of detection.

b) Classification

Classification or known as clustering is the process of grouping data documents into similar groups. This experiment was conducted to categorize spam messages into several types of message using three types of clusterers algorithm i.e. Cobweb, Hierarchical and k-Means clustering algorithm. Only spam messages were used from UCI dataset. There are 10 groups of spam messages manually identified in this dataset (i.e. by using own view and understanding the content of messages). The groups are Competition/Game, Chatting, Dating, Prize, Service, Finance, Ringtone, News, Advertisement and Voicemail. The results of performance are shown in Table 5.16.

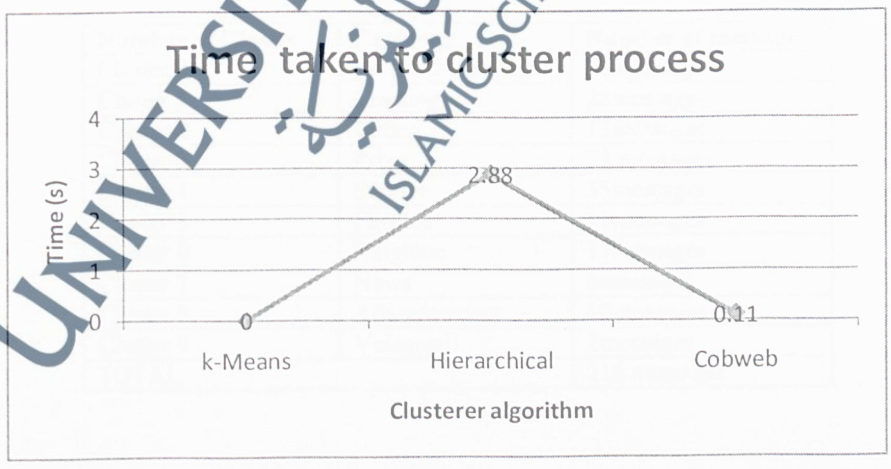


**Figure 5. 17:** Time taken to cluster dataset using Use Training Set

Figure 5.17 shows the performance of clusterer algorithms using WEKA. It can be found that k-Means is good in clustering the dataset with time taken is 0 seconds. Hierarchical is the last with 2.88 seconds and cobweb use only 0.11 seconds. However, this experiment was also carried out manually to compare the number of messages in each cluster using WEKA for k-Means algorithm. The process of manually tested as follow:-

1. Dataset used is UCI dataset (i.e. 216 messages from 747 spam messages are used)
2. View each messages to understand the meaning and content.
3. Result by grouping the messages into clusterer based on the understanding the content of messages.

.

Table 5.16 shows the number of messages in each clusters by manually tested and Figure 5.18 shows the number of messages using WEKA.

**Table 5. 16:** Number of messages in each cluster manually tested

| Number of Cluster | Category | Number of message |
|---|---|---|
| Cluster 0 | Competition/ Game | 21 messages |
| Cluster 1 | Chatting | 28messages |
| Cluster 2 | Dating | 13messages |
| Cluster 3 | Prize | 73 messages |
| Cluster 4 | Service | 35messages |
| Cluster 5 | Finance | 11messages |
| Cluster 6 | Ringtone | 17messages |
| Cluster 7 | News | 6messages |
| Cluster 8 | Advertisement | 10messages |
| Cluster 9 | Voicemail | 2messages |
| TOTAL | | 216 messages |

From Table 5.16, result shows that cluster 3 (i.e. category of prize) has the highest number of spam messages while cluster 9 (i.e. category of voicemail) has only two spam messages, less than others.

```
Clustered Instances

0       206 ( 95%)
1         2 (  1%)
2         1 (  0%)
3         1 (  0%)
4         1 (  0%)
5         1 (  0%)
6         1 (  0%)
7         1 (  0%)
8         1 (  0%)
9         1 (  0%)
```

**Figure 5. 18:** Number of instances in each cluster using k-Means

There are some differences between the results that manually tested in Table 5.16 with the results from WEKA using k-Means. The differences are discussed below.

- The types of cluster from WEKA are shown in form of number, so user cannot identify what category of spam for each number. For example, cluster 0 can be categorized as prize or service or other categories. Different with manually tested, we identify ourselves the category of spam in each cluster.
- The numbers of spam messages in each cluster are different between using WEKA and manually tested. This happens because our dataset contains messages that in form of text. Although pre-processing has been done but there are several characters of messages that are categorized into one group while the different is other group. So WEKA automatically detects the messages based on

characteristics or symbols available in messages. Different with manually tested, we do clustering the spam messages based on our own understanding the content of messages.

- The numbers of spam messages is not accurate in each cluster as cluster 2 until 9 contain only one message while the highest numbers of messages in cluster 0. Cluster 0 may have characteristics that mostly spam messages contain that character, so that it has highest number of spam messages.

From experiment of clustering, we can make the comparison which clusterer algorithm is better in term of time taken to cluster. However, WEKA has limitation in term of number of messages in each cluster where the messages is uneven in each of them as WEKA only suitable to see the performance of algorithm, not to see the results after clustering.

Overall, this experiment gives us understanding on which algorithms have the best performance in term of time and accuracy. Besides, we can know how WEKA is performed for detection and classification. In our research, this experiment gives us knowledge on how the process of detection and classification happens in WEKA besides, helps us to make validation of dataset that we will use for our experiment.

### 5.2.5 EXPERIMENT 5 (TRAINING AND TESTING DATASET FOR DETECTION AND CLASSIFICATION)

This experiment discussed about the process of detection and classification SMS messages using training and testing phase in WEKA. UCI dataset contain 4825 ham and 747 spam messages was used and divided into training phase and testing phase as shown in Table 5.17. Classification is supervised learning that need training and testing phase. Training phase is for algorithm to learn the structure of dataset while testing phase is to build the model after learning. Different with clustering that is unsupervised learning, means it need to find the structure of dataset itself and require only testing phase. The aim of this experiment is to see the results after we divided dataset into training and testing phase manually (i.e. manually chose how many messages in training and testing). As from the previous experiment, dataset used is not divided into both phase as the purpose of each experiment is different. Pre-processing is needed to filter the structure of messages from String to Nominal as several algorithms from WEKA cannot read the dataset.

**Table 5. 17:** Dataset in training and testing

| | Training phase (70%) | Testing phase (30%) | Total messages |
|---|---|---|---|
| Labelled as HAM | 3377 | 1448 | 4825 |
| Labelled as SPAM | 523 | 224 | 747 |
| Total | 3900 | 1672 | 5572 |

All 5572 SMS messages are used both for training and testing phase with the fraction of 7:3 (7 for training and 3 for testing). 70% of both ham and spam messages are used in training phase, as the more the dataset use for training, the better the model would be when it is applied in the testing phase. The other 30% of both ham and spam messages are used in testing phase.

a) Spam detection

The aim of this experiment is to identify the capability and performance of WEKA in detecting messages using different method. Four methods were used namely detection training and testing simultaneously, detection training and testing separately, detection using one message in testing phase and detection using different size of dataset in training phase. Only four methods are used in this experiment because we want to see the ability of WEKA in detecting spam messages using various size of dataset. Naïve Bayes (NB), Support Vector Machine (SVM) and k-Nearest Neighbour (k-NN) are three algorithms used to test their performance in detection process. Only these three algorithms are chosen as their popularity among researchers for detection process. Table 5.18 shows four methods in detection SMS messages.

**Table 5. 18:** Detection method

| Method | Training | Testing | Description | Purpose |
|---|---|---|---|---|
| Detection training and testing simultaneously | 3900 messages labeled with ham and spam. | 1672 unlabeled messages | These two phases of spam detection are run simultaneously. | To demonstrate the performance and output of detection by running both phase simultaneously. |
| Detection training and testing separately | 3900 messages labeled with ham and spam. | 1672 unlabeled messages | These two phases of spam detection are run separately. | To demonstrate the performance and output of detection by running both phase separately. |
| Detection using one message in testing phase | 3900 messages labeled with ham and spam. | 1unlabeled messages | These two phases of spam detection are run separately. As for the testing phase, a few different unlabeled messages are run repeatedly with only one unlabeled message run at one time. | To demonstrate the output of detection when using only one unlabeled messages in testing phase. |
| Detection using different size of dataset in training phase | 100 messages labeled with ham and spam. | 1672 unlabeled messages | These 2 phases of spam detection are run separately. As for the testing phase, 1,672 unlabeled messages are run repeatedly with a different size of database stored during training phase. | To identify the performance of classifiers algorithm using different size of dataset in training phase in term of accuracy and number of messages correctly classified. |
| | 1250 messages labeled with ham and spam. | | | |
| | 2500 messages labeled with ham and spam. | | | |

i. METHOD 1 (DETECTION TRAINING AND TESTING SIMULTANEOUSLY)

This approach run training and testing phase simultaneously using percentage split as the test option for running the dataset. The function of percentage split is that it can divide the dataset into training and testing based on the percentage. In this method, 5572 messages was used (i.e. 4825 ham and 747 spam) and divided 70% of the dataset for training and 30% for testing. Table 5.19 shows the results detecting training and testing simultaneously.

**Table 5. 19:** Results for method 1

| CLASSIFICATION ALGORITHM | Results of prediction | Accuracy | TP | FP | TN | FN | Processing Time |
|---|---|---|---|---|---|---|---|
| NB | 1446 ham; 226 spam | 94.56% | 1383 | 28 | 198 | 63 | 0.91 seconds |
| SVM | 1446 ham; 226 spam | 98.21% | 1444 | 28 | 198 | 2 | 2.48 seconds |
| k-NN | 1446 ham; 226 spam | 94.80% | 1445 | 86 | 140 | 1 | 0.01 seconds |

Table 5.19 shows results after running the training and testing phase simultaneously using percentage split as test mode in WEKA. The results only show the number of messages for testing phase (about 1672 messages). In term of accuracy, SVM is more accurate with the percentage is 98.21%, then goes to k-NN in second place with 94.80% and lastly NB with 94.56%. The k-NN is faster in detecting the messages compared to other with time require for this process is only 0.01 seconds. The k-NN can correctly classify the messages into ham (i.e. TP) about 1445 messages, different only 1 message with SVM. For TN (i.e. correctly classified the messages into spam), both NB and SVM have the same number which is 198 messages. So, it can be said that the performance of all algorithms is good by simultaneously running both raining and testing phase and they give readable output.

ii. METHOD 2 (DETECTION TRAINING AND TESTING SEPARATELY)

This method separately run the training and testing phase; with 3900 messages used for training and 1672 messages used for testing. Similar with the method 1, dataset needs to do preprocessing process. There was a problem when running dataset in testing phase because data used to train model and test set was not compatible. In order to run it, it need to choose "InputMappedClassifier" that was recommended by WEKA as shown in Figure 5.19 and Figure 5.20. Table 5.20 shows results for method two.

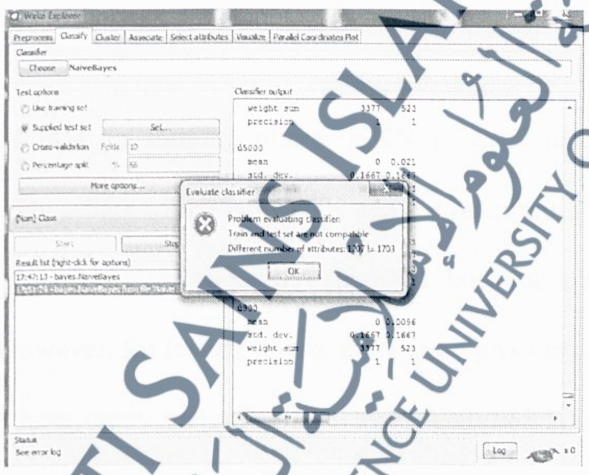


**Figure 5. 19:** Problem in testing phase

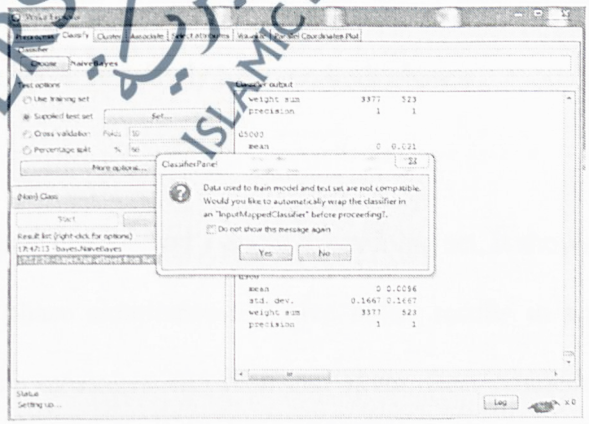


**Figure 5. 20:** Train and test set are not compatible

**Table 5. 20:** Results for training phase

| | | Training Phase | | | | | |
|---|---|---|---|---|---|---|---|
| **Classification Algorithms** | **Results of prediction** | **Accuracy** | **TP** | **TN** | **FP** | **FN** | **Processing Time** |
| NB | 3377 ham; 523 spam | 95.03% | 3245 | 62 | 461 | 132 | 0.64 seconds |
| SVM | 3377 ham; 523 spam | 99.33% | 3377 | 26 | 497 | 0 | 1.54 seconds |
| k-NN | 3377 ham; 523 spam | 99.85% | 3377 | 6 | 517 | 0 | 0.01 seconds |
| | | **Testing Phase** | | | | | |
| NB | 1449 ham; 223 spam | 94.38% | 1399 | 44 | 179 | 50 | - |
| SVM | 1449 ham; 223 spam | 93.78% | 1394 | 49 | 174 | 55 | - |
| k-NN | 1449 ham; 223 spam | 93.48% | 1447 | 107 | 116 | 2 | - |

Table 5.20 indicates that, in training phase, k-NN is good in performance with the accuracy is highest than others which is 99.85% and the processing time require only 0.01 seconds. Although SVM has higher accuracy than NB, but time taken to build the model is slower than NB. However, for testing phase, there is no processing time shown because this phase focus on how many messages able to be detected into ham and spam after learning from training phase. From both phase, it shows that k-NN able to detect more messages into ham (i.e. TP) compare other two. From this experiment, it shows that WEKA also can run dataset separately between both phases.

Making comparison between method 1 and 2 for training set, it shows that detection using training and testing phase simultaneously give better results in term of accuracy and correctly classified the messages into ham and spam. Less connection and relationship when separately run training and testing phase gives difficulties for algorithms to learn

and remember the structure of dataset. That is why results of detection using training and testing phase simultaneously has better than separately detection.

iii. METHOD 3 (DETECTION USING ONE MESSAGE IN TESTING PHASE)

In this method, 3900 labeled messages were used for training and one unlabeled messages for testing. Then this method was repeated using different messages for testing phase. This method also required dataset to do preprocessing to filter the messages. Similar with other method, It need to choose "InputMappedClassifier" that was recommended by WEKA. Table 5.21 and Table 5.22 show results for training and testing phase.

**Table 5. 21:** Results in training phase

| Classification Algorithms | Results of prediction | Accuracy | TP | FP | TN | FN | Processing Time |
|---|---|---|---|---|---|---|---|
| NB | 3377 ham; 523 spam | 95.03% | 3245 | 62 | 461 | 132 | 0.78 seconds |
| SVM | 3377 ham; 523 spam | 99.33% | 3377 | 26 | 497 | 0 | 1.58 seconds |
| k-NN | 3377 ham; 523 spam | 99.85% | 3377 | 6 | 517 | 0 | 0.01 seconds |

**Table 5. 22:** Results in testing phase

| Ham unlabeled messages | Classification Algorithms | Results of prediction | Accuracy | TP | FP | TN | FN | Processing Time |
|---|---|---|---|---|---|---|---|---|
| | NB | 1 ham; 0 spam | 0% | 0 | 0 | 0 | 1 | - |
| | SVM | 1 ham; 0 spam | 100% | 1 | 0 | 0 | 0 | - |
| | k-NN | 1 ham; 0 spam | 100% | 1 | 0 | 0 | 0 | - |
| Spam unlabeled messages | NB | 0 ham; 1 spam | 100% | 0 | 0 | 1 | 0 | - |
| | SVM | 0 ham; 1 spam | 100% | 0 | 0 | 1 | 0 | - |
| | k-NN | 0 ham; 1 spam | 100% | 0 | 0 | 1 | 0 | - |

From the Table 5.21, it can be found that all three classifiers algorithm have the highest accuracy which is 90% and above. k-NN is the best with the accuracy is highest than NB and SVM besides time taken to process the clustering require only 0.01 seconds, faster than others. Besides, the number of messages correctly classified as ham (TP) is same with original dataset, about 3377 ham messages and the number of messages correctly classified into spam (TN) is 517, less 6 messages from original dataset.

This method is tested using only one unlabeled message and the message can be from ham or spam. From the Table 5.22, result presents that when using ham message as unlabeled messages in testing phase, SVM and k-NN manage to correctly identify the message as a ham (TP) but NB detect the message as a spam (FN). When using spam messages as unlabeled messages in testing phase, all three classifiers manage to detect the message as a spam (TN).

From this experiment, it can be said that, by using only one unlabeled messages during testing phase, WEKA still manage to identify whether the message is a spam or ham based on learning process from training phase.

iv. METHOD 4 (DETECTION USING DIFFERENT SIZE OF DATASET IN TRAINING PHASE)

This method uses 1672 unlabeled messages in testing phase and these messages were run repeatedly with different size of dataset in training phase. The purpose of using different size for training phase because we want to see the performance of algorithm when the size of dataset increase. Same problem occur like method two and three in training phase because data used to train model and test set are not compatible and require to choose "InputMappedClassifier" that was recommended by WEKA. Again, dataset in this method also need to do pre-processing process.

**Table 5. 23:** Naïve Bayes

| Classification algorithm | Phases | Results | 100 messages | 1250 messages | 2500 messages |
|---|---|---|---|---|---|
| Naïve Bayes | Training phase | Results of prediction | 50 ham; 50 spam | 1002 ham; 248 spam | 2275 ham; 225 spam |
| | | Accuracy | 99% | 94.96% | 96.08% |
| | | True Positive | 50 | 959 | 2209 |
| | | False Positive | 1 | 20 | 32 |
| | | True Negative | 49 | 228 | 193 |
| | | False Negative | 0 | 43 | 66 |
| | | Processing Time | 0.02 seconds | 0.25 seconds | 0.48 seconds |
| | Testing phase | Results of prediction | 1448 ham; 224 spam | 1448 ham; 224 spam | 1448 ham; 224 spam |
| | | Accuracy | 86.66% | 93.06% | 94.92% |
| | | True Positive | 1248 | 1374 | 1411 |
| | | False Positive | 23 | 42 | 48 |
| | | True Negative | 201 | 182 | 176 |
| | | False Negative | 200 | 74 | 37 |
| | | Processing Time | - | - | - |

From Table 5.23, it was found that if the size of dataset is bigger, time taken to process the detection is also increase in training phase. The accuracy between three size of dataset is different depend on the number of messages in each dataset. However in testing phase,

the accuracy of detection increase as the size of dataset in training phase is bigger. This happens because the bigger size of dataset in training, it give huge change for machine learning to learn the structure of messages so that the accuracy in testing when using higher size is bigger.

**Table 5. 24:** Support Vector Machine

| Classification algorithm | Phase | Results | 100 messages | 1250 messages | 2500 messages |
|---|---|---|---|---|---|
| SVM | Training phase | Results of prediction | 50 ham; 50 spam | 1002 ham; 248 spam | 2275 ham; 225 spam |
| | | Accuracy | 100% | 100% | 99.72% |
| | | True Positive | 50 | 1002 | 2275 |
| | | False Positive | 0 | 3 | 7 |
| | | True Negative | 50 | 245 | 218 |
| | | False Negative | 0 | 0 | 0 |
| | | Processing Time | 0.02 seconds | 0.24 seconds | 0.49 seconds |
| | Testing phase | Results of prediction | 1448 ham; 224 spam | 1448 ham; 224 spam | 1448 ham; 224 spam |
| | | Accuracy | 93.18% | 93.42% | 92.76% |
| | | True Positive | 1403 | 1384 | 1386 |
| | | False Positive | 69 | 46 | 59 |
| | | True Negative | 155 | 178 | 165 |
| | | False Negative | 45 | 64 | 62 |
| | | Processing Time | - | - | - |

From Table 5.24, it presents that the time taken to process the detection is increase as we use bigger size of dataset. However the accuracy is different depend on the number of messages in each dataset. In testing phase, the number of accuracy that is correctly classified into ham and spam messages is increase from 100 messages to 1250 but decrease for 2500 messages. The decreasing result happens because when using 2500 messages, there are variations of types messages that will effect the performance during detection process.

**Table 5. 25:** k-Nearest Neighbour

| Classification algorithm | Training phase | Results | 100 messages | 1250 messages | 2500 messages |
|---|---|---|---|---|---|
| k-NN | Training phase | Results of prediction | 50 ham; 50 spam | 1002 ham; 248 spam | 2275 ham; 225 spam |
| | | Accuracy | 100% | 100% | 99.88% |
| | | True Positive | 50 | 1002 | 2275 |
| | | False Positive | 0 | 0 | 3 |
| | | True Negative | 50 | 248 | 222 |
| | | False Negative | 0 | 0 | 0 |
| | | Processing Time | 0 seconds | 0 seconds | 0.01 seconds |
| | Testing phase | Results of prediction | 1448 ham; 224 spam | 1448 ham; 224 spam | 1448 ham; 224 spam |
| | | Accuracy | 88.76% | 91.51 | 91.03 |
| | | True Positive | 1448 | 1437 | 1446 |
| | | False Positive | 185 | 131 | 148 |
| | | True Negative | 39 | 93 | 76 |
| | | False Negative | 0 | 11 | 2 |
| | | Processing Time | - | - | - |

From Table 5.25, result shows that the time taken to process the detection is increase as we use bigger size of dataset. However the accuracy is different depend on the number of messages in each dataset. In testing phase, the number of accuracy that is correctly classified into ham and spam messages is increase from 100 messages to 1250 but decrease for 2500 messages. The decreasing result happens because when using 2500 messages, there are variations of types messages that will effect the performance during detection process.

From Table 5.23 until Table 5.25, it can be found that SVM has higher accuracy in detecting messages into ham and spam using all three different size of dataset for both phases except for dataset contains 2500 messages, k-NN is highest in training phase with different between SVM is about 0.16%. However, k-NN is faster than NB and SVM in

term of time taken to detect the messages for all three different dataset. Overall, this experiment can be suggested that increasing size of dataset require more time for algorithms to learn the dataset. However, it gives benefit in testing phase where it can give huge opportunity to produce accurate results in testing phase.

b) Spam classification

Classification was conducted to cluster spam messages into categories of spam. It is supervised machine learning and only need testing phase. 224 spam messages were clustered using Cobweb, k-Means and Hierarchical in WEKA. Table 5.26 shows results in clustering spam messages.

**Table 5. 26:** Results in clustering

| **Clustering Algorithms** | **Results of prediction** | **Processing Time** |
|---|---|---|
| K-Means | Clustered Instances<br>Cluster 0 : 124 (55%)<br>Cluster 1 : 1 (0%)<br>Cluster 2 : 9 (4%)<br>Cluster 3 : 19 (8%)<br>Cluster 4 : 5 (2%)<br>Cluster 5 : 46 (21%)<br>Cluster 6 : 2 (1%)<br>Cluster 7 : 6 (3%)<br>Cluster 8 : 10 (4%)<br>Cluster 9 : 2 (1%) | 0.6 seconds |
| Hierarchical | Clustered Instances<br>Cluster 0 : 212 (95%)<br>Cluster 1 : 1 (0%)<br>Cluster 2 : 1 (0%)<br>Cluster 3 : 2 (1%)<br>Cluster 4 : 3 (1%)<br>Cluster 5 : 1 (0%)<br>Cluster 6 : 1 (0%)<br>Cluster 7 : 1 (0%)<br>Cluster 8 : 1 (0%)<br>Cluster 9 : 1 (0%) | 0.18 seconds |
| Cobweb | Clustered Instances<br>Cluster 0: 224 (100%) | 0.92 seconds |

From Table 5.26, k-Means algorithm is the best suited to cluster 224 spam messages into 10 groups in 0.6 second. Hierarchical is in second place with time taken is 0.12 slower than k-Means while cobweb require much time. Hierarchical and k-Means shows the output for all 10 clusters. However cobweb only show the messages into one cluster only. When doing classification process in WEKA, we need to define the number of cluster in k-Means and hierarchical but different with cobweb that no need for number of cluster. That is why cobweb only show one cluster. However, it depends to the dataset used. If we have bigger size of dataset or different format of dataset, cobweb will show result in several cluster. Besides, the function of cobweb that yields a clustering dendrogram called classification tree cause it automatically cluster the messages depend on the tree form.

All these three algorithms have the highest number of spam messages in cluster 0 because WEKA identify and cluster the messages based on the characteristics or structure of a messages. Each cluster represent different characteristic of messages in dataset. So majority of spam messages in dataset used contains the character (such as terms) that represent from cluster 0. However, it depend on the function of algorithms itself on how they cluster the dataset.

From this experiment, it can be say that WEKA can be used to identify which algorithms is the best in term of time processing. However, WEKA has the limitation in giving accurate result in each cluster besides it not shows each cluster represent which category of messages such as prize or service. This experiment can help in our future research by

making comparison between results of clustering using WEKA with our proposed algorithm.

Overall from experiment 5, it helps us understand how WEKA will output the result of detection using various method and size of dataset. We can see the capability and usability of algorithms in detecting SMS messages and how their performances are effected by different number of messages in training and testing phase.

## 5.3 DISCUSSION PROOF OF CONCEPT

Five experiments were carried out in the proof of concept to understand the process of detection and classification using WEKA. Experiment 1 was conducted to understand the formula used to present output after detection process and how the results are calculated. Correctly classified, incorrectly classified, accuracy, error rate, kappa, mean absolute error, root mean squared root, root relative squared error, relative absolute error, TP rate (True Positive), FP rate (False Positive), precision, recall, F-measure ,ROC (Receiver Operating Characteristics) and confusion matrix are examples of output for classification. However, several of them such as correctly classified, TP rate and FP rate can be used for clustering. By understanding how the result is calculated and steps required calculating, it helps in explaining the results. Each classifiers algorithm will show different results using same formula and calculation. In addition, the performance of each algorithm can be compared based on time taken to build the model, accuracy and correctly classified.

The purpose of doing experiment 2 is identify the capability of WEKA in detecting different format of dataset (i.e. label and unlabeled dataset) in testing phase and view the performance of classifiers algorithm between both dataset. For **label dataset**, NB and k-NN are good performance in term of accuracy and time taken to build the model for all three set options. However, for **unlabeled dataset**, the output shows all results in zero. This happens because WEKA could not identify how many messages from ham and spam as we removed the label in each message. So, we can conclude that unlabeled dataset only can be used for testing phase.

The aim of experiment 3 is to understand the process of classification (or known as clustering) using WEKA, what the output that is presented after the process. Three types of clusterer algorithms were used namely k-Means, Hierarchical and Cobweb. k-Means is good performance in clustering for both cluster mode (i.e. Classes to cluster evaluation and use training set) with time taken to build the model is faster than other and has highest accuracy. However, the limitation in this process is the number of messages in each cluster is not accurate and balance as clustering process in WEKA to test the performance of clusterer and not for results after clustering.

Experiment 4 was conducted to test UCI dataset for detection and classification to identify which algorithms has the best performance. In detection, k-NN is the best classifiers and good performance compared to other classifiers algorithms. Besides, the number of messages that are correctly classified into ham and spam (i.e. TP and TN) also accurate as compared to others. However, the performance and results for each classifier

may be changed using different dataset because each classifiers algorithm has their own theory that suitable for different dataset. In the classification process, k-Means is good performance with time require to build the model is 0 seconds, faster than hierarchical and cobweb. In addition, a test is done manually to identify the number of spam messages in each cluster by view and understand the meaning and contents of messages and the results was compared with WEKA. The numbers of spam messages in each cluster using k-Means is not same and accurate with the number manually tested due to the limitation of WEKA itself.

Experiment 5 was conducted to test the training and testing phase in detection and classification process. As for the detection process, there are two levels that applied: training and testing level. These two different levels are actually reflected the supervised machine learning characteristics. During training, a set of 3,377 ham and 523 spam labeled messages were running through a few different classifiers separately. Then, for testing, based on stored correlation attributes during the training level, a set of unlabeled messages were running through those classifiers. In spam detection process, there were four methods to test unlabeled messages in testing phase. In method one, 3900 messages labeled with ham and spam were run simultaneously using 1672 unlabeled messages and for method two, using same messages, they were run separately. For both experiment, results show that k-NN is good performance in term of time processing. In method three,3900 labeled messages were used in training phase and one unlabeled messages for testing phase. The unlabeled messages may be a ham or spam messages. Again, results show that k-NN is good performance for time taken and accuracy to detect spam and ham

messages. In method four, 1672 unlabeled messages in testing phase were run repeatly with different size of dataset in training phase. This method is to demonstrate the link between spam library (developed during training) with unlabeled messages run in testing phase, also to find the influence degree of number of messages used in training with the result of spam detection, in term of accuracy rate. As the number of messages in training phase increase and larger, the accuracy of classifiers for detecting is lower, same with the performance of time, it require more time to process the dataset. The results for accuracy and time processing only show for training phase as it want to train the dataset using classifiers and see the performance of classifiers, different with testing phase that only shows the results for detecting messages into ham and spam. For classification, k-Means is fasters than other clusterers algorithms. However, the numbers of messages in each cluster are not accurate due to several factors such as different function and role of algorithms and also different structure of dataset used.

From the proof of concept experiment, it helps in understanding how to use WEKA using various methods and structure of dataset. Overall, we can suggest that, the main function of WEKA is to test the performance between classifiers algorithms and clusterer algorithms, besides to test the validity of dataset used.

## 5.4 PROOF OF PERFORMANCE

The aim of this experiment is to investigate the performance of the proposed algorithms for detecting and classification spam messages. This subsection is divided into 2 main sections; namely detection phase and classification phase. Detection phase is the process to detect the SMS messages into ham and spam while classification phase is the process to categorize the spam messages into defined group.

### 5.4.1 DETECTION PHASE

Detection is the first phase of IMSM. Several experiments have been conducted and results are reported and discussed in five parts. Part 1 presents the detection of messages using AIS algorithms (i.e. Danger Theory and Negative Selection) while part 2 discusses the results for dataset validation. Then, part 3 and 4 reported the performance detection of the proposed algorithm using raw and clean dataset, and finally part 5 compare the results for raw and clean dataset.

i. Part 1: Detection using AIS Algorithm

This part discusses the detection process using two types of algorithms which are Negative Selection and Danger Theory. In first phase of IMSM, both algorithms are used for detecting SMS messages into ham and spam. These two algorithms are tested using WEKA using three different dataset (i.e. DIT, GIT and UCI) and results shown in Table 5.27.

**Table 5. 27:** Comparison of Results between Negative Selection algorithm with Danger Theory algorithm

| DIT | | | | | | |
|---|---|---|---|---|---|---|
| **Test Option** | **Classifier** | **Correctly Classified** | **Incorrectly Classified** | **Roc Area** | **Accuracy** | **Time Taken To Build Model (S)** |
| Use training set | Negative Selection | **94.60%** | 5.40% | 0.937 | 0.946 | 0.04 |
| | Danger Theory | 45.75% | 54.25% | 0.518 | 0.458 | 0.17 |
| **GIT** | | | | | | |
| **Test Option** | **Classifier** | **Correctly Classified** | **Incorrectly Classified** | **Roc Area** | **Accuracy** | **Time Taken To Build Model (S)** |
| Use training set | Negative Selection | **99.64%** | 0.36% | 0.989 | 0.996 | 0.05 |
| | Danger Theory | 78.49% | 21.51% | 0.743 | 0.8332 | 0.09 |
| **UCI** | | | | | | |
| **Test Option** | **Classifier** | **Correctly Classified** | **Incorrectly Classified** | **Roc Area** | **Accuracy** | **Time Taken To Build Model (S)** |
| Use training set | Negative Selection | **100%** | 0% | 1 | 1 | 0.02 |
| | Danger Theory | 82.95% | 17.05% | 0.504 | 0.827 | 0.7 |

From the Table 5.27, it can be suggested that the Negative Selection has obtained higher values for all measurement (i.e. higher correctly classified and less error, ROC area and accuracy) for three dataset as compared to the Danger Theory. Also, the time taken by WEKA to build this algorithm was shorter than the Danger Theory which indicates that the Negative Selection model learns quite fast and is evaluated even faster than the other model. Apart from that, from the results of comparison Table 5.27, it is shown that the Negative Selection algorithm is much better than the Danger Theory in performance where it shows all of its values of correctly classified are higher and lower incorrectly classified than the values in the other theory.

Having obtained these initial results and to improve the performance of Danger Theory for detection, we introduce three features; length, special characters and keyword for detection. In length of message, we assume messages that are more than 100 in length could potentially be classified as spam. SMS spam normally use standard and formal language to attract users and it give understanding to user about the contents of messages. Secondly, in terms of special characters, spammers prefer to use numbers or digits like phone number or code to attract users. Lastly in keywords of spam and ham, there are some familiar terms that are used by spammer to differentiate between spam and ham messages. These three proposed features were stipulated into five different algorithms and then, tested with three different datasets for their ability to detect spam. The used datasets named UCI Machine Learning (UCI), British English SMS Corpora (BEC) and Dublin Institute of Technology (DIT) respectively.

ii. Part 2: Dataset Validation

The purpose of doing dataset validation is to validate the dataset whether it can be used and read for testing several experiments. Three different dataset used in this part; namely DIT, BEC and UCI. Table 5.28 presents results of experiment conduct for validating datasets.

**Table 5. 28:** Results in WEKA using four classifiers

| Dataset | Classifier | Results in WEKA | | | | | |
|---|---|---|---|---|---|---|---|
| | | Correctly Classified | Incorrectly Classified | TP | FN | TN | FP |
| DIT Ham = 0 Spam = 1353 | NB | 1353=100 % | 0 = 0 % | 0 | 0 | 1353 | 0 |
| | SVM | 1353=100 % | 0 = 0 % | 0 | 0 | 1353 | 0 |
| | k-NN | 1353=100 % | 0 = 0 % | 0 | 0 | 1353 | 0 |
| | DT | 1353=100 % | 0 = 0 % | 0 | 0 | 1353 | 0 |
| BEC Ham = 450 Spam =425 | NB | 804 = 91.89 % | 71 = 8.11% | 413 | 34 | 391 | 37 |
| | SVM | 868 =99.2% | 7 = 0.8% | 450 | 7 | 418 | 0 |
| | k-NN | 872 = 99.66% | 3 = 0.34% | 450 | 3 | 422 | 0 |
| | DT | 750 = 85.71% | 125 = 14.29% | 434 | 109 | 316 | 16 |
| UCI Ham=4825 Spam=747 | NB | 5306= 95.23% | 266 = 4.77% | 4646 | 87 | 660 | 179 |
| | SVM | 5537=99.37% | 35 = 0.63% | 4825 | 35 | 712 | 0 |
| | k-NN | 5564= 99.86% | 8 = 0.14% | 4825 | 8 | 739 | 0 |
| | DT | 5416= 97.20% | 156 = 2.80% | 4807 | 138 | 609 | 18 |

From the Table 5.28, it can be found that the k-NN classifier produced the best detection rate where it managed to detect the highest number of spam and ham messages and is it also found that all classifiers managed to detect spam messages (TN) of the DIT dataset. As majority of chosen classifiers perform detection of ham and spam messages in a 'good' manner, it can be suggested that the chosen datasets are appropriate to be used for testing the proposed algorithms.

iii. Part 3: Algorithm 1-5 using raw dataset

This part presents the results of detection using three proposed features. These three proposed features were stipulated into five different algorithms and then, tested with three different datasets for their ability to detect spam. Five algorithms were created depends on the use of these three feature either using only one feature in an algorithm or by combining all the features. These features are included as the improvement for the

performance of Danger Theory in detecting spam and ham messages. Table 5.29 until Table 5.38 show results and explanation for detection process using all five algorithms and dataset used was not requiring any pre-processing or cleaning process to remove punctuation marks or symbol because these may help in process of detection. The summary result of detecting spam messages for three dataset is shown in Table 5.29.

**Table 5. 29:** Results for Algorithm 1 using raw dataset

| Detection features | Phases | Process | DIT | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | **1353** | **450** | **425** | **4825** | **747** |
| Keywords | PHASE 1 (spam keywords) | Spam | - | 1291 | 88 | 407 | 1192 | 724 |
| | | ToHam | | 62 | 362 | 18 | 3633 | 23 |
| | PHASE 2 (ham keywords) | Spam-Spam | - | 1291 | 88 | 407 | 1192 | 724 |
| | | ToHam-Ham | | 59 | 357 | 16 | 3501 | 20 |
| | | ToHam-Spam | | 3 | 5 | 2 | 132 | 3 |
| | RESULTS | TP | - | | 357 | | 3501 | |
| | | FN | - | | 93 | | 1324 | |
| | | TN | 1294 | | 409 | | 727 | |
| | | FP | 59 | | 16 | | 20 | |
| | | **HAM Messages** | - | | **357 @ 79.33%** | | **3369 @ 72.56%** | |
| | | **SPAM Messages** | **1294 @ 95.64%** | | **409 @ 96.24%** | | **727 @ 97.32%** | |
| | | **Accuracy** | **95.64%** | | **87.54%** | | **75.88%** | |

Table 5.29 shows results of detection for Algorithm 1. This algorithm uses only one feature which is 'keywords' and it requires two phases; phase 1 is for spam keywords and phase 2 for ham keywords. The explanation for process in each phase is discussed in Table 5.30. From Table 5.30, it can be stated that Algorithm 1 manages to detect 95.64% of spam messages for DIT dataset while for the BEC dataset, it can detect 357 ham messages from 450 messages and 409 spam messages from 425 messages. For UCI

dataset, about 3369 ham messages can be detected from 4825 messages and 727 spam messages from 747 messages. Algorithm 1 manages to detect accurate number of spam messages for all three dataset with the average percentage is 96.4% compare to ham messages with the average below 80%. Spam keywords are used in first phase of the algorithms, so all messages will be scanned in first phase then continue to the second phase. So several ham messages may incorrectly classified as spam messages (i.e. FN) lead the accuracy for ham detection is lower using this algorithm.

**Table 5. 30:** Process in each phase for Algorithm 1

| Phase | Process | Explanation |
|---|---|---|
| PHASE 1 | Spam | SMS messages contain keywords of spam are classified as spam. |
| | ToHam | SMS messages do not contain any keywords of spam are classified as ToHamand continue to the second phase. |
| PHASE 2 | Spam-Spam | No process occur as they are managed classifying as spam messages in phase one. |
| | ToHam-Ham | Messages that are classified as ToHam in phase 1 will go to second detection using ham keywords. If ToHam messages match and contain ham keywords, they are classified as ham messages and the process end. |
| | ToHam-Spam | Messages that classified as ToHam in phase 1 will go to second detection using ham keywords. If ToHam messages do not contain any ham keywords, they are classified as spam messages and the process end. |

**Table 5. 31:** Results for Algorithm 2 using raw dataset

| Detection features | Phases | Process | DIT | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | **1353** | **450** | **425** | **4825** | **747** |
| Length and Keywords | PHASE 1 (length and spam keywords) | Spam | - | 1108 | 36 | 367 | 486 | 660 |
| | | ToHam | - | 245 | 414 | 58 | 4339 | 87 |
| | PHASE 2 (ham keywords) | Spam-Spam | - | 1108 | 36 | 367 | 486 | 660 |
| | | ToHam-Ham | - | 218 | 409 | 46 | 4178 | 72 |
| | | ToHam-Spam | - | 27 | 5 | 12 | 161 | 15 |
| | RESULTS | TP | - | | 409 | | 4178 | |
| | | FN | - | | 41 | | 647 | |
| | | TN | 1135 | | 379 | | 675 | |
| | | FP | 218 | | 46 | | 72 | |
| | | **HAM Messages** | - | | **409 @ 90.89%** | | **4178 @ 86.59%** | |
| | | **SPAM Messages** | **1135 @ 83.89%** | | **379 @ 89.18%** | | **675 @ 90.36%** | |
| | | **Accuracy** | **83.89%** | | **90.06%** | | **87.10%** | |

Algorithm 2 uses the combination of two features and the features are 'message length' and 'keywords'. Two phases involved as shown in the Table 5.31 and the explanation for each phase as shown in Table 5.32. Phase 1 uses the combination of message length with spam keywords while phase 2 uses only ham keywords. It can be found that Algorithm 2 manages to detect 83.89% of spam messages for DIT dataset while 89.18% spam messages in BEC dataset. Meanwhile for UCI dataset, only 675 spam messages managed to be detected out of 747 messages. The different number of spam messages detection between these three dataset due to different size of each dataset. For ham messages detection, about 90.89% of ham messages manage to be detected for BEC dataset and 86.59% for UCI dataset.

**Table 5. 32:** Process in each phase for Algorithm 2

| Phase | Process | Explanation |
|---|---|---|
| PHASE 1 | Spam | SMS messages contain length more than 100 words and keyword of spam are classified as spam. |
| | ToHam | SMS messages do not match the features used are classified as ToHam and continue to the second phase. |
| PHASE 2 | Spam-Spam | No process occur as they are managed classifying as spam messages in phase one. |
| | ToHam-Ham | Messages that are classified as ToHam in phase 1 will go to second detection using ham keywords. If ToHam messages contain ham keywords, they are classified as ham messages and the process end. |
| | ToHam-Spam | Messages that classified as ToHam in phase 1 will go to second detection using ham keywords. If ToHam messages do not contains any ham keywords, they are classified as spam messages and the process end. |

**Table 5. 33:** Results for Algorithm 3 using raw dataset

| Detection features | Phases | Process | DIT | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | 1353 | **450** | **425** | **4825** | **747** |
| Characters and Keywords | PHASE 1 (Characters and spam keywords) | Spam | - | 1215 | 20 | 383 | 213 | 696 |
| | | ToHam | - | 138 | 430 | 42 | 4612 | 51 |
| | PHASE 2 (ham keywords) | Spam-Spam | | 1215 | 20 | 383 | 213 | 696 |
| | | ToHam-Ham | - | 132 | 425 | 38 | 4454 | 46 |
| | | ToHam-Spam | - | 6 | 5 | 4 | 158 | 5 |
| | RESULTS | TP | - | | 425 | | 4454 | |
| | | FN | - | | 25 | | 371 | |
| | | TN | 1221 | | 387 | | 701 | |
| | | FP | 132 | | 38 | | 46 | |
| | | **HAM Messages** | - | | **425 @ 94.44%** | | **4454 @ 92.31%** | |
| | | **SPAM Messages** | **1221@ 90.24%** | | **387 @ 91.06%** | | **701 @ 93.84%** | |
| | | **Accuracy** | **90.24%** | | **92.80%** | | **92.52%** | |

Algorithm 3 uses the feature of 'special characters' and 'keywords' in detection process. Phase 1 uses the combination of special characters and spam keywords while phase 2 uses only ham keywords and the explanation for each process is discussed in Table 5.34. From Table 5.33 it shows that this algorithm can detect 90.24% of spam messages from 1353 messages for DIT dataset. For BEC dataset, it manage to identify about 94.44%of ham messages out of 450 messages and 91.06% of spam messages out of 425 messages while for UCI dataset, about 371 ham messages that are incorrectly classified as spam and 46 spam messages that are incorrectly classified as ham messages. So, it can be suggested that using two features can give better performance in detecting spam and ham messages.

**Table 5. 34:** Process in each phase for Algorithm 3

| Phase | Process | Explanation |
|---|---|---|
| PHASE 1 | Spam | SMS messages contain special characters and keyword of spam are classified as spam. |
| | ToHam | SMS messages do not match the features used are classified as ToHam and continue to second phase. |
| PHASE 2 | Spam-Spam | No process occur as they are managed classifying as spam messages in phase one. |
| | ToHam-Ham | Messages that classified as ToHam in phase 1 will go to second detection using ham keywords. If ToHam messages contain ham keywords, they are classified as ham messages and the process end. |
| | ToHam-Spam | Messages that classified as ToHam in phase 1 will go to second detection using ham keywords. If ToHam messages do not contains any ham keywords, they are classified as spam messages and the process end. |

**Table 5. 35:** Results for Algorithm 4 using raw dataset

| Detection features | Phases | Process | DIT | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | **1353** | **450** | **425** | **4825** | **747** |
| Length, Character and Keywords | PHASE 1 (length) | Ham | - | 225 | 314 | 49 | 3729 | 76 |
| | | ToSpam | - | 1128 | 136 | 376 | 1096 | 671 |
| | PHASE 2 (characters and spam keywords) | Ham-Ham | - | 225 | 314 | 49 | 3729 | 76 |
| | | ToSpam-Spam | - | 1064 | 10 | 349 | 141 | 640 |
| | | ToSpam-ToHam | - | 64 | 126 | 27 | 955 | 31 |
| | PHASE 3 (ham keywords) | Ham-Ham-Ham | - | 225 | 314 | 49 | 3729 | 76 |
| | | ToSpam-Spam-Spam | - | 1063 | 10 | 348 | 141 | 638 |
| | | ToSpam-ToHam-Ham | - | 65 | 126 | 28 | 955 | 33 |
| | | ToSpam-ToHam-Spam | - | 0 | 0 | 0 | 0 | 0 |
| | RESULTS | TP | - | | 440 | | 4684 | |
| | | FN | - | | 10 | | 141 | |
| | | TN | 1063 | | 348 | | 638 | |
| | | FP | 290 | | 77 | | 109 | |
| | | **HAM Messages** | - | | **440 @ 97.78%** | | **4684 @ 97.08%** | |
| | | **SPAM Messages** | **1063@ 78.57%** | | **348 @ 81.88%** | | **638 @ 85.41%** | |
| | | **Accuracy** | **78.57%** | | **90.06%** | | **95.51%** | |

Table 5.35 shows result of detection process using algorithm 4. This algorithm uses all three features which are 'message length', 'special characters' and keywords'. Three phases involve; phase 1 uses length messages, phase 2 uses the combination of special characters and spam keywords and phase 3 uses ham keywords as explained in Table 5.36. Algorithm 4 manages to detect about 78.57% of spam messages for DIT dataset and 81.88% for BEC dataset. Meanwhile in UCI dataset, it manages to detect 85.41% of spam messages. For ham messages, BEC dataset manage to detect 440 messages out of 450 while UCI dataset manage to detect 4684 out of 4825 messages. Algorithm 4 can detect highest number of ham messages compare to spam messages because the first feature used in phase one is length of message. If the length of message less than 100 words, it classified as ham messages. However, this algorithm gives good results in detection phase.

**Table 5. 36:** Process in each phase for Algorithm 4

| Phase | Process | Explanation |
|---|---|---|
| PHASE 1 | Ham | SMS messages do not contains length more than 100 words are classified as ham. |
| | ToSpam | SMS messages contains length more than 100 words are classified as ToSpam andcontinue to the second phase. |
| PHASE 2 | Ham-Ham | No process occur as they are managed classifying as ham messages in phase one. |
| | ToSpam-Spam | SMS Messages that are classified as ToSpam in phase I will go to second detection using special character and spam keywords. If ToSpam messages contain all features used, they are classified as spam messages and the process end. |
| | ToSpam-ToHam | Messages that classified as ToSpam in phase I will go to second detection using character and spam keywords. If ToSpam messages do not containall the features used, they are classified as ToHam messages and continue to third phase of detection. |
| PHASE 3 | Ham-Ham-Ham | No process occur as they are managed classifying as ham messages in phase one and two. |
| | ToSpam-Spam-Spam | No process occur as they are managed classifying as spam messages in phase two. |
| | ToSpam-ToHam-Ham | SMS messages that are classified as ToHam messages from phase two will go to the last phase of detection using ham keywords. If they contain the keyword, they are classified as ham messages and the process end. |
| | ToSpam-ToHam-Spam | SMS messages that are classified as ToHam messages from phase two will go to the last phase of detection process using ham keywords. If they do not match any of ham keywords, they are classified as spam messages and the process end. |

**Table 5. 37:** Results for Algorithm 5 using raw dataset

| Detection features | Phases | Process | DIT dataset | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | **1353** | **450** | **425** | **4825** | **747** |
| Length, Characters and Keywords (2x) | PHASE 1 (length) | ToHam1 | - | 225 | 314 | 49 | 3729 | 76 |
| | | ToSpam | - | 1128 | 136 | 376 | 1096 | 671 |
| | PHASE 2 (characters and spam keywords) | ToHam1-Toham1 | - | 225 | 314 | 49 | 3729 | 76 |
| | | ToSpam-Spam | - | 1064 | 10 | 349 | 141 | 640 |
| | | ToSpam-ToHam2 | - | 64 | 126 | 27 | 955 | 31 |
| | PHASE 3 (spam keywords) | ToHam1-Toham1-Ham | - | 111 | 314 | 20 | 3715 | 27 |
| | | ToSpam-Spam-Spam | - | 1064 | 10 | 349 | 141 | 640 |
| | | ToSpam-ToHam2-ToHam2 | - | 64 | 126 | 27 | 955 | 31 |
| | | ToHam1-Toham1-Spam | - | 114 | 0 | 29 | 14 | 49 |
| | PHASE 4 (ham keywords) | ToSpam-Spam-Spam-Spam | - | 1064 | 10 | 349 | 141 | 640 |
| | | Tospam-ToHam2-ToHam2-Ham | - | 65 | 126 | 28 | 955 | 33 |
| | | ToSpam-ToHam2-ToHam2-Spam | - | 0 | 0 | 0 | 0 | 0 |
| | | ToHam1-ToHam1-Spam-Spam | - | 113 | 0 | 28 | 14 | 47 |
| | | ToHam1-ToHam1-Ham-Ham | - | 111 | 314 | 20 | 3715 | 27 |
| | RESULTS | TP | - | | 440 | | 4670 | |
| | | FN | - | | 10 | | 155 | |
| | | TN | 1177 | | 377 | | 687 | |
| | | FP | 176 | | 48 | | 60 | |
| | | **HAM Messages** | - | | **440@ 97.78%** | | **4670@ 96.79%** | |
| | | **SPAM Messages** | **1177@ 86.99%** | | **377 @ 88.71%** | | **687@ 91.79%** | |
| | | **Accuracy** | **86.99%** | | **93.37%** | | **96.14%** | |

The process of detection in algorithm 5 is same with algorithm 4 except it require 4 phases and for the third phase, spam keywords are used again. Table 5.37 shows results for each phase in Algorithm 5 while Table 5.38 shows the explanation in each phase. By using algorithm 5, it manages to detect 1177 spam messages from 1353 messages for DIT dataset. In addition, about 97.78% ham messages can be detected from 450 messages and 88.71% spam messages from 425 messages. For UCI dataset, the different of ham messages with actual messages is 155 and 60 messages for spam. From this result, it can be suggest that the combination of three features with repeating one of the features in Algorithm 5 still can give better performance and readable result.

**Table 5. 38:** Process in each phase for Algorithm 5

| Phase | Process | Explanation |
|---|---|---|
| PHASE 1 | ToHam1 | SMS messages do not contain length more than 100 words are classified as ToHam1 and continue to third phase of detection. |
| | ToSpam | SMS Messages contains length more than 100 words are classified as ToSpam and continue to second phase of detection. |
| PHASE 2 | ToHam1-ToHam1 | SMS messages that are classified as ToHam1 in phase one will go to third phase of detection. No process occur in this phase for ToHam1. |
| | ToSpam-Spam | Messages that are classified as ToSpam in phase one will go to second detection using special character and spam keywords. If ToSpam messages contain all features used, they are classified as spam messages and the process end. |
| | ToSpam-ToHam2 | Messages that are classified as ToSpam in phase one will go to second detection using character and spam keywords. If ToSpam messages do not contains all the features used, they are classified as ToHam2 messages and continue to fourth phase of detection. |
| PHASE 3 | ToHam1-ToHam1-Ham | SMS messages that classified as ToHam1 from phase one and phase two will go through third phase of detection by scanning using only two keywords of spam messages. If ToHam1 messages do not match with the spam keywords available in the library, they are classified as HAM messages and the process ends. |
| | ToSpam-Spam-Spam | No process occurs as they are managed classified as spam messages in phase two. |
| | ToSpam-ToHam2-ToHam2 | SMS messages that are classified as ToHam2 from phase two will go to fourth phase of detection. No process occur in this phase for ToHam2. |
| | ToHam1-ToHam1-Spam | SMS messages that are classified as ToHam1 from phase one and phase two will go through third phase of detection by scanning using only two keywords of spam messages. If ToHam1 messages match with the spam keywords available in the library, they are classified as SPAM messages and the process ends. |
| PHASE 4 | ToSpam-Spam-Spam-Spam | No process occurs as they are managed classified as spam messages in phase two. |
| | ToSpam-ToHam2-ToHam2-Ham | If ToHam2 messages match with the keywords of ham messages available in the library, the messages are classified as HAM messages and the process ends. |
| | ToSpam-ToHam2-ToHam2-Spam | If ToHam2 messages do not match with the keywords of ham messages available in the library, they are classified as SPAM messages and the process ends. |
| | ToHam1-ToHam1-Spam-Spam | No process occurs as they are managed classified as spam messages in phase three. |
| | ToHam1-ToHam1-Ham-Ham | No process occur as they are managed classified as ham messages in phase three. |

**Table 5. 39:** Summary result for raw dataset using five algorithms

| Dataset | Algorithm | Detection Feature(s) | Simulation Results | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Correctly Classified | Incorrectly Classified | TP | FN | TN | FP |
| DIT | 1 | Keywords | 1294 = 95.64% | 59 = 4.36% | 0 | 0 | 1294 | 59 |
| | 2 | Length and Keywords | 1135= 83.89% | 218 = 16.11% | 0 | 0 | 1135 | 218 |
| | 3 | Characters and keywords | 1221 = 90.24 % | 132= 9.76% | 0 | 0 | 1221 | 132 |
| | 4 | Length, Character and Keywords | 1063 = 78.57% | 290 = 21.43% | 0 | 0 | 1063 | 290 |
| | 5 | Length, Character and Keywords (2x) | 1177 = 87.00% | 176 = 13.00 % | 0 | 0 | 1177 | 176 |
| BEC | 1 | Keywords | 766 = 87.54% | 109 = 12.46 % | 357 | 93 | 409 | 16 |
| | 2 | Length and Keywords | 788= 90.06% | 87 = 9.94% | 409 | 41 | 379 | 46 |
| | 3 | Characters and keywords | 812 = 92.8 % | 63 = 7.2% | 425 | 25 | 387 | 38 |
| | 4 | Length, Character and Keywords | 788 = 90.06% | 87 = 9.94% | 440 | 10 | 348 | 77 |
| | 5 | Length, Character and Keywords (2x) | 817 = 93.37% | 58 = 6.63% | 440 | 10 | 377 | 48 |
| UCI | 1 | Keywords | 4228 = 75.88 % | 1344 = 24.12% | 3501 | 1324 | 727 | 20 |
| | 2 | Length and Keywords | 4853= 87.10% | 719 = 12.90% | 4178 | 647 | 675 | 72 |
| | 3 | Characters and keywords | 5155 = 92.52% | 417 = 7.48% | 4454 | 371 | 701 | 46 |
| | 4 | Length, Character and Keywords | 5322 = 95.51% | 250 = 4.49% | 4684 | 141 | 638 | 109 |
| | 5 | Length, Character and Keywords (2x) | 5357 = 96.14% | 215 = 3.86% | 4670 | 155 | 687 | 60 |

Overall Results of detection using five algorithms are shown in Table 5.39. From the Table 5.39, it can be stated that Algorithm 1 manages to detect highest spam messages (TN) as compared to other algorithms for all three dataset. Here we can suggest that the keywords of spam messages have the highest priority in detecting phase for spam messages as it gives better performance for Algorithm 1 compare to others. For correctly classified messages into ham and spam, each dataset has different algorithm that give highest accuracy in detection the messages. In DIT dataset, algorithm 1 is more accurate in correctly classified messages than others with percentage is 95.64%. Algorithm 3 is more accurate for dataset of BEC with 92.8% while Algorithm 5 produced more accurate detection for datasets of UCI with 96.14%. The different performance of each algorithm due to different number of messages available in each dataset besides the different structure and contains of messages that might affect the performance of detection.

iv. Part 4: Algorithm 1 -5 using clean dataset

Clean dataset (i.e. all symbol or punctuation marks are removed) is used for this part to test its capability in detecting the spam and ham messages using three features. Table 5.40 until Table 5.44 show the results of detection using clean dataset and Table 5.45 shows the summary of results using clean dataset.

**Table 5. 40:** Results for Algorithm 1 using clean dataset

| Detection features | Phases | Process | DIT | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | **1353** | **450** | **425** | **4825** | **747** |
| Keywords | PHASE 1 (Spam keywords) | Spam | - | 1287 | 98 | 404 | 1214 | 721 |
| | | ToHam | | 66 | 352 | 21 | 3611 | 26 |
| | PHASE 2 (Ham keywords) | Spam-Spam | - | 1287 | 98 | 404 | 1214 | 721 |
| | | ToHam-Ham | - | 63 | 347 | 19 | 3432 | 23 |
| | | ToHam-Spam | - | 3 | 7 | 2 | 179 | 3 |
| | RESULTS | TP | - | | 347 | | 3432 | |
| | | FN | - | | 105 | | 1393 | |
| | | TN | 1290 | | 406 | | 724 | |
| | | FP | 63 | | 19 | | 23 | |
| | | **HAM Messages** | - | | **347 @ 77.11%** | | **3432 @ 71.13%** | |
| | | **SPAM Messages** | **1290 @ 95.34%** | | **406 @ 95.53%** | | **724 @ 96.92%** | |
| | | **Accuracy** | **95.34%** | | **86.06%** | | **74.59%** | |

Table 5.40 shows the result of detection using Algorithm 1 and the explanation of the process in each phase shown in Table 5.30. Result indicates that Algorithm 1 manages to detect highest number of spam messages for all three dataset (i.e. 95.34% for DIT dataset, 95.53% for BEC dataset and 96.92% for UCI dataset). For ham messages, it manages to detect about 347 messages out of 450 messages for BEC dataset and 3432 messages out of 4825 messages in UCI dataset.

**Table 5. 41:** Results for Algorithm 2 using clean dataset

| Detection features | Phases | Process | DIT | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM s | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | **1353** | **450** | **425** | **4825** | **747** |
| Length and Keywords | PHASE 1 (length and spam keywords) | Spam | - | 1084 | 35 | 355 | 478 | 643 |
| | | ToHam | - | 269 | 415 | 70 | 4347 | 104 |
| | PHASE 2 (Ham keywords) | Spam-Spam | - | 1084 | 35 | 355 | 478 | 643 |
| | | ToHam-Ham | - | 240 | 408 | 57 | 4127 | 87 |
| | | ToHam-Spam | - | 29 | 7 | 13 | 220 | 17 |
| | RESULTS | TP | - | | 408 | | 4127 | |
| | | FN | - | | 42 | | 698 | |
| | | TN | 1113 | | 368 | | 660 | |
| | | FP | 240 | | 57 | | 87 | |
| | | **HAM Messages** | - | | **408 @ 90.67%** | | **4127 @ 85.53** | |
| | | **SPAM Messages** | **1113 @ 82.26%** | | **368 @ 86.59%** | | **660 @88.35** | |
| | | **Accuracy** | **82.26%** | | **88.69%** | | **85.91%** | |

From Table 5.41, it can be found that Algorithm 2 manages to detect 1113 spam messages out of 1353 messages for DIT dataset. Only 408 ham messages from 450 messages and 368 spam messages from 425 messages manage to be detected for BEC dataset. Lastly for UCI dataset, the different of ham messages after detection from actual messages is 698 and 87 for spam messages. The process in each phase for Algorithm 2 has been explained in Table 5.32.

**Table 5. 42:** Results for Algorithm 3 using clean dataset

| Detection features | Phases | Process | DIT | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | **1353** | **450** | **425** | **4825** | **747** |
| Characters and Keywords | PHASE 1 (Characters and spam keywords) | Spam | - | 1214 | 22 | 382 | 214 | 695 |
| | | ToHam | - | 139 | 428 | 43 | 4611 | 52 |
| | PHASE 2 (ham keywords) | Spam-Spam | - | 1214 | 22 | 382 | 214 | 695 |
| | | ToHam-Ham | - | 133 | 421 | 39 | 4394 | 47 |
| | | ToHam-Spam | - | 6 | 7 | 4 | 431 | 5 |
| | RESULTS | TP | - | | 421 | | 4394 | |
| | | FN | - | | 29 | | 645 | |
| | | TN | 1220 | | 386 | | 700 | |
| | | FP | 133 | | 39 | | 47 | |
| | | **HAM Messages** | - | | **421 @ 93.56%** | | **4394 @ 91.07%** | |
| | | **SPAM Messages** | **1220 @ 90.17%** | | **386 @ 90.82%** | | **700 @ 93.71%** | |
| | | **Accuracy** | **90.17%** | | **92.23%** | | **91.42%** | |

Table 5.42 present results of detection clean dataset using Algorithm 3 and the explanation of process in each phase shown in Table 5.34. This algorithm manages to detect 90.17% of spam messages from 1353 messages in DIT dataset and in BEC dataset, 93.56% of ham messages successfully detect from 450 messages and 90.82% of spam messages from 425 messages. The number of ham and spam messages that can be detected from UCI dataset also different with actual messages.

**Table 5. 43:** Results for Algorithm 4 using clean dataset

| Detection features | Phases | Process | DIT | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | **1353** | **450** | **425** | **4825** | **747** |
| Length, Characters and Keywords | PHASE 1 (length) | Ham | - | 249 | 337 | 60 | 3809 | 92 |
| | | ToSpam | - | 1104 | 113 | 365 | 1016 | 655 |
| | PHASE 2 (characters and spam keywords) | Ham-Ham | - | 249 | 337 | 60 | 3809 | 92 |
| | | ToSpam-Spam | - | 1045 | 12 | 339 | 139 | 625 |
| | | ToSpam-ToHam | - | 59 | 101 | 26 | 877 | 30 |
| | PHASE 3 (ham keywords) | Ham-Ham-Ham | - | 249 | 337 | 60 | 3809 | 92 |
| | | ToSpam-Spam-Spam | - | 1045 | 12 | 338 | 139 | 623 |
| | | ToSpam-ToHam-Ham | - | 60 | 101 | 27 | 877 | 32 |
| | | ToSpam-ToHam-Spam | - | 1 | 0 | 0 | 0 | 0 |
| | RESULTS | TP | - | | 438 | | 4686 | |
| | | FN | - | | 12 | | 139 | |
| | | TN | 1046 | | 338 | | 623 | |
| | | FP | 309 | | 87 | | 124 | |
| | | **HAM Messages** | - | | **438 @ 97.33%** | | **4686 @ 97.12%** | |
| | | **SPAM Messages** | **1063 @ 78.57%** | | **338 @ 79.53%** | | **623 @ 83.40** | |
| | | **Accuracy** | **78.57%** | | **88.69%** | | **95.23%** | |

From the Table 5.43, it can be stated that algorithm 4 manages to detect 78.57% of spam messages from 1353 messages in DIT dataset and 97.33% of ham messages from 450 messages with 79.53% of spam messages from 425 messages for BEC dataset. In UCI dataset, it can detect about 4686 out of 4825 of ham messages and 623 spam messages out of 747 messages. The explanation of process in each phase is shown in Table 5.36.

**Table 5. 44:** Results for Algorithm 5 using clean dataset

| Detection features | Phases | Process | DIT | | BEC | | UCI | |
|---|---|---|---|---|---|---|---|---|
| | | | HAM | SPAM | HAM | SPAM | HAM | SPAM |
| | | **Actual Messages** | - | **1353** | **450** | **425** | **4825** | **747** |
| Length, Characters and Keywords (2x) | PHASE 1 (length) | ToHam1 | - | 249 | 337 | 60 | 3809 | 92 |
| | | ToSpam | - | 1104 | 113 | 365 | 1016 | 665 |
| | PHASE 2 (characters and spam keywords) | ToHam1-Toham1 | - | 249 | 337 | 60 | 3808 | 92 |
| | | ToSpam-Spam | - | 1045 | 12 | 339 | 139 | 625 |
| | | ToSpam-ToHam2 | - | 59 | 101 | 26 | 877 | 30 |
| | PHASE 3 (spam keywords) | ToHam1-Toham1-Ham | - | 127 | 336 | 25 | 3804 | 33 |
| | | ToSpam-Spam-Spam | - | 1045 | 12 | 339 | 139 | 625 |
| | | ToSpam-ToHam2-ToHam2 | - | 59 | 101 | 26 | 877 | 30 |
| | | ToHam1-Toham1-Spam | - | 122 | 1 | 35 | 5 | 59 |
| | PHASE 4 (ham keywords) | ToSpam-Spam-Spam-Spam | - | 1045 | 12 | 339 | 139 | 625 |
| | | Tospam-ToHam2-ToHam2-Ham | - | 60 | 101 | 27 | 877 | 32 |
| | | ToSpam-ToHam2-ToHam2-Spam | - | 0 | 0 | 0 | 0 | 0 |
| | | ToHam1-ToHam1-Spam-Spam | - | 121 | 1 | 34 | 5 | 57 |
| | | ToHam1-ToHam1-Ham-Ham | - | 127 | 336 | 25 | 3804 | 33 |
| | RESULTS | TP | - | | 437 | | 4681 | |
| | | FN | - | | 13 | | 144 | |
| | | TN | 1166 | | 373 | | 682 | |
| | | FP | 187 | | 52 | | 65 | |
| | | **HAM Messages** | - | | **437 @ 97.11%** | | **4681 @ 97.02%** | |
| | | **SPAM Messages** | **1166 @ 86.18%** | | **373 @ 87.76%** | | **682 @ 91.30%** | |
| | | **Accuracy** | **86.18%** | | **92.57%** | | **96.25%** | |

From the Table 5.44, it can be stated that Algorithm 1 manages to detect 86.18% of spam messages from 1353 messages while in BEC dataset, it can detect 97.11% of ham messages from 450 messages and 87.76% of spam messages from 425 messages. For UCI dataset, about 4681 ham messages can be detected from 4825 messages and 682 spam messages from 747 messages. The explanation of process in each phase is discussed in Table 5.38.

**Table 5. 45:** Summary results of detection in clean dataset

| Dataset | Algorithm | Detection Feature(s) | Simulation Results | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Correctly Classified | Incorrectly Classified | TP | FN | TN | FP |
| DIT | 1 | Keywords | 1290 = 95.3437% | 63 = 4.6563% | 0 | 0 | 1290 | 63 |
| | 2 | Length and Keywords | 1113 = 82.2616% | 240 = 17.7384% | 0 | 0 | 1113 | 240 |
| | 3 | Characters and keywords | 1220 = 90.1700% | 133 = 9.8300% | 0 | 0 | 1220 | 133 |
| | 4 | Length, Character and Keywords | 1046 = 77.3097% | 309 = 22.8381% | 0 | 0 | 1046 | 309 |
| | 5 | Length, Character and Keywords (2x) | 1166 = 86.1789% | 187 = 13.8211% | 0 | 0 | 1166 | 187 |
| BEC | 1 | Keywords | 753 = 86.0571 | 124 = 14.1714 | 347 | 105 | 406 | 19 |
| | 2 | Length and Keywords | 776 = 88.6857% | 99 = 11.3143% | 408 | 42 | 368 | 57 |
| | 3 | Characters and keywords | 807 = 92.2286% | 68 = 7.7714% | 421 | 29 | 386 | 39 |
| | 4 | Length, Character and Keywords | 776 = 88.6857% | 99 = 11.3143% | 438 | 12 | 338 | 87 |
| | 5 | Length, Character and Keywords (2x) | 810 = 92.5714% | 65 = 7.4286% | 437 | 13 | 373 | 52 |
| UCI | 1 | Keywords | 4156 = 74.5872% | 1416 = 25.4128 | 3432 | 1393 | 724 | 23 |
| | 2 | Length and Keywords | 4787 = 85.9117% | 785 = 14.0883% | 4127 | 698 | 660 | 87 |
| | 3 | Characters and keywords | 5094 = 91.4214% | 692 = 12.4192% | 4394 | 645 | 700 | 47 |
| | 4 | Length, Character and Keywords | 5309 = 95.2800% | 263 = 4.7200% | 4686 | 139 | 623 | 124 |
| | 5 | Length, Character and Keywords (2x) | 5363 = 96.2491% | 209 = 3.7509% | 4681 | 144 | 682 | 65 |

Table 5.45 shows the overall results of detection using clean dataset in term of correctly classified into ham and spam. Algorithms 1 is good performance in term of accuracy for DIT dataset and algorithms 5 is more accurate for both BEC and UCI dataset. One of the factor that give different performance of algorithm in each dataset is because of the size of messages with the content in dataset is not same each other. In DIT dataset, Algorithm 1 has highest accuracy compare to others. Three features are used in detection phase which are length of messages, special character and keywords of spam and ham. As this experiment using clean dataset, it gives effect the performance of algorithms that used length of messages and special characters as the features. That is why Algorithm 1 is good for DIT dataset. However, for BEC and UCI dataset, both of them have algorithm 5 as the best performance because the algorithm using repeating keywords of spam in phase two and three for detection. Besides, DIT dataset that do not contain ham messages is another factor why it is different with BEC and UCI in term of accuracy.

v. Part 5: Comparison between raw and clean dataset

The comparison is made to compare the performance of five algorithms using raw and clean dataset for three dataset (i.e., DIT, BEC and UCI) as shown in Figure 5.21 until Figure 5.23. Raw dataset is dataset that contains all symbol and punctuation marks while clean dataset is dataset that only contains world and has gone through preprocessing to clean unimportant data.

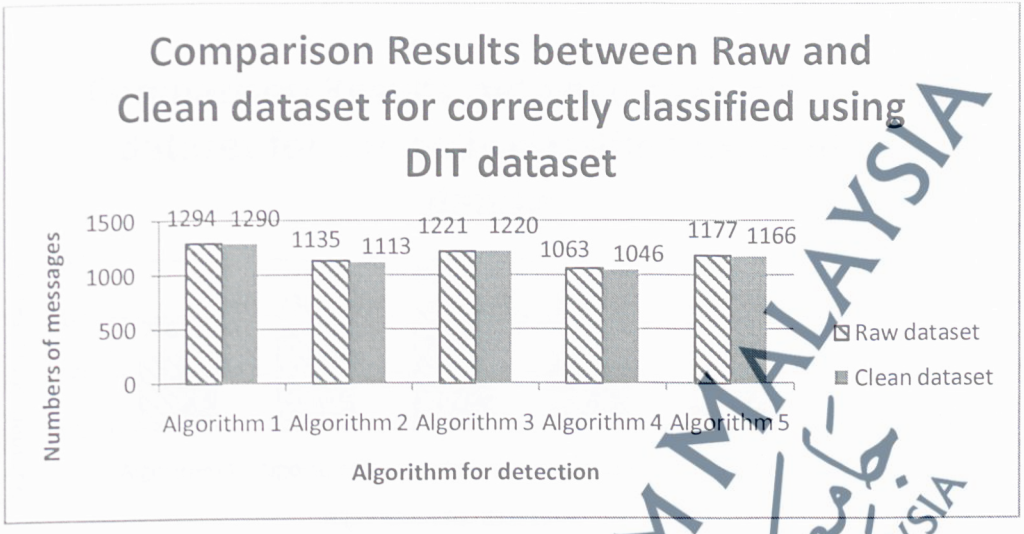


**Figure 5. 21:** Comparison results for DIT dataset

Figure 5.21 shows the comparison results of detection using raw and clean dataset for DIT dataset. From Figure 5.21, it indicates that raw dataset has highest accuracy for all five algorithms. This is because length of messages with special character is two features that are used in these algorithms. When these features are applied using clean dataset, it affects the performance of detection in term of length of messages as we remove the punctuation and symbol in the messages.

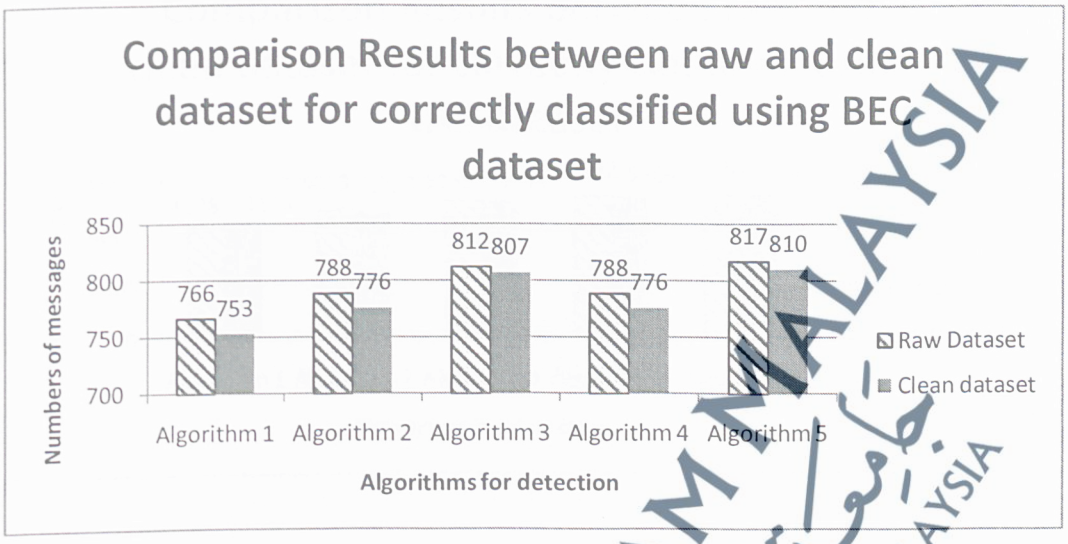


**Figure 5. 22:** Comparison results for BEC dataset

From Figure 5.22, it can be found that raw dataset give accurate results and higher number of detection compared to clean dataset. As clean dataset remove the punctuation and symbol in the messages, it will affect the length of message that is why the raw dataset has better results. However, the different between both types of dataset is not huge, hence it can suggest that clean dataset also give fair performance and results.

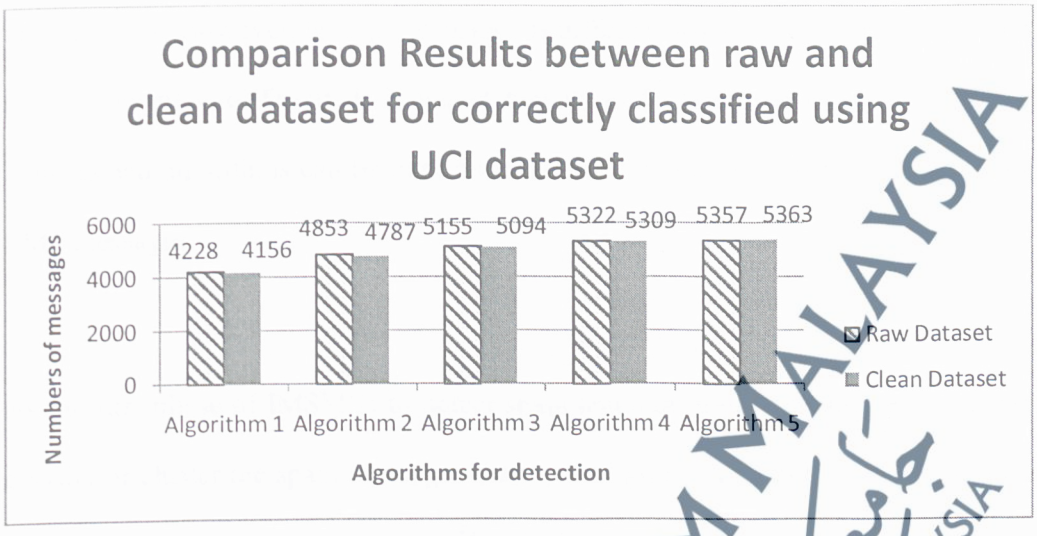


**Figure 5. 23:** Comparison results for UCI dataset

From Figure 5.23, it can be stated that raw dataset is highest in correctly classified messages into ham and spam for Algorithm 1 until 4. For Algorithm 5, clean dataset is more accurate with small different number of messages with raw dataset. Sometimes, the bigger size of dataset with various form and contents of messages give affect the performance of algorithms. So it can be suggest that clean dataset has highest accuracy for Algorithm 5 due to the various numbers of messages in dataset with different contents and style. Hence, although length of messages and special characters give big effect towards the performance of detection process for clean dataset, but the content of messages that have many keywords cause it has better result compare to raw dataset.

From this experiment, it can be suggest that using 'length of messages', 'special character' and 'keywords' as the features in enhancing the performance of Danger Theory for detecting SMS messages give better results and has reliability in term of performance

and accuracy. However, their performance is different when using different size of dataset as the content is different. It is hoped that these features can be enhanced or another features and algorithms can be introduced in order to improve the process of detection in SMS messages.

As the first phase of IMSM is to detect spam and ham messages, the second phase is to classify or cluster the spam messages into several groups and categories. Spam messages are chosen because we want to identify which group of spam messages that always give harm and loss to mobile users. Next section discusses the proof of performance for classification process.

### 5.4.2 CLASSIFICATION PHASE

This section presents several results of classification that are divided into six parts. Part 1 reported the results for testing the proposed algorithm called HICNA and part 2 discusses the results for validation the algorithm. Part 3 compares the results between experiment in part 1 and 2. The discussion on how AIS algorithms are used in HICNA is discussed in part 4, with part 5 shows how HICNA gives output for both detection and classification process simultaneously and lastly the comparison results between HICNA and k-Means clustering algorithm are discussed to evaluate their performances.

i. Part 1: Testing

The aim of this experiment is to test the proposed algorithm which is HICNA to view its capability for classification (i.e. clustering) using different dataset. There are three phases involved in this part. In the phase one, it clusters the spam messages using common keywords (Appendix C). The messages that do not match with any keywords are grouped to cluster "Miscellaneous". Common keywords mean that the messages have familiar and frequent keywords for different dataset. Then all messages in cluster "Miscellaneous" will be clustered again in phase two using uncommon keywords.

Phase two uses uncommon keywords (Appendix C) to cluster updated spam messages. Uncommon keywords are keywords that are new and not familiar. Every year, spammers will update their style of spam messages by using new contents and term so that they can avoid any software of filter spam messages. By having this phase, the messages can be clustered although new updated of spam messages produced.

The last phase is for the expert judgement process and this is the final phase for clustering process. Although the messages have been clustered into their own group but there are a few that are incorrectly classified. This problem occurs because some messages contain keywords for certain cluster although the meaning is different and it called as False Positive (i.e. FP). To overcome this, all messages from FP and also messages from cluster 'Other' are clustered again based on expert judgment (i.e. messages are structured into correct cluster based on the understanding the meaning of messages). The results of clustering in these three phases are shown in the Table 5.46.

**Table 5. 46:** Results of classification using different dataset

| PHASE 1 | PHASE 2 | PHASE 3 | UCI machine Learning | | British English Corpora | | Dublin Institute of Technology | | SMSv.0.1 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | TP | FP | TP | FP | TP | FP | TP | FP |
| Financial | | | 30 | 0 | 22 | 0 | 81 | 0 | 8 | 0 |
| Offer | | | 56 | 0 | 36 | 0 | 63 | 0 | 20 | 0 |
| Prize | | | 179 | 8 | 59 | 5 | 208 | 9 | 120 | 3 |
| Service | | | 30 | 0 | 18 | 0 | 50 | 0 | 12 | 0 |
| Games/Competition | | | 59 | 1 | 34 | 0 | 57 | 7 | 25 | 1 |
| Ringtone | | | 69 | 6 | 44 | 3 | 75 | 6 | 26 | 2 |
| Chat | | | 69 | 6 | 48 | 3 | 98 | 12 | 21 | 3 |
| Date | | | 20 | 2 | 10 | 1 | 72 | 3 | 10 | 1 |
| Sex | | | 41 | 6 | 30 | 3 | 64 | 5 | 11 | 3 |
| Voice mail | | | 12 | 0 | 8 | 0 | 58 | 0 | 4 | 0 |
| Mail message | | | 2 | 0 | 2 | 0 | 4 | 0 | 0 | 0 |
| Claim | | | 2 | 0 | 2 | 0 | 194 | 0 | 0 | 0 |
| Entertainment | | | 9 | 0 | 6 | 0 | 16 | 0 | 3 | 0 |
| Miscellaneous | | | 140 | 0 | 91 | 0 | 271 | 0 | 49 | 0 |
| | Financial | | 5 | 2 | 1 | 2 | 25 | 3 | 4 | 0 |
| | Offer | | 3 | 0 | 2 | 0 | 8 | 0 | 1 | 0 |
| | Prize | | 14 | 8 | 6 | 3 | 20 | 9 | 8 | 5 |
| | Service | | 27 | 3 | 19 | 3 | 35 | 4 | 8 | 0 |
| | Game/Competition | | 14 | 3 | 11 | 2 | 15 | 4 | 3 | 1 |
| | Chat | | 3 | 7 | 3 | 3 | 26 | 10 | 1 | 3 |
| | Sex | | 1 | 1 | 0 | 1 | 12 | 2 | 1 | 0 |
| | Date | | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| | Advertisement | | 9 | 0 | 5 | 0 | 8 | 0 | 4 | 0 |
| | Entertainment | | 8 | 1 | 7 | 1 | 10 | 4 | 1 | 0 |
| | Job | | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| | Voicemail | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| | Mail message | | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| | Claim | | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| | Other | | 29 | 0 | 20 | 0 | 66 | 0 | 9 | 0 |
| | | Financial | 35 | | 23 | | 111 | | 12 | |
| | | Offer | 64 | | 40 | | 77 | | 24 | |
| | | Prize | 199 | | 69 | | 232 | | 129 | |
| | | Service | 59 | | 39 | | 90 | | 20 | |
| | | Game/Competition | 89 | | 53 | | 88 | | 36 | |
| | | Ringtone | 69 | | 44 | | 75 | | 26 | |
| | | Chat | 74 | | 52 | | 127 | | 23 | |
| | | Date | 22 | | 11 | | 74 | | 11 | |
| | | Sex | 45 | | 33 | | 82 | | 12 | |
| | | Voicemail | 16 | | 11 | | 65 | | 5 | |
| | | Mail message | 2 | | 2 | | 6 | | 0 | |
| | | Claim | 2 | | 2 | | 198 | | 0 | |
| | | Entertainment | 31 | | 20 | | 37 | | 8 | |
| | | Advertisement | 19 | | 9 | | 19 | | 9 | |
| | | Job | 0 | | 0 | | 7 | | 0 | |
| | | Other | 21 | | 17 | | 65 | | 7 | |
| | | **Total** | **747** | | **425** | | **1353** | | **322** | |

Table 5.46 shows the results for clustering SMS spam messages using four different dataset. True Positive (TP) and False Positive (FP) were used as a measurement to measure the clustering process. TP measures how many messages are correctly classified in right cluster while FP measures how many messages are wrongly classified. In the first phase, it was found that six clusters obtained FP which are "Prize", "Game/Competition", "Sex", "Ringtone", "Chat" and "Date". This happens because some messages may have same keywords that exactly with the cluster but with different meaning. Messages that do not match with any keywords in each cluster will go to cluster "Miscellaneous" and they will be clustered again in phase two.

In the phase two, two new clusters were created based on uncommon keywords available and the clusters are "Advertisement" and "Job". Every year, the contents of spam messages will change using new terms and words. So, new clusters were created due to updated spam messages. All spam messages in cluster "Miscellaneous" continue the process of clustering into their own groups. Messages that do not match with any keywords were grouped in cluster "Other". Finally, in the phase three, messages labeled as FP and messages from cluster "Other" will be clustered manually in order to group them into identified categories.

The overall results of clustering SMS spam messages showed in phase three using two combination techniques which are keywords and expert judgment. It was found that majority of spam messages are categorized in cluster "Prizes" as all four dataset has the highest number of spam messages in that cluster. The gift, money and prize that are

offered have interest and attract users to believe and they feel curious to try it. Cluster "Job" has lowest number of spam messages as normally users know how to find job in right source or website and are aware that usually company will call or send an email to offer the job. The second highest is cluster "Game/Competition" for dataset UCI, BEC and SMSv.0.1 but for dataset DIT is cluster "Claim". However, after going through all three phases in HICNA, there are some messages that cannot be grouped and they are classified as cluster "Other". This is because SMS messages have different and variation of text and style, so it gives limitation for HICNA to cluster them using common and common keywords. Although human judgment is used to cluster messages based understanding the content, sometimes it is hard as the meaning of message is not clear. In summary, different numbers of messages for each dataset give effect the number of messages in each cluster, therefore all clusters in each dataset has a different number and results. For this experiment, Delaney et al., (2012) are used as a benchmark for clustering spam messages into several types of spam. However there are limitations in their research that has been discussed in Chapter 4, Section 4.3.2.

ii. Part 2: Validation

The purpose of this experiment is to make a validation whether the results of combining all spam dataset can give accurate results or same results with previous experiment (i.e. experiment in part 1) and to see the reliability of HICNA in classifying spam messages. FadhilahSpam dataset was used to cluster the spam messages into several groups. This dataset is collection of spam messages from four sources of dataset (i.e. Dublin, UCI,

BEC and SMSv.0.1). FadhilahSpam dataset contains 2847 number of spam messages and they were tested in three phases. Table 5.47 shows the results for clustering SMS spam messages using all three phases.

Table 5.47 shows the number of spam messages using three phases in HCNA. Again it can be verified that the cluster "Prize" in phase one has the highest number of spam messages while cluster "Job "is the lowest one. Message related to prize has attracted user to try and subscribe it with many offer provided to user. That is why cluster prize get higher number of messages compared to other cluster. Spam messages that do not match with any spam keywords will go to cluster "Miscellaneous" and this cluster will be run again in phase two. From 2847 number of spam messages, only 3.86% messages cannot be clustered and there are also several messages that wrongly classified in certain clusters (i.e. FP).

In the phase two, we use spam messages from cluster "Miscellaneous" and cluster it using uncommon spam keywords. In this phase, cluster "Service" get the higher number of messages while cluster "Voicemail" do not has any messages.

The final stage shows that cluster "Prize" has the higher number of spam messages, then goes to cluster "Game/Competition" and "Chat" with the different number of messages between both clusters are ten messages. The least messages go to cluster "Job".

**Table 5. 47:** Results of classification using FadhilahSpam dataset

| PHASE 1 | PHASE 2 | PHASE 3 | FadhilahSpamDataset | |
|---|---|---|---|---|
| | | | TP | FP |
| Financial | | | 141 | 0 |
| Offer | | | 175 | 0 |
| Prize | | | 566 | 25 |
| Service | | | 110 | 0 |
| Game/Competition | | | 175 | 9 |
| Ringtone | | | 214 | 17 |
| Chat | | | 236 | 24 |
| Date | | | 112 | 7 |
| Sex | | | 146 | 17 |
| Voicemail | | | 82 | 0 |
| Mail message | | | 8 | 0 |
| Claim | | | 198 | 0 |
| Entertainment | | | 34 | 0 |
| Miscellaneous | | | 551 | 0 |
| | Financial | | 35 | 1 |
| | Offer | | 14 | 0 |
| | Prize | | 48 | 25 |
| | Service | | 89 | 10 |
| | Game/Competition | | 43 | 10 |
| | Chat | | 33 | 23 |
| | Sex | | 14 | 4 |
| | Date | | 3 | 0 |
| | Advertisement | | 26 | 0 |
| | Entertainment | | 26 | 6 |
| | Job | | 2 | 0 |
| | Voicemail | | 0 | 3 |
| | Mail message | | 2 | 0 |
| | Claim | | 4 | 0 |
| | Other | | 124 | 0 |
| | | Financial | 181 | |
| | | Offer | 205 | |
| | | Prize | 629 | |
| | | Service | 208 | |
| | | Game/Competition | 266 | |
| | | Ringtone | 214 | |
| | | Chat | 276 | |
| | | Date | 118 | |
| | | Sex | 172 | |
| | | Voicemail | 97 | |
| | | Mail message | 10 | |
| | | Claim | 202 | |
| | | Entertainment | 96 | |
| | | Advertisement | 56 | |
| | | Job | 7 | |
| | | Other | 110 | |
| | | **Total** | **2847** | |

iii. Part 3: Comparison between experiment part 1 and experiment part 2

The purpose of making comparison between both experiments is to see the feasibility and capability of the proposed model (i.e. HICNA) in clustering spam messages using various datasets. Figure 5.24 shows the number of messages in each cluster for experiment 1 and 2.

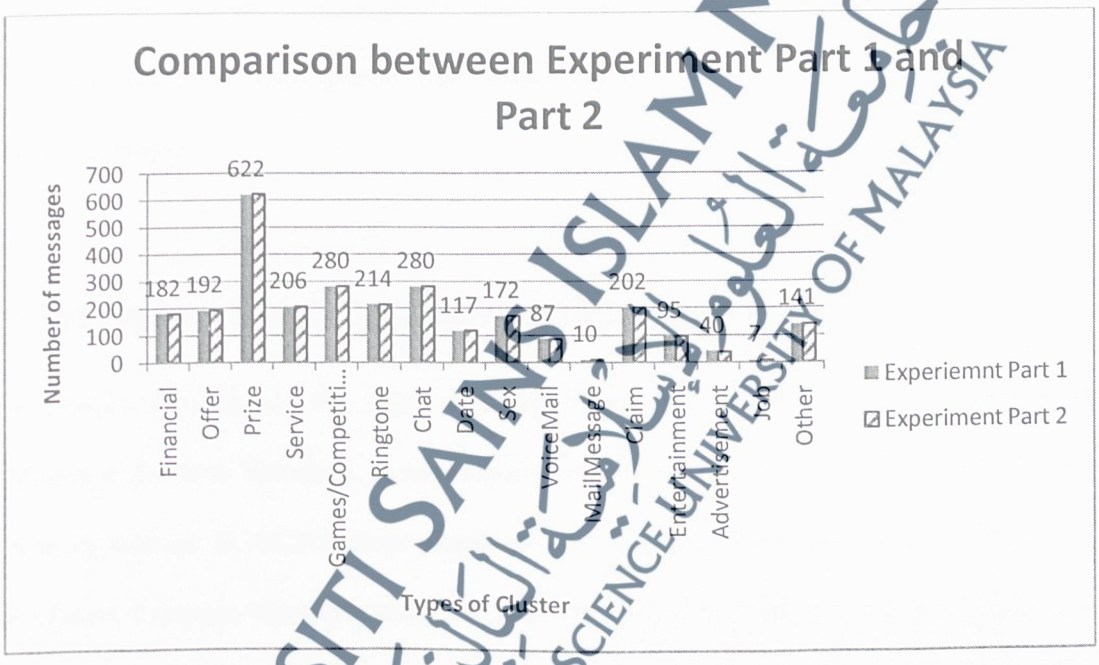


**Figure 5. 24:** Number of messages in each cluster between experiment part 1 and part 2

From Figure 5.24, it shows the numbers of messages that exactly same between previous experiments (experiment part 1) with the experiment part 2. It can be concluded that the proposed algorithm can be used in clustering phase with convincing results although using different dataset.

The top five clusters that have highest numbers of spam messages (i.e. above 200 spam messages) are "Claim", "Service", "Chat"," Game/Competition" and the highest cluster is cluster "Prize" with the number of messages are 622 messages. From the list of these five clusters, it can be suggested that majority of spammers choose to send messages that offer prize, gift, money or any offer that give convenience to mobile users so it easy to attract them to involve with the content of messages. Cluster "Mail messages" and cluster "Job" is the lowest number of messages as users nowadays knowledgeable about how to find the right job from legal sources and receive correct mail messages through email, not SMS messages.

iv. Part 4:Immune Network Theory and Clonal Selection in HICNA

In classification phase, two types AIS algorithms are used namely Immune Network Theory and Clonal Selection. A new algorithm is proposed named HICNA by combining both algorithms. In HICNA three phases involve to cluster the spam messages. Phase one is cluster the spam messages using common keywords and phase two using uncommon keywords. Last phase using the expert judgment based on the understanding the meaning of messages.

Immune Network Theory occurs in all three phases of HICNA. When the spam messages are scanned using common and uncommon keywords in each cluster, they will attract and grouped into cluster that has close relationship and interaction based on the keywords. Same goes with expert judgment, when we understand the meaning of messages, we can

identify which cluster that has similar contents of messages. In BIS, Immune Network Theory occurs when there is interaction between an antibody with an antigen or between themselves. In HICNA, this theory happens when there is interaction between messages with the keywords and meaning of the cluster.

Clonal Selection happens when the cell is cloned into two types of cell namely plasma cell and memory cell. Plasma cell will fight the bacteria while memory cell will remember the structure of bacteria and prepare when seconds attack happens. In HICNA, it occurs when spam messages are clustered into several clusters and categories. In addition, it also happens in phase two when two new clusters are introduced. Besides, memory cell happens in phase two when several messages that cannot be clustered in phase two will be clustered again in this phase. So HICNA will memorize the structure of messages during first phase and will take action in second phase of HICNA.

v.Part 5 : Detection and classification using HICNA

The aim of this experiment is to identify whether the combination of detection and classification process can be done using HICNA. Only three dataset contains ham and spam messages were used namely UCI, BEC and SMSv.0.1 and the results are shown in Table 5.48. The discussion of process using HICNA for combination of both processes as follow:-

a. The purpose of this experiment is to see the capability of HICNA in detection and classification simultaneously.

b. The number of spam messages is shown in three classes namely TP, FP and Ham. True Positive (i.e. TP) means the spam messages correctly classified into identified group. False Positive (i.e. FP) is the spam messages incorrectly classified into identified group while Ham is for the ham messages.

c. If spam messages managed to identified into correct cluster, they will classified as TP.

d. If spam messages incorrectly classified into correct cluster, they will classified as FP.

e. If ham messages classified into several cluster available, they will classified as FN (i.e. the ham messages incorrectly classified into spam messages).

f. If ham messages managed to be classified into cluster "Miscellaneous" or cluster "Other", they are classified as TN (i.e. the messages are correctly classified as ham).

**Table 5. 48:** Results using HICNA for detection

| Phase 1 | Phase 2 | Phase 3 | UCI machine Learning | | | British English Corpora | | | SMSv.0.1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | TP | FP | HAM | TP | FP | HAM | TP | FP | HAM |
| Financial | | | 30 | 0 | 22 | 22 | 0 | 1 | 8 | 0 | 0 |
| Offer | | | 56 | 0 | 9 | 36 | 0 | 0 | 20 | 0 | 1 |
| Prize | | | 179 | 8 | 39 | 59 | 5 | 4 | 120 | 3 | 5 |
| Service | | | 30 | 0 | 7 | 18 | 0 | 0 | 12 | 0 | 1 |
| Game/ Competition | | | 59 | 1 | 115 | 34 | 0 | 10 | 25 | 1 | 9 |
| Ringtone | | | 69 | 6 | 4 | 44 | 3 | 0 | 26 | 2 | 0 |
| Chat | | | 69 | 6 | 130 | 48 | 3 | 7 | 21 | 3 | 11 |
| Date | | | 20 | 2 | 21 | 10 | 1 | 3 | 10 | 1 | 4 |
| Sex | | | 41 | 6 | 52 | 30 | 3 | 7 | 11 | 3 | 10 |
| Voicemail | | | 12 | 0 | 0 | 8 | 0 | 0 | 4 | 0 | 0 |
| Mail message | | | 2 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| Claim | | | 2 | 0 | 5 | 2 | 0 | 1 | 0 | 0 | 1 |
| Entertainment | | | 9 | 0 | 2 | 6 | 0 | 0 | 3 | 0 | 1 |
| Miscellaneous | | | 140 | 0 | 4419 | 91 | 0 | 417 | 49 | 0 | 959 |
| | Financial | | 5 | 2 | 110 | 1 | 2 | 18 | 4 | 0 | 18 |
| | Offer | | 3 | 0 | 1 | 2 | 0 | 1 | 1 | 0 | 0 |
| | Prize | | 14 | 8 | 83 | 6 | 5 | 12 | 8 | 5 | 8 |
| | Service | | 27 | 3 | 37 | 19 | 3 | 3 | 8 | 0 | 6 |
| | Game/ Competition | | 14 | 3 | 107 | 11 | 2 | 12 | 3 | 1 | 10 |
| | Chat | | 3 | 7 | 247 | 3 | 3 | 24 | 1 | 3 | 32 |
| | Sex | | 1 | 1 | 189 | 0 | 1 | 21 | 1 | 0 | 66 |
| | Date | | 1 | 0 | 9 | 1 | 0 | 1 | 0 | 0 | 0 |
| | Advertisement | | 9 | 0 | 29 | 5 | 0 | 3 | 4 | 0 | 5 |
| | Entertainment | | 8 | 1 | 198 | 7 | 1 | 21 | 1 | 0 | 42 |
| | Job | | 0 | 0 | 49 | 0 | 0 | 4 | 0 | 0 | 1 |
| | Voicemail | | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | Mail message | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Claim | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Other | | 29 | 0 | 3360 | 20 | 0 | 297 | 9 | 0 | 771 |
| | | Financial | 35 | | 132 | 23 | | 19 | 12 | | 18 |
| | | Offer | 64 | | 10 | 40 | | 1 | 24 | | 1 |
| | | Prize | 199 | | 122 | 69 | | 16 | 129 | | 13 |
| | | Service | 59 | | 44 | 39 | | 3 | 20 | | 7 |
| | | Game/Competition | 89 | | 222 | 53 | | 22 | 36 | | 19 |
| | | Ringtone | 69 | | 4 | 44 | | 0 | 26 | | 0 |
| | | Chat | 74 | | 377 | 52 | | 31 | 23 | | 43 |
| | | Date | 22 | | 30 | 11 | | 4 | 11 | | 4 |
| | | Sex | 45 | | 241 | 33 | | 28 | 12 | | 76 |
| | | Voicemail | 16 | | 0 | 11 | | 0 | 5 | | 0 |
| | | Mail message | 2 | | 0 | 2 | | 0 | 0 | | 0 |
| | | Claim | 2 | | 5 | 2 | | 1 | 0 | | 1 |
| | | Entertainment | 31 | | 200 | 20 | | 21 | 8 | | 43 |
| | | Advertisement | 19 | | 29 | 9 | | 3 | 9 | | 5 |
| | | Job | 0 | | 49 | 0 | | 4 | 0 | | 1 |
| | | Other | 21 | | 3360 | 17 | | 297 | 7 | | 771 |
| | | **Total** | **747** | | **4825** | **425** | | **450** | **322** | | **1002** |

From Table 5.48, it indicates that, there are ham messages that can be clustered into categories and these messages classified as FN (i.e. ham messages incorrectly classified as spam). Although detection process using common and uncommon keywords of spam, some ham messages may have the same keywords as spam even though the meaning is different. The ham messages that are grouped into cluster "Other" are the messages that correctly classified as ham (i.e. TN). From Table 5.48, it can be found that 3360 ham messages can be detected as ham from 4825 for UCI dataset. For BEC dataset, only 297 ham messages managed to be detected from 450 messages and lastly in SMSv.0.1 dataset, 771 ham messages out of 1002 message manage to be detected. From this experiment, it can be suggested that HICNA can be used for both process. However, since HICNA uses Immune Network Theory and Clonal Selection, it is not that suitable to be used during detection although it capable of doing it. This is because the nature of themselves since their main function is to categories, not for detection.

Overall, from experiment proof of performance in classification, it can be concluded the proposed algorithm called HICNA managed to cluster spam messages into identified categories and help us in recognize which types of messages that are always sent by spammers. For the future, hoped this algorithm can be improves using other features to cluster the spam messages.

vi. Part 6: Comparison between HICNA with k-Means using WEKA

The purpose of this experiment is to make comparison between the performances of HICNA with k-Means clustering algorithms in WEKA for classifying spam messages into categories. k-Means is chosen as its performance is better and faster as compared to other classifiers algorithms in WEKA. DIT dataset contains 1353 spam messages were used in this experiment. Table 5.49 shows the comparison results between both techniques.

**Table 5. 49:** Comparison results of classification using HICNA and k-Means in WEKA

| Types of Cluster | Number of Messages using HICNA | Types of Cluster | Number of Messages using k-Means |
|---|---|---|---|
| Financial | 111 | 0 | 1338 |
| Offer | 77 | 1 | 1 |
| Prize | 232 | 2 | 1 |
| Service | 90 | 3 | 1 |
| Game/Competition | 88 | 4 | 1 |
| Ringtone | 75 | 5 | 1 |
| Chat | 127 | 6 | 1 |
| Date | 74 | 7 | 1 |
| Sex | 82 | 8 | 1 |
| VoiceMail | 65 | 9 | 1 |
| MailMesage | 6 | 10 | 1 |
| Claim | 198 | 11 | 1 |
| Entertainment | 37 | 12 | 1 |
| Advertisement | 19 | 13 | 1 |
| Job | 7 | 14 | 1 |
| Other | 65 | 15 | 1 |

From the Table 5.49, results show huge different between both techniques. The number of spam messages in each cluster is different using HICNA and k-Means in WEKA. This is because, the main function of WEKA is to see the performance of algorithms in term of time taken to build the model for classification, not see the results after classification process. Besides, WEKA shows types of cluster in form of number without knowing each

number represent which types of cluster. In addition, only cluster number 0 contains highest number of spam and the rest of cluster only having one spam messages. One of the reason is because WEKA may cluster the spam messages based on the structure and characteristic of messages, different with HICNA that using keywords of the spam message. So, most of the spam messages contains characters that closer with cluster 0 as our dataset in form of text messages.

## 5.5 DISCUSSION PROOF OF PERFORMANCE

Proof of performance is another experiment conducted in this research to identify the performance of detection and classification using proposed algorithms. For detection phase, there are five parts of test have been carried out to proof the performance of detection.

The first experiment has been conducted using AIS algorithms (i.e. Negative Selection and Danger Theory). Results clearly show that Negative Selection model has higher accuracy in detecting spam and ham in datasets. The values of accuracy given by Negative Selection model in all of three datasets are nearly accurate to the actual number of ham and spam in datasets. It can be concluded that the Negative Selection algorithm is much better than Danger Theory in terms of performance. Having understood to improve the performance of Danger Theory in detection, three features are introduced for detection; length, special characters and keyword and these features were stipulated into five algorithms.

From the conduct of experiments, it was found that the detection feature that is based on keywords (i.e. spam and ham) is still producing good detection results. For all five (5) algorithms, Algorithm 1 (i.e. using our own keywords of spam and ham) performs well and better in detecting spam messages for both raw and clean dataset. Here, we argue that in order to have an optimum detection result that method needs to have a 'sound' list of keywords. With respect to the clients' mobile environment, it is preferable to have a minimum list of keywords due to their limitation. On the other hand, if it meant to be implemented in servers' environment, it should be working perfectly. In addition, detection using keywords is sometimes 'unscrupulous' due to various language styles.

The number of messages in each dataset can also affect the detection rate as different datasets may have different number of spams and hams and also contains different message structures. In term of accuracy for raw dataset, results suggest that Algorithm 5 produced high accuracy for UCI dataset as compared to others but Algorithm 3 performed better for the BEC dataset albeit using only two of the proposed features. The similar condition is occurred for the DIT dataset. In this dataset, Algorithm 1 produced higher accuracy as compared to others although this algorithm used only one feature; which is keywords. We anticipate different detection results are due to the 'immature' of the algorithms and there is a need for improvement. When it combined with the machine learning classifier, we expect it to improve and perform better.

We reported that our model produced a 'fair' detection results and we expect this is caused by two conditions. First is due to the way we do our detection. Specifically, the

algorithms work on a 'phase-by-phase' basis and they were non-iterative (i.e. static detection). Using static detection might limit the detected results, as messages that are less than 100 of length were not detected at the first stage (Algorithm 2, 4 and 5). Second is due to the nature of the messages itself, as not all messages contain special characters (Algorithm 3). However, from our second validation, we can suggest that our algorithms managed to detect all spam messages that contain 'spam' special characters, and greater than 100 in length.

Besides, a test was conducted to identify the capability of clean dataset in detection using all five algorithms. Results suggest that, although the special characters (i.e. one of the features for detection) are removed and give effect the length of messages (i.e. another feature for detection) , it still can produce nice results. Here we can suggest that our algorithms applicable and can be used in detection phase.

From clustering point of view, it was found that the proposed clustering algorithm named HICNA using the combination of Clonal Selection and Immune Network Theory based on features keywords of spam messages is still producing good and better clustering results. Four experiments have been conducted for testing and validation the algorithm. Experiment 1 is conducted to test the algorithm using four different dataset contains text spam messages. In term of accuracy, we can say that our proposed algorithm can give a better result as majority of spam messages in each dataset is clustered to their dedicated categories. Experiment 2 is done for the validation purpose using the combination of four dataset used from experiment 1. Again the results show that our proposed model can give

better results for clustering the text spam messages. Experiment 3 is carried out to identify the capability of HICNA in detection and classification simultaneously. Results suggest that HICNA enable to detect the messages into ham and spam although it is huge different number for ham messages with the actual ham messages. The last experiment is to make comparison of classification process using HICNA and k-Means clustering algorithm in WEKA. Results can be concluded that the limitation of WEKA give affect towards the performance of k-Means clustering algorithms. That is why it produces different results between HICNA.

The purpose of doing clustering process is to identify the motif of spammers sending the messages. Maybe their motive is just for fun, or to get profit from users. By doing clustering, we can identify the patterns of the text messages by looking which group that having larger number of spam messages. From the conduct of experiment 1, it can be suggested that cluster "Prize" has the highest number of spam messages for all four dataset. From this scenario, we can say that most of spammers send the spam messages in form of prize because they believe by sending messages that offering prize and money or any voucher, it can easily attract people to get that prize. Cluster "Game/Competition" is the second higher number of messages for all dataset except DIT while Cluster "Chat" is the third higher for dataset UCI, BEC and DIT. The different number of spam messages in each cluster for all dataset depends on the size of dataset itself. The lowest number of messages is cluster "Job" for all dataset except DIT because mostly users will find the job through the Internet and not SMS in mobile phone. Overall from classification spam messages, it can be said that mostly spammers aims to send spam messages for the

purpose of obtaining money and stealing personal information from users. They believe that by sending attractive and interesting messages, user tends to believe without any doubt.

## 5.6 SUMMARY

Two types of experiment were conducted in this research for spam messages. The first experiment is the proof the concept of detection and clustering using WEKA. WEKA is an open source for data mining and it is normally used by researchers in detection and classification. However, the main function of WEKA is to identify the best performance of algorithms. Another experiment is to proof the performance of proposed algorithms. In detection phase, there are five part of test carried out to detect the messages into ham and spam and to identify the usability of five proposed algorithm and AIS algorithms used in this phase. Meanwhile for classification, there are three part of test available to test the capability of HICNA to cluster the messages. From this experiment, it can be suggested that the proposed algorithm in detection and classification can give good results.