particular window is either an attack window or not.

Function determineThresholdProbability(S, N, TF, t)

Input: Stream S, Window size for stream N, Traffic statistics TF, Error proportion t percent

Output: Threshold probability P_t for S

```
1. Seed Rng ()
2. parray[] ← 0
   // Grouping Learning traffic
3. Split traffic TF uniformly at random into 10 groups G1 to G10
4. for i = 1 to 10 do
5.     G ← TF − Gi
6.     B ← Gi
       /* Learn from 9/10th of traffic
7.     Stemp ← TCPTrain (SF, G)
       // Compute probability of remaining 1/10th of traffic
8.  for every window W in B do
9.     P ← determineProbability (W, Stemp)
10.   add P to parray;
11.  end
12. end
    //Sieving lower probabilities
13. result ← (t/100 * len(parray) + 1)th smallest element of parray
14. return result
```

5.3.7 Deploy procedure

This module tests the input traffic and computes if a window is clean or not.

Function Deploy(S, IT, AWC)

Input: Stream S, Input Traffic IT, Abnormal Window Count (AWC)

```
1. A ← 0;
2. for every packet window W in T do
3.     P ← determineProbability (W, S)
4.    if P < Threshold probability of S then
5.          increment A
```