CHAPTER 6

Detection and Containment of Second-Channel Propagating Monomorphic Characteristic worm based on Illegal Traffic using the Proposed ECB Method

6.1. Introduction

In this chapter, the combined algorithm namely, **E**nhanced **C** 4.5 Algorithm and **B**lacklist (**ECB**) method is used to detect the second channel propagating worms and containment of malicious traffic created by botnet propagation. The worms delivered through Second Channel affects the network through their botnet propagation, where botnets are used to propagate worms, spams and spywares. This propagation creates illegal traffic and tends to Distributed Denial-of-Service through their abnormal behaviors in the network. The malicious IP addresses that create traffic through TCP/UDP transmissions are monitored. Based on their TCP/UDP flow characteristics, the Internet worms are detected and classified. The classified malicious IP addresses are blocked.

Incoming traffic into the network are scanned and the attributes like source IP, destination IP, port addresses, protocol, average payload packet length, variance of payload packet length for time interval, number of packets exchanged for time interval and that exchanged per second, size of the first packet in the flow, average time between packets in time interval, number of reconnects for a flow and number of flows from the address over the total number of flows are generated per hour. The attributes are the collection of network flow for a specified time interval. These attributes are extracted from TCP and UDP transmissions. Based on the value of the attribute vector, C 4.5 classifies malicious and non-malicious traffic flows.

C4. 5 with Pearson's Correlation Coefficient give a better classification accuracy based on their flow characteristics. Detected unused addresses are assigned with negative values. The weight of the negative values is computed and blocked using the blacklist method. The proposed method contains different steps and is discussed below.

6.2. Steps of the Proposed Contribution Three

The objective of contribution three is to detect and contain Monomorphic characteristic worms based on their illegal traffic from unused IP addresses with better accuracy. Containment of these malicious IP addresses blocks the malicious traffic flow in the network.

Figure.6.1 shows the three different steps of the proposed contribution three namely,

- Analysis and Preprocessing
- Detection and Classification of Internet Worms
- Containment of Internet Worms

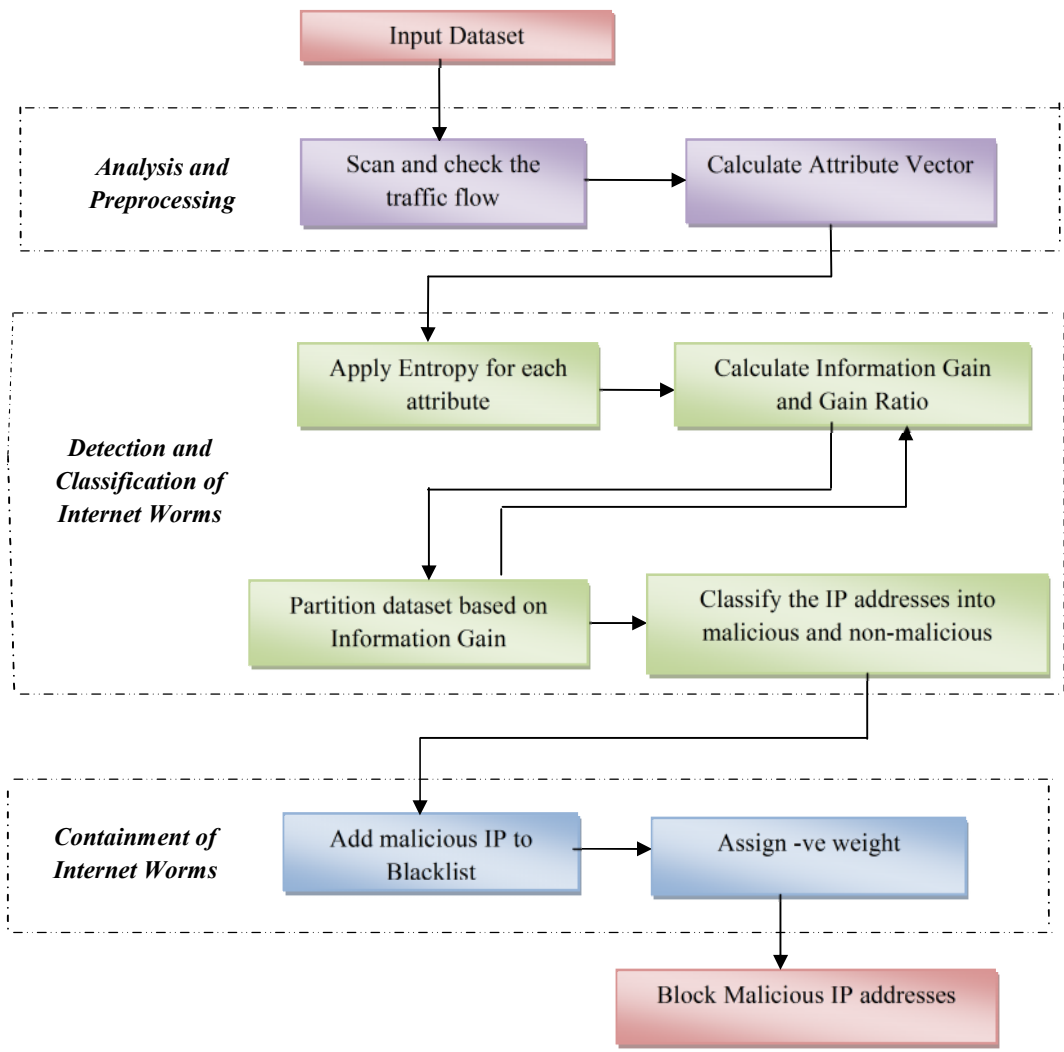


Figure.6.1. Block diagram of the Proposed Contribution

To detect the illegal traffic created by botnet propagation and containment of its infection, a combined method called ECB method is proposed. ***ECB*** is a combination of ***E***nhanced ***C*** 4.5 Algorithm and ***B***lacklist method .

6.2.1. Analysis and Preprocessing

In the analysis step, the traffic through TCP/UDP transmissions and their flows are analyzed. The flow analysis is done using the set of attributes.

6.2.1.1. Attribute Vector Selection

Attribute represents the numeric value that contains the collection of flows gathered at a given time Window T. It consists of source and destination IP address and ports of flow of destination and source, average time for the packets and average length of packets exchanged in the time interval.

Attribute vector consists of the collection of attributes, that gathers the characteristics of individual flow for a specified time interval. The attribute vector is measured by comparing the total number of flows made by a particular single address with the total flows count made in some limited time period. Attributes in the proposed work are given in the table.6.1.

Table.6.1. Attributes from Network Flow

Attribute	Description
SrcIP	Flow Source IP address
SrcPort	Flow Source Port address
DstIP	Flow Destination IP address
DstPort	Flow Destination IP address
Protocol	Transport layer protocol or mixed
APL	Average payload packet length of time interval T
PV	Variance of Payload packet length for time interval T
PX	Number of Packets exchanged for a time interval T
PPS	Number of Packets exchanged per second in time interval T
FPS	Size of the first packet in the flow
TBP	Average time between packets in time interval T
NR	Number of reconnects for a flow
FPH	Number of flows from this address over the total number of flows generated per hour

After the analysis of the attributes selection, the detection and classification of the traffic flow is applied and are discussed in next section.

6.2.2. Detection and Classification of Internet Worms

The **C** 4.5 with **P**earson's correlation **C**oefficient (**CPC**) method is used to detect and classify the malicious traffic traces. The C 4.5 based decision tree algorithm is applied for classifying the detected flows. Some limitations in C4.5 can be overcome using Enhanced C 4.5 algorithm which uses Pearson's Correlation Coefficient as entropy.

In Decision tree learning, decision tree is used as a predictive model which plots from the initial variables or features to target variable value based on several input variables. In classification trees, target variable contains the fixed set of values in tree models where leaves denotes the class labels and branches denotes the conjunctions of features that lead to those class labels. Each non-leaf node represents the one of the input features and path from nodes to children denote possible value of the labeled input features. Each leaf denotes target variable value or a class or a probability distribution over the classes represented by the path from the root to the leaf. The target variable can be achieved by splitting the initial set into subsets based on an attribute value test and this process is repeated in a recursive manner called recursive partitioning for each derived subsets. This process is occurred until subset at a node has all the same value of the target variable or no longer splitting is required and no additional value is added to the predictions.

6.2.2.1. C4.5 Algorithm

C4.5 is one of the decision tree based algorithm with a big tree and it contains certain attribute values and finalizes the decision rule using pruning method. It has features such as handling missing values, categorization of continuous attributes, pruning of decision trees and rule derivation. The most significant attributes are selected by considering all the samples, in which root nodes are considered as the top nodes of the tree. The subsequent nodes, which are termed as branch node, receive the sample information. The decision is made when it is terminated in the leaf node. Root node to leaf node is a path defined by several nodes in which rules are generated. The algorithm of the C4.5 is shown in table.6.2.

Table.6.2. Algorithm of C 4.5

Step 1: If all cases are of the same class, the tree is a leaf and is returned labeled with this class.
Step 2: For each attribute, calculate potential information provided by a test on the attribute.
Step 3: Also calculate the gain in information that results from a test on the attribute.
Step 4: Find best attribute to branch depending on the current selection

6.2.2.2. Entropy and Gain ratio

Information theory is a mathematical formula which is based on probability theory and statistics with conditions and parameters that influence the transmission and processing of information. The essential parts of information are called entropy.

Here entropy is implemented and is defined as to measure or calculate the disorder of the data. It is defined as

$$Entropy(\bar{y}) = -\sum_{j=1}^{n} \frac{|y_j|}{|\bar{y}|} \, log \frac{|y_j|}{|\bar{y}|} \quad - \quad (6.1)$$

iterating over all possible values of $|\bar{y}|$. The conditional Entropy is

$$Entropy(j|\bar{y}) = \frac{|y_j|}{|y|} log \frac{|y_j|}{|\bar{y}|} \quad - \quad (6.2)$$

The entropy gives better result when it is applied in small or medium number of values. When large values are used for entropy, it produces minimum information or low quality of result. Due to this problem, gain value tends to loss the overall gain value. To overcome the limitations of C 4.5, Gain ratio is used by dividing the training sets into two based on its test attributes. To choose an attribute, the gain ratio considers both the number and size of branches.

The gain is defined as,

$$Gain(\bar{y}, j) = Entropy\big(\bar{y} - Entropy(j|\bar{y})\big) \quad - \quad (6.3)$$

The aim of the gain ratio is to maximize the gain by dividing the overall entropy due to split argument $\bar{y}$ by value j.

The above discussed entropy in C 4.5 has some limitations and they are listed below:

- Gives poor results when a large number of distinct values are used by both continuous and discrete attributes.
- There is no particular technique for predicting information gain. So the information gain is identified after attribute value is generated. The system performs poor due to the mismatch of attribute values.
- The system fails when the number of attributes is higher than the information gain.
- The decision tree entropy results in uncertainty.
- Difficult to select the succeeding attribute value, if the previous attribute value is less and it leads to unconditional selection of attributes.
- When same valued attributes are used in decision tree generation, the split up is a complex task which ends with unbalanced trees.

To overcome the above limitations in C4.5, Pearson's Correlation Coefficient is used. Pearson Correlation coefficient solves the unconditional selection of attributes, and accuracy based on mismatch of the attribute value and uncertainty in entropy. The next section discusses the Pearson's Correlation Coefficient in detail.

6.2.2.3. Pearson's Correlation Coefficient

Pearson's correlation is used because it implements the linear relationship to identify the relation between the variables which is a statistical model that obtains optimal results. Pearson correlation evaluates the linear connections by assessing the weight and route of two variables. The connections of those variables are denoted from -1 to +1. Correlation Coefficient will receive +1 as maximum value if the variables are related by increasing relationship, else if the variables are related by decreasing relationship, the Correlation Coefficient will receive a minimum value -1. Moreover, a value 0 will be received if the variables are not related. When the Correlation coefficient value exceeds 0.8, then it is termed as strong correlation.

If X and Y are any two variables, the Pearson correlation coefficient is defined as:

$$r_{xy} = \frac{\sum XY - \frac{(\sum X)(\sum Y)}{n}}{\left|\left(\sum X^2 - \frac{(\sum X)^2}{n_x}\right)\left(\sum Y^2 - \frac{(\sum Y)^2}{n_y}\right)\right|} \quad - \quad (6.4)$$

Entropy can be defined as

$$E(S) = \frac{\sum XY - \frac{(\sum X)(\sum Y)}{n}}{\left|\left(\sum X^2 - \frac{(\sum X)^2}{n_x}\right)\left(\sum Y^2 - \frac{(\sum Y)^2}{n_y}\right)\right|} \quad - \quad (6.5)$$

Can be written as

$$E(S) = \frac{\sum XY - \frac{(\sum X)(\sum Y)}{n}}{\sqrt{(SS_X)(SS_Y)}} \quad - \quad (6.6)$$

where,

- ΣX refers to the sum of all X
- ΣY refers to the sum of all Y
- $\sum X^2$ denote to the sum of squares of each X
- $\sum Y^2$ denotes to the sum of squares of each Y
- $\sum XY$ refers the sum of the product of X and Y
- *n* refers to the number of data pairs
- n_x refers to the *x* data pairs
- n_y refers to the *y* data pairs
- *S* refers to the entropy of set of items
- S_X refers to the entropy of set of items of X
- S_Y refers to the entropy of set of items of Y

6.2.3. Containment of Internet Worms

The malicious IP addresses detected from the anomalous traffic are blocked and containment is performed using Blacklists and Whitelists techniques. The Blacklist contains the collection of distinct IP addresses which sends undesired traffic to the destination and the Whitelist contains the collection of distinct IP addresses which send legitimate traffic to the destination. So the addresses in the network would either belong to Blacklist or Whitelist but not to the both sets.

All the addresses in both Blacklist and Whitelist are assigned with weight '*w'* according to its priority. If the address is of Blacklist, it is arranged with negative weight and referred as blacklisted. If the address is of Whitelist, it is assigned with positive weight and referred as not blacklisted. So, on the basis of the priorities of the addresses, anomalous traffic is blocked. The algorithm applied for malicious IP addresses blocking is given in table.6.3.

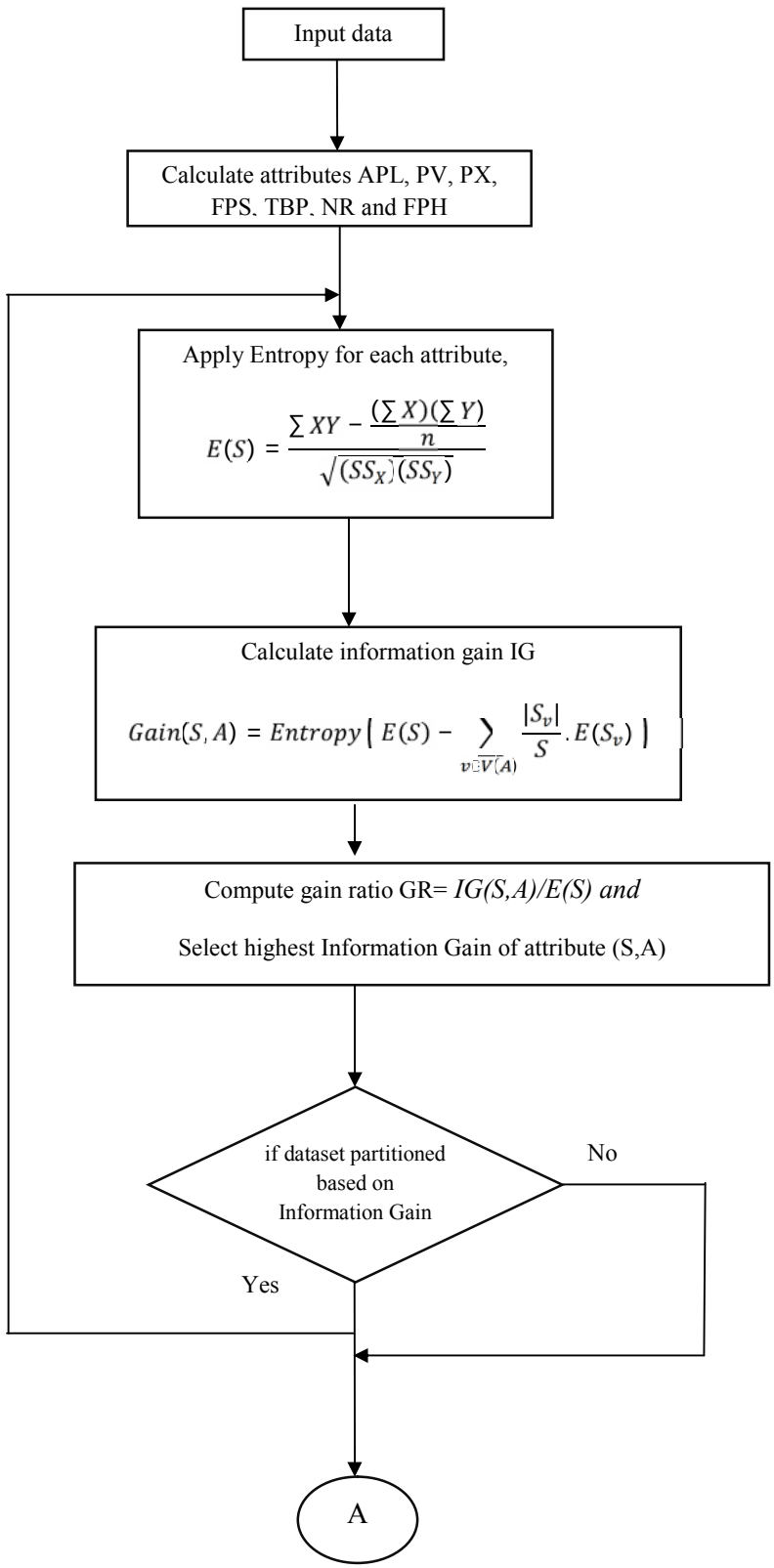
Table.6.3. Malicious IP Address Containment Algorithm

```
If (IP address is non-malicious)
    White list= IP address
    IP address = +ve weight
Else
    Blacklist = malicious IP address
```

The various steps and the methods used for detection and containment of second channel based propagation scheme and monomorphic format worms are discussed in detail above.

6.3. Flow diagram of the Proposed Contribution Three – ECB Method

The proposed method ECB detects the illegal traffic flows using **E**nhanced **C** 4.5 and containment using **B**lacklist method. The flow diagram of the proposed contribution three is shown in figure.6.2 below.


Input data
Calculate attributes APL, PV, PX, FPS, TBP, NR and FPH
Apply Entropy for each attribute,
$E(S) = \frac{\sum XY - \frac{(\sum X)(\sum Y)}{n}}{\sqrt{(SS_X)(SS_Y)}}$
Calculate information gain IG
$Gain(S, A) = Entropy\left[E(S) - \sum_{v \in V(A)} \frac{|S_v|}{S} . E(S_v) \right]$
Compute gain ratio GR= *IG(S,A)/E(S) and*
Select highest Information Gain of attribute (S,A)
if dataset partitioned based on Information Gain
No
Yes
A

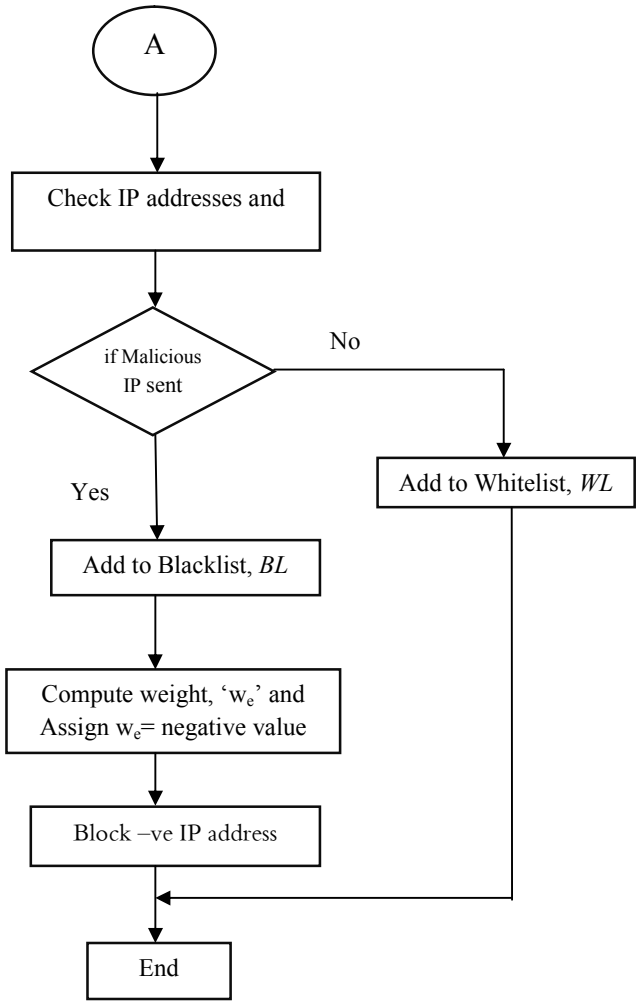


Figure.6.2. Flow diagram of the Proposed Contribution Three – ECB Method

6.4. Steps involved in the Proposed ECB Method

The steps used for the detection of illegal traffic flows and containment of its entry into network are discussed below:

Step 1: *Initialize Botnet trace B as root node N and analyze it by measuring and comparing the attribute vector A.*

Step 2: *Apply entropy (S) for each attributes* $in\,attribute\ vector\ A$

$$E(S) = \frac{\sum XY - \frac{(\sum X)(\sum Y)}{n}}{\sqrt{(SS_X)(SS_Y)}}$$

Step 3: *Calculate information gain IG*

$$Gain(S, A) = Entropy\left[E(S) - \sum_{v \in V(A)} \frac{|S_v|}{S} . E(S_v) \right]$$

Step 4: *Compute gain ratio (GR),* $GR = IG(S, A)/E(S)$ *and select attribute A with highest Information Gain (S,A) and split the dataset B into two branches depends on attributes with highest Information Gain.*

Step 5: *The dataset is partitioned by computing the entropy, information gain and gain ratio until the leaf node L is achieved.*

Step 6: *Check IP addresses that send illegal traffic and classify into malicious and non-malicious. The leaf node L contains malicious and non-malicious*

Step 7*: If (IP address is non-malicious)*

Then Assign White list= IP address and IP address = +ve weight

Else 1

Assign Blacklist = Malicious IP address and Malicious IP address = -ve weight

Block IP address

End if

Six steps discussed above are followed for detection and containment of illegal traffic through the second channel transmitting through TCP and UDP protocols. The algorithm proposed is discussed below.

6.5. Pseudo code of ECB Method

The algorithmic procedures applied for detecting the malicious traffic and containment of those detected traffic is shown in table.6.4.

Table.6.4. Pseudocode of ECB Method

```
Input : Botnet dataset D
Output : Block malicious address
Begin
If (dataset D is input)
 Check for base cases
     If (samples S in the list  same class) then
     Create a  node to choose that class
            Compute entropy E for each attribute A(i)
            Compute Information Gain IG(D,attribute list)
            Compute gain ratio=
     Let a  be the attribute with the highest information gain
     Create a decision node that splits on a.
     If (D partitioned into sublists based on a)
          i=1
          do
          {
              Compute entropy E for each attribute A(i)
              Compute Information Gain IG(D,attribute list)
              Compute gain ratio= IG(S,A)/E(S)

         }while(i<= n and leaf node L is achieved)
     Else
        Classify IP address into malicious and non-malicious.
       Assign Leaf node L  c  malicious and non-malicious IP address.
    End if
         Blacklist = malicious IP address
        Malicious IP address = -ve weight
       Block IP address
   End if
 End if
End
```

The section above briefly discussed the various steps involved in contribution three using the proposed ECB method. The pictorial representation through flow diagram, steps involved and pseudo code of the proposed contribution three are also explained. The performance of the proposed ECB method is evaluated using the various parameters and the results obtained are explained in the section below.

6.6. Experimental Setup and Results

For experimentation, traces are downloaded from the web sources and are stored in MYSQL database. Then they are analyzed by measuring and comparing the attribute vector which contains collection of network flows. After analyzing, the incoming flows are detected and classified into malicious and non-malicious traffic traces. The classified malicious traffic flows are blocked. The implementation of the above methods is done using JAVA.

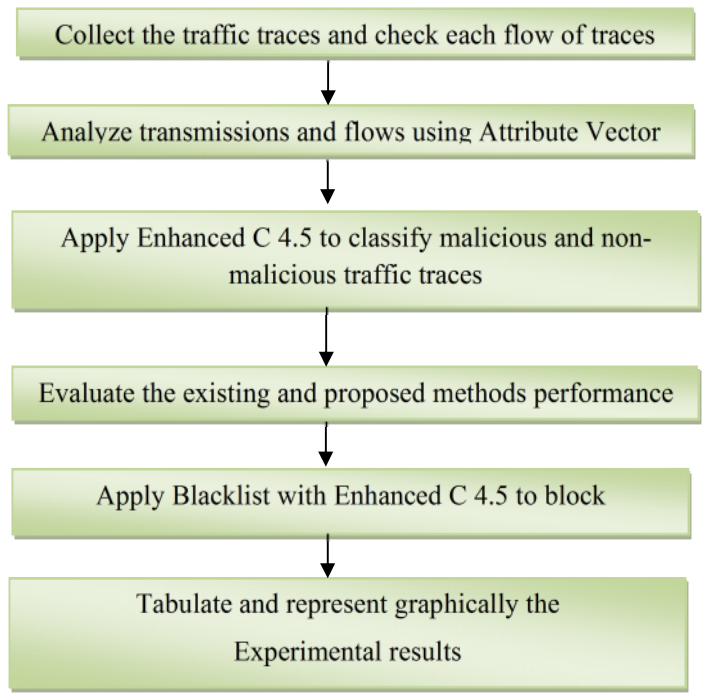


Figure.6.3. Experimentation Methodology for Contribution Three

In the proposed contribution three, the experiment uses the botnet trace as dataset which is collected from the website http://www.uvic.ca/engineering/ece/isot/datasets/index.php, which contains three various traffic data traces. This dataset contains a total of 5, 03,219 records with both malicious and non-malicious traffic data. The dataset contains the traffic of the infected hosts IP addresses. The traffic traces are from web, email and streaming media. The traces of storm and Waledac botnet worms are used for experimentation. A sample dataset is shown in Annexure III. The proposed ECB method is implemented and the figure.6.3 shows experimentation methodology.

The proposed **C** 4.5 with **P**earson co-efficient **C**orrelation (**CPC**) are compared with the existing Reduced Error Pruning Tree (REP Tree) [14] method. The programs are experimented using the collected dataset from the Honeynet Project, Ericsson Lab and Lawrence Berkeley National Laboratory. All the three datasets are merged and stored in a single file. The file is extracted through Wireshark and is evaluated. The experiments are performed and the results obtained are tabulated and shown in table.6.5 and figures 6.4 to 6.8.

Proposed **C** 4.5 with **P**earson co-efficient **C**orrelation (**CPC**) method for detection is evaluated based on the parameters such as Memory Utilization, Time Consumption, Precision value, Recall value and Accuracy.

Table.6.5. Performance Comparison of Detection Results for Existing and Proposed CPC Method

Parameters	Existing REPTree (David Zhao et al.)	Proposed CPC	%of Improvement
Memory Utilization (mb)	2250	**1200**	**9.46**
Time Consumption (ms)	100	**60**	**40**
Precision Value (%)	67	**81**	**17.28**
Recall Value (%)	79	**88**	**10.22**
Accuracy (%)	65	**76**	**14.47**

From the above table.6.5, it is observed that the proposed **C** 4.5 with **P**earson co-efficient **C**orrelation (**CPC**) method provides better performance compared to that of the REP Tree.

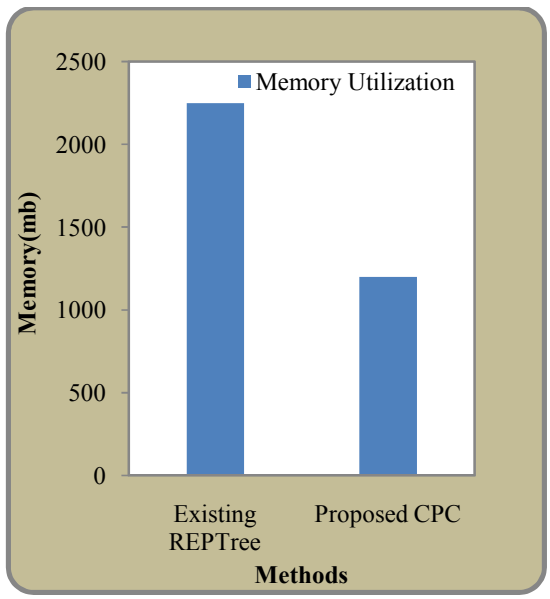


Figure.6.4. Comparison of Memory Utilization for Contribution Three

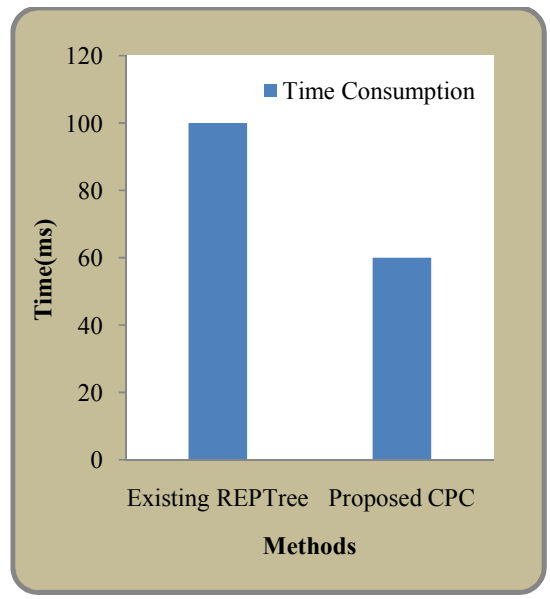


Figure.6.5. Comparison of Time Consumption for Contribution Three

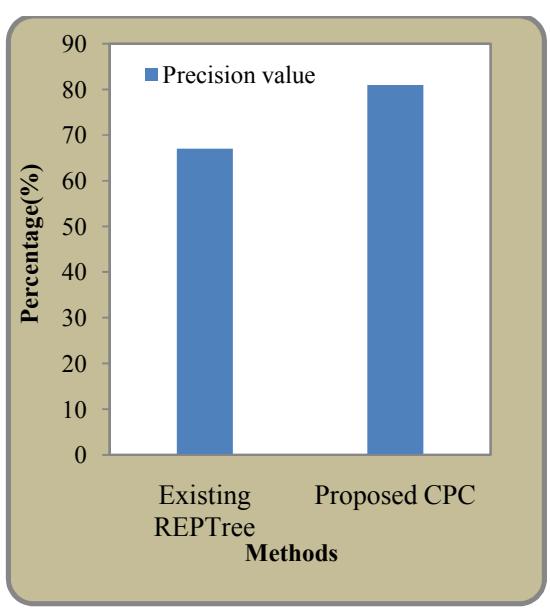


Figure.6.6. Comparison of results for Precision Value for contribution three

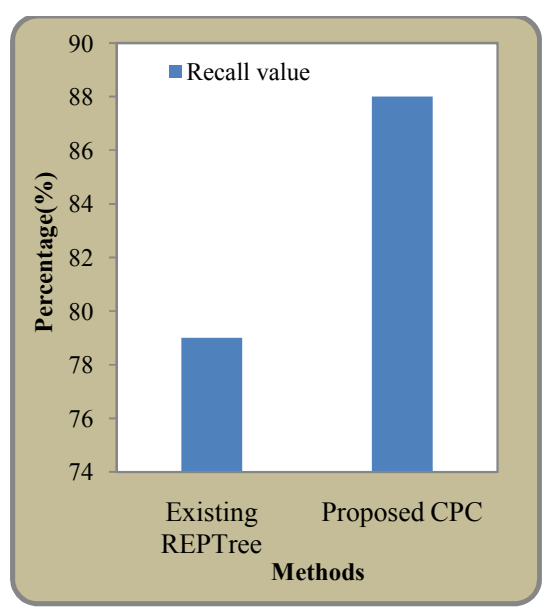


Figure.6.7. Comparison of results for Recall Value for contribution three

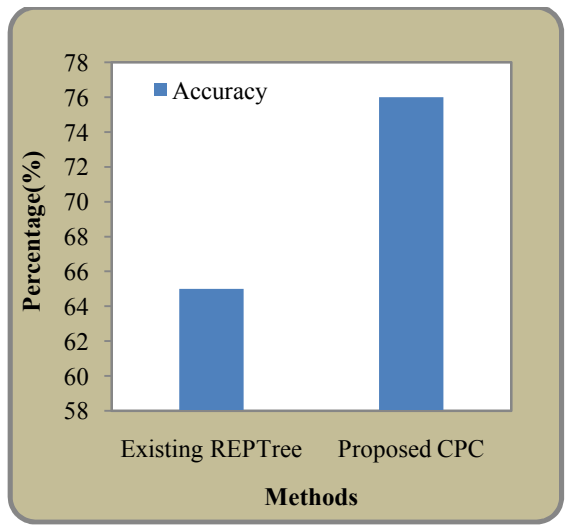


Figure.6.8. Comparison of results for Accuracy for Contribution Three

It is observed from the figures.6.4 to 6.8, that the proposed CPC method on the average has achieved minimum memory utilization with 1200mb and time consumption of 60ms. The accuracy attained by the proposed method is also improved by 14.47%.

The proposed Containment technique is evaluated using Detection Rate and Containment Rate and the result is shown in figure.6.9. It is very important to note that when the detection rate is 73.45%, the containment rate is 73.45%. The detected Malicious IP addresses are blocked using the proposed ECB method. This shows that the containment rate is 100%, that is, all the detected malicious IP addresses are blocked.

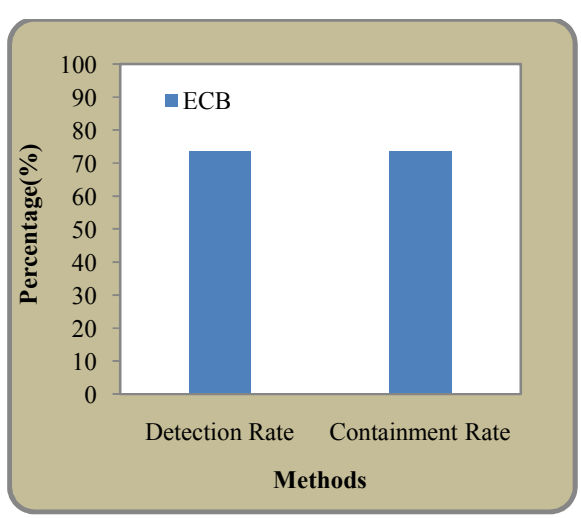


Figure.6.9. Results for Containment due to Contribution Three

The Malicious IP addresses detected are completely blocked using the ***ECB*** Method over the time period of 20 ms.

6.7. Chapter Summary

In this chapter, second channel propagating worm creating malicious traffic flows between source and destination IP addresses are identified and illegal traffic exchanged from unused addresses are detected. In analysis step, network traffic flows are analysed based on attribute vectors. In detection and classification step, enhanced C4.5 with Pearson's Correlation Coefficient is used to classify the network flow through Botnet and the method overcomes the limitations of existing Reptree approach. The proposed CPC provides better detection rate of illegal traffic from IP address. The ECB method provides better blocking of detected IP address.

The experimental results show the detection accuracy of 76% during detection and containment rate of 100% over 20ms of time. In other words all the detected Illegal traffic creating IP addresses are blocked in containment step. The detection accuracy is improved by 14.47% compared to the existing Rep Tree method. There are few worms that perform their transmission through the closed ports and cause failures of connection attempts. To detect those connection attempt failures, Contribution four is proposed and explained in chapter 7.