4. BEHAVIOUR TRUST MODEL

4.1 INTRODUCTION

Predicting the future action, based on the past behaviour requires the system to be computationally evaluated from its earlier experience. If the behaviour is predictable, then the risk associated with a system can be visualized, or else, it may risk the environment, and create a chain reaction to destabilize the organizational security system. Therefore, the uncertainty and context-dependent nature of trust can be solved through the ideology of behaviour assessment. Hence the behaviour of the cloud entities namely the services and its user’s needs to be analysed directly with the metrics at different time instances using Behaviour Trust Model (BTM). In this section, there are two phases. First, trust decision ranks the attributes and measures them, and second, it computes a trust value.

4.2 PROCESS INVOLVED IN BTM

The sequences of steps involved in the behaviour trust model are shown in Figure 4.1. The broker who is responsible for collecting the evidence and computing the trust score starts the process of assessing the client’s capability. Once the check is cleared, then the service gets initiated at the user’s desk. Then during the usage of the application, the broker...
measures various parameters to evaluate the service behaviour and simultaneously collects the user’s behaviour. The broker evaluates the trust score based on the TCAM Trusted Services Evaluation Algorithm (TTSE) and TCAM Trusted Client Evaluation Algorithm (TTCE). Finally, the trust score is categorized and sent to Attestation Process Model.

4.3 CLIENT CAPABILITY CHECK

The BTM initiates to measure basic configuration of the clients and finds whether the client device is capable enough to get initiated.

4.3.1 Attributes for capability check

1. Network Speed (NS): Services rendered by cloud primarily depends on the good bandwidth connectivity without any fluctuation in data rate. The required network speed is calculated based on the size of the application instantiated at the user desk. The speed is reported for availability, at every time interval (T).

2. Available Memory (AM): Every application has certain memory requirements, and this is calculated based on the application’s instance size at a frequency (T).

3. Firewall Protection (FP): A system protected with a firewall can prevent known malware from being executed. So, an assurance of protection can be guaranteed. Hence presence or absence of firewall helps the security of the system in force.

4. Trusted Platform Module (TPM): As previously explained in section 2.4, TPM plays a vital role in securing key information of the device through its hardwired Platform Configuration Register (PCR). Therefore higher weightage must be ascertained for the presence of TPM.

5. History of MisTrust (HMT): Previous history of misbehaviour is considered to be the most vulnerable attributes taken into account for the present instance. Hence to consider the HMT attributes, the
earlier value is categorized into, revocable trust and irrevocable trust. The revocable trust provides a chance for the user to improve upon his trust score while irrevocable trust entities are removed and marked as vulnerable.

4.3.2 Weightage assignment for TCC attributes

Now, these metrics must be weighed using AHP ranking by comparing two attributes on a scale of 1 to 9. The comparison factor was considered after receiving opinions from expert members of the domain. A decision matrix arrived using AHP is in Table 4.1, along with its corresponding weightage and priority ranking.

Table 4.1: Decision matrix for CCA using AHP

<table>
<thead>
<tr>
<th>Client Capability Attributes (CCA)</th>
<th>NS</th>
<th>AM</th>
<th>FP</th>
<th>TPM</th>
<th>HMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS</td>
<td>1</td>
<td>0.5</td>
<td>0.333</td>
<td>0.25</td>
<td>0.2</td>
</tr>
<tr>
<td>AM</td>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td>0.25</td>
<td>0.2</td>
</tr>
<tr>
<td>FP</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0.333</td>
<td>0.2</td>
</tr>
<tr>
<td>TPM</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>0.333</td>
</tr>
<tr>
<td>HMT</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Weights (W)</td>
<td>0.057</td>
<td>0.079</td>
<td>0.121</td>
<td>0.254</td>
<td>0.487</td>
</tr>
<tr>
<td>Priority Ranks</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Principal Eigen Value = 5.2 and Consistency Ratio (C.R) = 0.04

4.3.3 TCAM Client Capability (TCC) algorithm

Having met the criteria of assigning relative weights with C.R < 0.1, and Eigen value of 5.2, the process of measuring the attributes begins as per the TCAM Client Capability (TCC) algorithm. It has two algorithms, the first is Attribute measurement Algorithm for TCC presented below in Algorithm 2, collects the measurements of each attribute and sends it for processing the trust score based on weights.
Algorithm 2: Attributes measurement algorithm for TCC

for CloudUser(i) {
    Get Network_Speed, Available_Memory, Firewall_Enabled, TPM_Enable, Historyof_MisTrust

1. Network_Speed()  
   Start_time T₁  
   while(service_resource) {  
      Capture End-Time Tₙ  
      Capture Resource_Size in Kbytes  
      Duration=(End_Time - Start_Time)(in Kbps) }  

2. Available_Memory()  
   Window.performance.memory.usedHeapSize (in MB)  

3. Firewall_Enabled()  
   if windows  
      execute:netsh advfirewall.show.pubic = yes  
   else if debian or openSUSE  
      execute:ufw(unix firewall) status = yes  

4. TPM_Enabled()  
   if windows  
      Microsofttmp path win32_tpm get isEnabled –initialValue  
   else if  
      Probe tpm_bios  

5. Historyof_MisTrust()  
   If Check_ID_inDatabase == true && Check_Block_Status == false {  
      then Mistrust = false;  
      Check finalTrustValue and  
      classify Historyof_MisTrust (HighlyTrusted || Trusted )  
   else Mistrust = true;  
      Check finalTrustValue and  
      classify Historyof_MisTrust (revocable || irrevocable)  
   }  
   elseif Check_ID_inDatabase == false {  
      then Mistrust = false  
      classify Historyof_MisTrust (NewUser)  
   }  
}

The second algorithm is the Trust value and rejection condition algorithm for TCC presented in Algorithm 3 below, describes the steps involved in normalizing the measurements between 0 to 1 and assigning weights to each attribute. A final trust value is calculated along with the rejection condition for eliminating the malicious clients.
Algorithm 3: Trust value and rejection condition algorithm for TCC

for CloudUser(i) { 
  Normalise Network_Speed, Available_Memory, Firewall_Enable, TPM_Enable, History_of_MisTrust

  // Network_Speed (NS)
  Calculate Best_case_Bandwidth(BW) based on App. Size
  if BW <= 1/4 Best_Case(BW) then NS = 0
  elseif BW > 1/4 && < 1/3 Best_Case(BW) then NS = 0.25,
  elseif BW > 1/3 && < 1/2 Best_Case(BW) then NS = 0.5,
  elseif BW > 1/2 && < 1 Best_Case(BW) then NS = 0.75,
  else NS = 1,
  Assign Weightage (W_{NS});

  // Available_Memory (AM)
  if Available_Memory >= Required_memory then AM = 1 else 0;
  Assign Weightage (W_{AM});

  // Firewall_Enable (FE)
  if Firewall_Enable = 1 then FE = 1 else FE = 0;
  Assign Weightage (W_{FE});

  // TPM_Enable
  if TPM_Enable = 1 then TPM = 1 else TPM = 0;
  Assign Weightage (W_{TPM});

  // History_of_MisTrust (HMT)
  Check category of previous trustValue
  for CloudUser(i) at time (t - 1) {
    if Trust_{TCC} categorised as Highly Trusted then HMT = 1,
    elseif Trusted then HMT = 0.75,
    elseif NewUsers then HMT = 0.5,
    elseif RevocableMistrust then HMT = 0.25,
    elseif IrrevocableMistrust then HMT = 0
  } 
  Assign Weightage (W_{HMT});

  // Capability Check Assessment
  find trust_Value
  Trust_{TCC} = NS * W_{NS} + AM * W_{AM} + FE * W_{FE} + TPM * W_{TPM} + HMT * W_{HMT}

  find Rejection_Condition
  if RC = 1/5^{th} of Trust_{TCC}
    reject CloudUser(i) for to access the service
  else
    start Instantiation of Service()
}
4.3.4 Trust score and rejection condition for TCC

If there are \( m \) measurement attributes for Client Capability Attributes (CCA\(_m\)) then, for each attribute, weights are assigned (\( W_{m}^{CCA} \)), the generalized trust value calculation can then be given as in equation 4.1.

\[
\text{Trust}_{TCC} = \sum_{1}^{m} \text{CCA}_m \ast W_{m}^{CCA} \tag{4.1}
\]

Having found a value for checking the capability of client devices, it is now necessary to identify a condition for rejecting the non-eligible customers. To fix the criteria, a minimum value is needed, which can be judged based on the basic requirements to run the application. The necessities considered are given below.

- Minimum network speed
- Minimum available memory to run the application
- The history of mistrust should be revocable trust.

Based on these requirements approximately 1/5\(^{th}\) of maximum \( \text{Trust}_{TCC} \) value is considered for rejection condition.

4.4 TCAM TRUSTED SERVICE EVALUATION (TTSE)

As the service is been initiated and running on the client’s device, the next step is to collect the evidences to estimate the trust for that service.

4.4.1. Attributes of service trust

1. Virtual Machine ID (VM): Every cloud service is hosted on a virtual machine at remote data center. During the usage of the application, the VM may change its physical location, generally termed as VM migration. The location change can be attributed to various reasons like resource re-allocation, overload or a routine scheduling. These changes must be informed to the users through the virtual machine identity (VM ID). This indication is a strong capability of control over
the visibility, assuring that the system is performing as per the Service Level Agreements (SLA).

2. **Data Size (DS):** It represents the size of the application being downloaded, for running the service. If there is any deviation in size, then it is an indication of irrelevant code is being embedded. This code may most likely be a threat to the service running. More than that, the service which is infested with vulnerable code can act as a malware and steal information from the users’ desk. Hence change in data size justifies the top most priority given in evaluating the trust for services.

3. **Application Response Time (ART):** The time taken to deliver the service at the earliest defines the turnaround efficiency of the service. Hence response time is an indicator of good quality which can function as a trust case to evaluate the application.

4. **Incomplete Service Initialization (ISI):** A basic attribute to identify the reliability of the service is, the number of time the service is available when requested. Hence any incomplete service initialization reduces the QoS, which in turn can affect availability.

5. **Service Completion Status (SCS):** Every cloud service used by the consumer must have gone through the process of initialization and completion. A service is said to have completed if it has been utilized properly by the client. Therefore if the service is successfully completed, then it can also be said that the service has achieved its reliability and availability. Moreover only if the service gets completed, users have the chance of giving an opinion in the form of feedback or user’s rating.

6. **Log File Stored (LFS):** Every service running on the client device, must store its log information. This log information stores various events and actions performed by the service while it is running. It helps in identifying a violation of SLA and holds the provider accountable [67]. Therefore, the service statistics guarantees the service’s readiness to face auditing challenges.
Thus based on the expert opinion and literary review the three attributes, namely, data size, service completion status, and VM identity are considered most significant, and hence the weights have to be assigned relatively higher.

### 4.4.2 Weightage assignment for TTSE attributes

The decision matrix arrived by using the AHP is given in Table 4.2, along with the weights assigned and the priority of ranking.

<table>
<thead>
<tr>
<th>Service Trust Attributes (STA)</th>
<th>VM</th>
<th>DS</th>
<th>ART</th>
<th>ISI</th>
<th>SCS</th>
<th>LF</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>DS</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>ART</td>
<td>0.5</td>
<td>0.333</td>
<td>1</td>
<td>0.5</td>
<td>0.25</td>
<td>1</td>
</tr>
<tr>
<td>ISI</td>
<td>0.2</td>
<td>0.5</td>
<td>2</td>
<td>1</td>
<td>0.333</td>
<td>3</td>
</tr>
<tr>
<td>SCS</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>LFS</td>
<td>0.2</td>
<td>0.25</td>
<td>1</td>
<td>0.333</td>
<td>0.333</td>
<td>1</td>
</tr>
<tr>
<td>Weights (W)</td>
<td>0.279</td>
<td>0.229</td>
<td>0.074</td>
<td>0.109</td>
<td>0.249</td>
<td>0.058</td>
</tr>
<tr>
<td>Priority Ranking</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

Principal Eigen Value = 6.25 and Consistency Ratio (C.R) = 0.041

### 4.4.3 TCAM trusted service evaluation algorithm

Since the consistency ratio is C.R < 0.1, the weightage assigned through AHP can be accepted. Now, the attributes are measured in real time using TCAM trusted service evaluation algorithm. It has two algorithms, where the first is Attributes measurement algorithm for TTSE, presented below in Algorithm 4.
Algorithm 4: Attributes Measurement algorithm for TTSE

for User_ID_Service(i)
{

Get VM_ID, Data_Size, Response_Time, Incomplete_Service_Initialization,
Service_Completion, Log_File_Stored.

1. VM_ID ()
   on service_initialized
   get VM_ID at time T1 and T2
   compare VM_ID@T1 == VM_ID@T2
   If VM-ID change updated.
   then VM_Status = “Ok”
   else report VM_Status = “deviation”

2. Data_Size ()
   Check page_bytes = (html).length

3. Application_Response_Time ()
   Application_Response_Time = Network Response Time +
   Transaction Response Time

   Network Response Time = Data_Size/Bandwidth +
   Round Trip Time

   Transaction Response Time = Server Response Time +
   Client Response Time

4. Incomplete_Service_Initialization ()
   if Application_Response_Time = null then ISI = true

5. Service_Completion_Status ()
   if timeSpent_onPage> No of basic operation performed
   then Service_completion=1
   else Service_completion=0

6. Log_File_Stored ()
   if update_query (parameters)
   then check sql database
   set log = true
   else log = false
}

52
The second algorithm is the Trust value and rejection condition algorithm for TTSE described below in Algorithm 5.

**Algorithm 5: Trust value and rejection condition algorithm for TTSE**

```plaintext
for User_ID_Service(i) {
    Normalize VM_ID, Data_Size, Response_Time,
    Incomplete_Service_Initialization, Service_Completion, Log_File_Stored.

    // Virtual Machine Identity(VM)
    if VM_status = “deviation”
        then VM = 0 else VM = 1
        Assign Weightage \( W_{VM} \)

    // Data_Size(DS)
    if Data_Size >= Application Code(page byte)
        then DS = 0 else DS = 1
        Assign Weightage \( W_{DS} \)

    // Application_Response_Time(ART)
    if ART > 1/5th of Average ART then ART = 1
    elseif ART is between 1/4 && 1/3 then ART = 0.75
    elseif ART > 1/3 && >1 of Avg. ART then ART = 0.5
    elseif between 2*Avg.ART to Avg. ART = 0.25,
        else > 2* Avg. ART then ART = 0
        Assign Weightage \( W_{RT} \)

    // Incomplete_Service_Initialization(ISI)
    if ISI = true then ISI = 0 else ISI = 1
    assign weightage \( W_{ISI} \)

    // Service_Completion_Status(SCS)
    if Service_Completion_Status = True,
        then SCS = 1 else SCS = 0, find Avg(SCS)
        Assign Weightage \( W_{SCS} \)

    // Log_File_Stored()
    if Log_File_Stored = true
        then LFS = 1 else LFS = 0
        Assign Weightage \( W_{LFS} \)
}
```

Trust Value for Services \( T_{QoS}(S) = \)

\[
    VM \cdot W_{VM} + DS \cdot W_{DS} + RT \cdot W_{RT} + ISI \cdot W_{ISI} + SCS \cdot W_{SCS} + LFS \cdot W_{LFS}
\]

Final Trust Value \( T(S) = T_{QoS}(S) \cdot W_{QoS} + T_{HTS}(S) \cdot W_{HTS} \)

Rejection Condition \( (RC_S) \) is deviation in Data Size

53
4.4.4 Trust value with rejection condition for TTSE

Let \( n \) be the number of attributes considered and for TTSE and the weights assigned for each attributes towards the Service Trust Attributes (STA) be \( W_n^{STA} \), then the QoS trust value for services \( (Trust_{QoS}) \) is calculated as in equation 4.2.

\[
Trust_{QoS} = \sum_{i=1}^{n} STA_n * W_n^{STA}
\]  

(4.2)

The final trust score \( (Trust_{TTSE}) \) requires the knowledge of history of trust \( (Trust_{HTS}) \) hence the trust score of earlier iterations is included [68]. However for a new service provider, the history don’t exist and therefore the trust is calculated without the aid of \( Trust_{HTS} \), and the final score would be QoS trust value only. Hence any new service has to gain its value through consistent evaluation of its historical evidences.

\[
Trust_{TTSE} = Trust_{QoS} * W_{TQoS} + Trust_{HTS} * W_{HTS}
\]  

(4.3)

where,

\[
Trust_{HTS} = \sum_{i=1}^{t-1} Trust_{QoS}
\]  

(4.4)

\( W_{TQoS} \) – is the weightage for QoS Service Trust

\( W_{HTS} \) – is the weightage for history of QoS Service Trust

Essential criteria to assess the behaviour of the service are change in data size. Though other criteria are essential, the level of damage can be severe if malware gets injected into the application. Also, the priority of write condition over reading must be taken into consideration to assess for selection or rejection cases. Since embedding vulnerable code into the application is a write condition it is considered with the highest priority for rejection. Thus the selection condition is application code must be free from malicious behaviour, which corresponds to the value \( DS = 1 \), along with its weightage. Threshold Value is \( DS * W_{DS} \), where \( DS = 1 \).

Once the trust score for service is estimated, the next step is to categorize the trust value and store it in the repository. Further, the values are sent to the next stage of evaluation, i.e., to the attestation engine.
4.4.5 Trusted service categorization

The process of categorization is based on the quantitative approach of mean, median and quartiles. Quartiles rank the data values into four equal groups and mean, and median identifies the distribution of values. The Table 4.3, list out the trust zone corresponding to the distribution.

Table 4.3: Trust categorization

<table>
<thead>
<tr>
<th>Trust Zone</th>
<th>Quantification Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Trust</td>
<td>&gt;3rd Quartile</td>
</tr>
<tr>
<td>Trusted</td>
<td>&gt;Median to &lt; 3rd Quartile</td>
</tr>
<tr>
<td>Safe Zone</td>
<td>&gt;1st Quartile to Median</td>
</tr>
<tr>
<td>Revocable Trust</td>
<td>&gt;Rejection Condition to &lt; 1st Quartile</td>
</tr>
<tr>
<td>Irrevocable Trust</td>
<td>&lt;Rejection Condition</td>
</tr>
</tbody>
</table>

The zones of trust are depicted in Figure 4.2 through a rhombus shaped visualization model. The top most vertex points to the high trusted services, while the bottom most points to the services with zero value.

Figure 4.2: Trust Zones for Services
The center occupies services which are safe however the threshold for distrusted services marks the point of rejection. Below the threshold value are those services which are irrevocable and hence would be removed from trust evaluation during the next iteration. The revocable services are that which have the minimum quality, however, has a chance to improve. Thus this visualization model can enable users and providers as well to judge how their services have been improving upon.

4.5 TCAM TRUSTED CLIENT EVALUATION (TTCE)

User Behaviour Analytics (UBA) finds a unique behavioural pattern which establishes a level of confidence for every user’s typical behaviour. Finding any significant deviations or any suspicious events can trigger appropriate policies to quarantine or block that accounts activity. The process of client evaluation henceforth is an essential criterion to access the resources of the service of cloud [69]. The assessment may be while the service is in execution or after completion. This process can effectively identify the trustworthiness of the cloud consumers.

Hence various attributes must be considered for evaluation, which is broadly classified below.

1. Trusted QoS Client Assessment
2. Trusted Authentication Client Assessment

4.5.1 Trusted QoS Client Assessment (TQCA)

The attributes considered for QoS trust assessment of client are described as follows.

1. Feedback History (FH): If the consumer is intended to use the application and finds it suitable for the recommendation, then feedback score after the completion of service must be taken. This feedback can enhance the commitment or the trust relationship with service and provider. The reputation of the client increases and his trustworthiness gets a boost during the next iteration. In our model the score of the
feedback is not the primary, rather it is whether the feedback is given or not is the focus. Hence an average history of feedback is taken to evaluate the trustworthiness.

2. **IP Address Check (IPC):** Verifies the client address for any violation in simultaneous access. Moreover, change in IP address frequently is also an indication of misbehaviour, which can be tested through a higher level of authentication. Though Single Sign On (SSO) has been implemented for the TCAM, there have been numerous Distributed Denial of Service (DDoS) attacks and IP spoofing attacks using other's Internet Protocol. Hence a good trust value must be given if this attribute is verified to satisfaction.

3. **Successful Login Attempt (SL):** Every successful authentication of a user is considered to be a case where the user is showing an interest in utilizing the benefits of the service. So, the number of successful login attempts indicates good response towards service, which in turn benefits the feedback history and reputation of other entities.

4. **SLA Acceptance (SA):** SLA binding can assure the privacy obligation associated for trusting a user. Hence every user must comply with various service level agreements imposed by providers and brokers. Violation of SLA breaches the trust relating to the entities. Further, during the event of any claims or arbitrations, the SLA policy helps to fight the rights of the entities. In the case of failure to respect the SLA, the belief associated with the client trust fails, because of which certain legal challenges need to be faced by the cloud consumer.

5. **Data Size at Client (DSC):** The application code size again plays a vital role in judging the quality of user behaviour. Any change in data size can identify the deviations of user’s unwanted code injection into the service [70]. The user may intentionally inject vulnerable codes for the purpose of altering the resource utilization, billing information or enter into the administrative domain.
6. *Users Time Spent (UTS):* Time spent with the application is directly proportional to the real intention to use that service. Hence the more the time spent, the more the service gets utilized. More the utilization guarantees that the service is meeting the objective of the user. This in turn satisfies the expectation of the user.

### 4.5.1.1 Weightage assignment for TQCA

The decision matrix arrived by using the AHP for weighing the client is given in Table 4.4, along with priority of ranking between the attributes. Since user’s misbehaviour can be judged through a change in data size, this attribute is considered to be the number 1 priority. On the other hand, user’s good behaviour can be judged based on the responsibility of giving feedback (good or bad); therefore, it is considered next in the order of priority.

Untrusted clients would hardly spend their time in using the service, so user spending time follows third, with authentication ranked fourth. Other attributes namely IP check and service agreements are rated accordingly to their level of mistrust capabilities.

#### Table 4.4: Decision matrix for TQCA attributes

<table>
<thead>
<tr>
<th>Client Trust Attributes (CTA)</th>
<th>FH</th>
<th>IPC</th>
<th>SL</th>
<th>SA</th>
<th>DSC</th>
<th>UTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FH</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>IPC</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
<td>4</td>
<td>0.333</td>
<td>0.5</td>
</tr>
<tr>
<td>SL</td>
<td>0.5</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>0.333</td>
<td>0.5</td>
</tr>
<tr>
<td>SA</td>
<td>0.166</td>
<td>0.25</td>
<td>0.25</td>
<td>1</td>
<td>0.166</td>
<td>0.2</td>
</tr>
<tr>
<td>DSC</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>UTS</td>
<td>0.5</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>Weights (W)</td>
<td>0.258</td>
<td>0.102</td>
<td>0.129</td>
<td>0.036</td>
<td>0.295</td>
<td>0.178</td>
</tr>
<tr>
<td>Priority</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Principal Eigen Value = 6.15 and Consistency Ratio (C.R) = 0.041
4.5.1.2 Trusted QoS client assessment algorithm

As the consistency ratio is 0.041, which meets the requirements, the weights are accepted for further evaluation. The Trusted QoS client assessment algorithm provides the methods for measuring the attributes based on two algorithms. The first is the attributes assessment algorithm for TQCA presented below in Algorithm 6.

Algorithm 6: Attributes assessment algorithm for TQCA

```
for CloudUser(i) {
    Get Feedback History, IP_Address_Check, Successful_Login_Attempt, SLA_Acceptance, Data_Size_Client, Users_Time_Spent

    1. Feedback_History ( )
       if feedback = true
           then count feedback ++

    2. IP_Check ( )
       Check IP_address_of_UserID
       For UserID(i) at time t1 is IP = current IP address
       If UserID(i) at Time T2 is IP2
           then check if IP2== IP1
           then check location for deviation
           else check
           if IP2UserID(i) ==IP1UserID(i)

    3. Successful_Login_Attempt ( )
       if Authentication for CloudUser(i) is success
           then
               count Successful_Login_Attempt ++

    4. SLA_Acceptance ( )
       If SLA_Acceptance is true
           then allow service
           else report failure to comply

    5. Data_Size_atClient ( )
       Check pagebytes after time \( t = (html).length \)

    6. Users_Time_Spent ( )
       Set Idle_Duration_InSeconds(30)
       initiatePage(currentPage)
       Start_Time
       while window.onload
           timeSpent=time.getTime_On_CurrentPage_inSec()
}
```
The second algorithm is the Trust value assessment algorithm for TQCA as presented below in Algorithm 7.

**Algorithm 7: Trust value assessment algorithm for TQCA**

```plaintext
for Cloud_User(i) {
    Normalize Feedback_History, IP_Address_Check, SLA_Acceptance, Successful_Login_Attempt, Data_Size_Client, Users_Time_Spent.

    // Feedback_History (FH)
    FH = No_of_times_Feedback_Given / No_of_times_Successfully_completed the service
    Assign Weightage W_{FH}

    // IP Check(IPC)
    if IP2UserID(i) == IP1UserID(i) then IPC = 1
    else IPC = 0
    Assign weightage W_{IPC}

    // Successful_Login_Attempt(SL)
    if Log_In for >=1 to <=5 then SA = 0.25;
    else if Log_In >5 to <10 then SA = 0.5;
    else SA = 1
    Assign weightage W_{SL}

    // SLA_Acceptance (SA )
    if SLA_Check_Status = true then SA =1
    Assign weightage W_{SA}

    // Data_Size_Client (DSC)
    if Size_of_Data > Application_Code then DSC = 0
    else DSC = 1
    Assign weightage W_{DSC}

    // User_Time_Spent (UTS)
    if Time_spent <30 seconds then UTS = 0,
    elseif Time_spent >=30 && <=100 then UTS = 0.5
    else Time_spent >100 then UTS = 1;
    Assign Weightage W_{UTS}

    QoS_Trust_Score = FH \times W_{FH} + IPC \times W_{IPC} + SL \times W_{SL} + SA \times W_{SA} + 
    DSC \times W_{DSC} + UTS \times W_{UTS}
}
```
4.5.1.3 Trust value for TQCA

Let \( p \) be the number of attributes considered for Client Trust Attributes (CTA), and the weights assigned to each attribute is \( W_p^{CTA} \), then the QoS trust value for client \( (\text{Trust}_\text{QoS}_{TC}) \) can be as given in equation 4.5.

\[
\text{Trust}_\text{QoS}_{TC} = \sum_{1}^{p} CTA_p * W_p^{CTA}
\]  

(4.5)

Since, QoS evaluation is one of the parameters, the other parameters being the authentication, formulated in the following section.

4.5.2 Trusted Authentication Client Assessment (TACA)

The stage 2 of client assessment identifies the deviations in the authentication process. The necessity to judge the cloud consumer’s behaviour arises from the well-known mechanism of Identity Management (IdM) and access control [71]. Unusual patterns or anomalies are an indication of compromised credentials or malicious activity. These activities can be monitored using a threshold. For example, if a user has frequently failed login attempts during a short period of time then it is a security event worth alerting.

A baseline behavioural pattern must be in force for every organization, which can create a fingerprint like equivalent for each and every user’s activity. Once the user’s activity begins to deviate significantly towards an abnormal and untrusted behaviour then a risk rating can be elevated. This risk rating is nothing but the trust value of the client that triggers appropriate strategy to be followed to quarantine the account’s activity. Finding such behavioural patterns for authentication can help to identify situations where the client’s account may have been compromised. In case if the user’s trust is good and genuine then the finding may alert the user that the account has been hijacked. Therefore our Trusted Authentication Client Assessment (TACA) module derives a formula based on trial and error approach. For login evaluation, an enforcement behaviour policy is defined for the number of attempts to be made within a specified time limit.
4.5.2.1 Enforcement behaviour policy for login attempts

Every user is monitored for his login attempts made during a period of 24 hours. To establish trust, certain enforcement policies must be adopted as a rule. The conditions are framed based on the maximum number of time a user could attempt to log in successfully. For this case, if 5 minutes is the average time taken to use the cloud application, then for 24 hours the maximum number of attempts would be 288. Considering this scenario the behaviour policy for a number of login attempts would be based on the given formula in equation 4.6, which is adopted through trial and error method.

\[
\text{Trust}_{\text{LA}_{\text{TC}}} = \frac{(a^2 - b^3) * 0.3}{2a} \quad \{ 0 < a < 288 \}, \quad 0 < b < 12
\]  

(4.6)

where

- \( \text{Trust}_{\text{LA}_{\text{TC}}} \) - is the Trust value for login attempts
- \( a \) - is the number of successful login attempts
- \( b \) - is the number of failed login attempts

The following policy is being applied for evaluating the trust score.

1. Any attempt to gain access by trying out the wrong password for >5 times continuously are warned.
2. After 8th attempt, the user is blocked for 12 hours.
3. Step-wise increase and exponential decrease policy to punish bad behaviour.

4.5.3 Final trust evaluation for TTCE

The trust value for TTCE is generated as in equation 12, through the combined value of the following trust scores.

1. \( TQCA \): Trusted quality client assessment provides a value that is currently measured and evaluated. Therefore the present service run by the customer for which our aim to develop the model is given the highest weightage.
2. **TACA**: Trusted authentication client assessment judges the client’s ability to login without any malicious intent.

3. **TCC**: Trusted capability check identifies the user’s device for trustworthiness. This attributes evaluated at the initial stage is considered because; the behaviour of the client can be gained to a certain extent by judging the physical device.

4. **HTC**: History of trust calculated during all the previous iteration for trust during time period (1 to t-1). This value derived using equation 4.7, is essential for understanding the past behaviour of client. However for a new user, the HTC would have no value and has to gain trust through subsequent evaluation only.

   \[
   \text{Trust}_{TTCE} = \text{Trust}_{QoS} \times W_{TQC} + \text{Trust}_{LA} \times W_{TLC} + \text{Trust}_{TCC} \times W_{TCC} + \text{Trust}_{HTC} \times W_{HTC} 
   \]

   where,

   \[
   \text{Trust}_{HTC} = \sum_{1}^{t-1} \text{Trust}_{QoS} 
   \]

   The weightage calculated using AHP for the client trust attributes are given in Table 4.5. Based on this, the clients are categorized using the rhombus shaped visualization model similar to Figure 4.2 as trusted or untrusted and if trusted, at what scale does the trust progresses.

<table>
<thead>
<tr>
<th>Final Trust Attributes</th>
<th>TQCA</th>
<th>TACA</th>
<th>TCC</th>
<th>HTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weightage</td>
<td>0.421</td>
<td>0.143</td>
<td>0.151</td>
<td>0.286</td>
</tr>
<tr>
<td>Priority Ranking</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Overall the Behaviour Trust Model computes the trust value for the two most important entities namely, the cloud service and the cloud consumers based various quality attributes and security attributes. Having given a generalized model, next our objective is to implement the model in a real-time environment. The following subsection presents the implementation of an image processing application in the cloud environment.
4.6 IMPLEMENTATION

OpenStack implementation of the TCAM justifies the realistic behaviour of the cloud entities. Therefore, the image processing application that is deployed, is tested with cloud consumers who are actually in need of that application. Hence the SaaS was deployed in the public internet domain with a static IP address of 122.165.134.81, during the evaluation period of around 30 days. Totally around 250 users consumed the service, of which around 37 are specifically identified to be monitored for the research work.

4.6.1 Features of software service

The software service developed is an image processing application. The need for selecting the choice of image processing as the test case is explained in section 7.2.3. The application is named as TCAM Image Processing Service (TIPS), which is developed to include multiple services with trust migration facility.

Figure 4.3: Screenshot of the actual image processing application

The features of the application TIPS as depicted in the image of Figure 4.3 are as follows.

1. Developed fully in open source technology platform using PHP and Python languages.
2. Uses Component Based Application Integration to get the service dynamically from different providers.

3. Uses Single Sign On (SSO) to overcome identity management.

4. The user must select a minimum of 2 services for launching the application to evaluate the cooperative nature of the providers.

5. Totally eight cloud service providers offer their services of which 5 are image processing and three for storage service.

6. The size of the test case application is around 20Kbytes, making the initiating process quick.

7. Highlights trust score of each service to the users before selecting; this enables to increase the reputation and trusted services.

4.6.2 Attacker model and assumptions

The SaaS which has been used as the test case, must identify the possibilities of attacks for making the environment un-trustable, because, it needs to be analyzed for its attack resistance capability and performance. Some of the assumptions are listed below.

- An insider who has malicious intents to disengage the system - malicious insider attack - is tested by engaging few users to login through different credentials. This attack was identified through TQCA algorithm by cross validating the IP, login attempts and its locations.

- Malware injection attacks are scripted by inserting an additional piece of code that can able to access administrative details. The data size metrics identifies the change and reports the possibility of an attack.

- Out of the eight providers listed, 1 or 2 can misbehave by not cooperative with trustable service leading to suppress the functioning of others (Collusion attack).

- DoS attack is modelled by performing known users to make multiple requests for service through an automated scripted code.

Thus to mitigate these attack possibilities our TCAM implement the three stage layered approach and identifies trusted and untrusted entities.
4.6.3 Results and observation for TCC

The first step is to find the capability of the client device for its trustworthiness based on its configuration parameters. Of the five configurations, the network speed and available memory are primary for getting the service to start. Since our TIPS are of 20Kbytes in size, the memory availability is determined at >=10MB as the core requirement, and network speed can be specific as in Table 4.6.

The following tables provide the normalization of values between 0 - 1 for the 37 sampled consumers along with their distribution.

**Table 4.6: Network speed requirement for TIPS and distribution**

<table>
<thead>
<tr>
<th>Network Speed in Kbytes</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 14.4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>14.4 – 19.2</td>
<td>0.25</td>
<td>2</td>
</tr>
<tr>
<td>19.2 - 28.8</td>
<td>0.5</td>
<td>18</td>
</tr>
<tr>
<td>28.8 – 57.6</td>
<td>0.75</td>
<td>3</td>
</tr>
<tr>
<td>&gt;57.6</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

**Table 4.7: Available memory distribution**

<table>
<thead>
<tr>
<th>Available Memory</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;10MB</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>&gt;10MB</td>
<td>1</td>
<td>35</td>
</tr>
</tbody>
</table>

**Table 4.8: Firewall enabled distribution**

<table>
<thead>
<tr>
<th>Firewall Enabled</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Enabled</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>Enabled</td>
<td>1</td>
<td>21</td>
</tr>
</tbody>
</table>

**Table 4.9: TPM enabled distribution**

<table>
<thead>
<tr>
<th>TPM Enabled</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Enabled</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>Enabled</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 4.10: History of mistrust distribution

<table>
<thead>
<tr>
<th>History of Mistrust</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Irrevocable Trust</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Revocable Trust</td>
<td>0.25</td>
<td>4</td>
</tr>
<tr>
<td>New User</td>
<td>0.5</td>
<td>3</td>
</tr>
<tr>
<td>Trusted</td>
<td>0.75</td>
<td>8</td>
</tr>
<tr>
<td>Highly Trusted</td>
<td>1</td>
<td>19</td>
</tr>
</tbody>
</table>

Based on the above distribution, the trust value can be categorized according to TCC algorithms and quantification value in Table 4.3. As per the algorithm, the rejection condition is set as 1/5th of the maximum, which is 0.2.

Table 4.11: TCC distribution statistics

<table>
<thead>
<tr>
<th>Distribution Statistics</th>
<th>Trust Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.014</td>
</tr>
<tr>
<td>Max</td>
<td>1</td>
</tr>
<tr>
<td>Mean</td>
<td>0.560</td>
</tr>
<tr>
<td>First Quartile</td>
<td>0.474</td>
</tr>
<tr>
<td>Median</td>
<td>0.595</td>
</tr>
<tr>
<td>Third Quartile</td>
<td>0.703</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.227</td>
</tr>
<tr>
<td>Variance</td>
<td>0.051</td>
</tr>
</tbody>
</table>

Table 4.12: Trust categorization for TCC

<table>
<thead>
<tr>
<th>Trust Zone</th>
<th>Quantification Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Trust</td>
<td>&gt;0.703</td>
<td>6</td>
</tr>
<tr>
<td>Trusted</td>
<td>&gt;0.595 to &lt;0.703</td>
<td>9</td>
</tr>
<tr>
<td>Safe Zone</td>
<td>=&gt;0.474 to &lt;=0.595</td>
<td>11</td>
</tr>
<tr>
<td>Revocable Trust</td>
<td>&gt;0.2 to &lt;0.474</td>
<td>8</td>
</tr>
<tr>
<td>Irrevocable Trust</td>
<td>&lt;0.2</td>
<td>3</td>
</tr>
</tbody>
</table>

Thus, it can be seen from the final trust evaluation for capability check that, there have been totally three users whose trust score are rejected. There are eight users with revocable trust value, who must be monitored as soon as the service gets initialized.
4.6.4 Results and observation for TTSE

The service gets started at the user's desk. The BTM evaluates the attributes and performs the assessment as per the TTSE algorithm. The three most important attributes namely the data size, VM identity and service completion status must be carefully analyzed. Since the data size of our TIPS is 20Kbytes, the data size violation may be considered with a 5% deviation at 21Kbytes. In Figure 4.4, the VM identity is presented visually for the application as an indication towards control over the visibility of virtual machine.

Figure 4.4: VM identity number

The following tables show the distribution of the users after being normalized their measurements in the range of 0 to 1.

Table 4.13: VM identity distribution

<table>
<thead>
<tr>
<th>VM Identity</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change in VM</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>No Changes in VM</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Since the implementation of the cloud is set up on a reliable server with total control, the chance of VM attacks is extremely low. Hence all the VM confirm to the identity evaluation norms as in Table 4.13.

Table 4.14: Change in data size distribution

<table>
<thead>
<tr>
<th>Data Size</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 21 Kbytes</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>&gt; 21 Kbytes</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>

Next, the change in data size is listed in Table 4.14. Since the 34 consumers are sampled from 250 general users, for the purpose of the case study, few are prompted to perform attacks to change in data size. The BTM
detected all those who have attempted to modify the downloaded application size. The response time depends on the bandwidth, and its distribution is given in Table 4.15 as per the average response time which in our case is evaluated as 2300ms.

<table>
<thead>
<tr>
<th>Application Response Time</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;1/5th Average ART (AART*)</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>&gt;1/4 to &lt;1/3 of AART</td>
<td>0.75</td>
<td>9</td>
</tr>
<tr>
<td>&gt;1/3 to &lt;1 of AART</td>
<td>0.5</td>
<td>9</td>
</tr>
<tr>
<td>&lt;2*AART to AART</td>
<td>0.25</td>
<td>3</td>
</tr>
<tr>
<td>&gt;2*AART</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Though there are only three users, who have a poor response time the incompletion service initialization status is nil. Based on the minimum time spent with the service, i.e., 180 seconds, the service completion status is distributed accordingly in Table 4.16.

<table>
<thead>
<tr>
<th>Data Size</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;180</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>&lt;180</td>
<td>0</td>
<td>14</td>
</tr>
</tbody>
</table>

Deviation for log file storage is nil because as soon as the application gets loaded, the script embedded inside initiates the process of storing the log file as per SLA. Thus trust value for software service ($Trust_{TTSE}$) can be categorized as trusted or untrusted as mentioned in Table 4.17 below. Also based on the categorization of trust scores, in Table 4.18, the services are classified.

As per the algorithm the rejection condition for service is fixed at the malicious activity by the service if there has been any additional code gets injected, which can be evaluated through change in data size. Though there
are 7 services which have a deviation in data size, they are found have better score than the threshold level. Hence of the 34 services no single service falls under the non-trusted category. However, 7 services are under the revocable category are to be monitored further, which may be done by having a second level attestation process.

Table 4.17: TTSE distribution statistics

<table>
<thead>
<tr>
<th>Distribution Statistics</th>
<th>Trust Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.397</td>
</tr>
<tr>
<td>Max</td>
<td>0.999</td>
</tr>
<tr>
<td>Mean</td>
<td>0.778</td>
</tr>
<tr>
<td>First Quartile</td>
<td>0.695</td>
</tr>
<tr>
<td>Median</td>
<td>0.750</td>
</tr>
<tr>
<td>Third Quartile</td>
<td>0.962</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.202</td>
</tr>
<tr>
<td>Variance</td>
<td>0.0408</td>
</tr>
</tbody>
</table>

Table 4.18: Trust categorization for TTSE

<table>
<thead>
<tr>
<th>Trust Zone</th>
<th>Quantification Value</th>
<th>No. of Service Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Trust</td>
<td>&gt; 0.962</td>
<td>7</td>
</tr>
<tr>
<td>Trusted</td>
<td>&gt;0.750 to &lt; 0.962</td>
<td>9</td>
</tr>
<tr>
<td>Safe Zone</td>
<td>&gt;=0.695 to &lt;= 0.750</td>
<td>11</td>
</tr>
<tr>
<td>Revocable Trust</td>
<td>&gt;0.279 to &lt; 0.695</td>
<td>7</td>
</tr>
<tr>
<td>Irrevocable Trust</td>
<td>&lt;0.279</td>
<td>0</td>
</tr>
</tbody>
</table>

From the Table 4.18, it can be inferred that as the mean trust score is above the half way mark, the service, in general, have performed better. The reason is attributed to the precondition of the service level agreements enforced by the provider. However, the behaviour of the service while being accessed by the consumer should be monitored.

A sample of 5 users along with their service trust score is compared at 10 continuous time intervals. The progress of trust value for the service is depicted in Figure 4.5. It can be seen that after T1 and T2 the trust value gradually increases, this is mainly to the fact that the trust value is reported
back as an adaptive trust mechanism and hence the users perform better and prevent any malicious activity. Therefore user’s action improves the services trustworthiness.

![Service Behavior at various Time Intervals](image)

**Figure 4.5: Service trust score for 5 cloud consumers**

The variation in the trust value is because of the dynamic nature of the network and changes due to application response time. Moreover, since the service depends on the feedback factor given by the client, every service instance may differ, and therefore the feedback may also change. Having found that the service trust is satisfactory, the client trust score must be evaluated to complete the assessment of BTM.

### 4.6.5 Results and observation for TTCE

The final process of evaluating the client behaviour requires the feedback history and data size at the client to be the most important attributes. As previously mentioned, in section 4.4, the evaluation of TTCE is a 2 stage process, initially to find the QoS trust value and the login attempts trust value.

Finally, history and client capability trust are added to evaluate the final trust score for TTCE.
4.6.5.1 TQCA results

The first QoS attribute is the feedback history, which is the ratio of feedback given at the end after completion of the service.

Table 4.19: Feedback history distribution

<table>
<thead>
<tr>
<th>Normalized Feedback Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>0.5</td>
<td>4</td>
</tr>
<tr>
<td>&gt;0.5 to &lt;1</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
</tr>
</tbody>
</table>

Change in IP address is found for only four users, while the successful login attempt is calculated based on the number of time the user has logged and used the system successfully. Since it is a cloud application, every usage must be charged. Therefore, it is expected that the user logs in at every time the service gets initiated. The distribution is in Table 4.20.

Table 4.20: Successful login attempt distribution

<table>
<thead>
<tr>
<th>Successful Login Attempt</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;5</td>
<td>0.25</td>
<td>21</td>
</tr>
<tr>
<td>&gt;5 to &lt;10</td>
<td>0.5</td>
<td>8</td>
</tr>
<tr>
<td>&gt;10</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

SLA acceptance is by the enforcement process, so there are no users who have failed to accept the agreements. Data size deviation can significantly alter the behaviour of the user hence Table 4.14 is accepted for this calculation. Finally, user time spent is calculated as per Table 4.21.

Table 4.21: User time spent distribution

<table>
<thead>
<tr>
<th>User Time Spent</th>
<th>Normalized Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;50 sec</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>&gt; 50 to &lt;100</td>
<td>0.5</td>
<td>10</td>
</tr>
<tr>
<td>&gt;100</td>
<td>1</td>
<td>9</td>
</tr>
</tbody>
</table>
The trust score for the TQCA is calculated as per the formula in equation 10, and the distribution statistics of the results for TQCA is given in Table 4.22.

<table>
<thead>
<tr>
<th>Distribution Statistics</th>
<th>Trust Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.068</td>
</tr>
<tr>
<td>Max</td>
<td>0.935</td>
</tr>
<tr>
<td>Mean</td>
<td>0.585</td>
</tr>
<tr>
<td>First Quartile</td>
<td>0.466</td>
</tr>
<tr>
<td>Median</td>
<td>0.605</td>
</tr>
<tr>
<td>Third Quartile</td>
<td>0.685</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.216</td>
</tr>
<tr>
<td>Variance</td>
<td>0.0467</td>
</tr>
</tbody>
</table>

### 4.6.5.2 TACA results

The behaviour of failed login attempts must be separately calculated for the purpose of enhancing good behaviour and also to identify vulnerable users. Using the formula in equation 11, the results are as in Table 4.23.

<table>
<thead>
<tr>
<th>Distribution Statistics</th>
<th>Trust Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0</td>
</tr>
<tr>
<td>Max</td>
<td>1</td>
</tr>
<tr>
<td>Mean</td>
<td>0.541</td>
</tr>
<tr>
<td>First Quartile</td>
<td>0.418</td>
</tr>
<tr>
<td>Median</td>
<td>0.482</td>
</tr>
<tr>
<td>Third Quartile</td>
<td>0.78</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.276</td>
</tr>
<tr>
<td>Variance</td>
<td>0.076</td>
</tr>
</tbody>
</table>

By applying the enforcement authentication policy, 3 users have been blocked for a policy violation, while each one has been given a warning and an final attempt to clear the rule of login attempt.
4.6.5.3 Final TTCE results

By applying the formula in equation 12, the final trust value for the client is evaluated. The distribution of the results is in Table 4.24.

**Table 4.24: TTCE distribution statistics**

<table>
<thead>
<tr>
<th>Distribution Statistics</th>
<th>Trust Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.108</td>
</tr>
<tr>
<td>Max</td>
<td>0.613</td>
</tr>
<tr>
<td>Mean</td>
<td>0.423</td>
</tr>
<tr>
<td>First Quartile</td>
<td>0.334</td>
</tr>
<tr>
<td>Median</td>
<td>0.450</td>
</tr>
<tr>
<td>Third Quartile</td>
<td>0.508</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.119</td>
</tr>
<tr>
<td>Variance</td>
<td>0.014</td>
</tr>
</tbody>
</table>

The categorization of the values for client trust evaluation is given in Table 4.25. The rejection condition is the level of malicious interaction with the service with the intention of harming the service provider. Hence Data size is considered to be the choice for rejection condition.

**Table 4.25: Trust categorization for TTCE**

<table>
<thead>
<tr>
<th>Trust Zone</th>
<th>Quantification Value</th>
<th>No. of Cloud Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Trust</td>
<td>&gt; 0.508</td>
<td>9</td>
</tr>
<tr>
<td>Trusted</td>
<td>&gt;0.450 to &lt; 0.508</td>
<td>8</td>
</tr>
<tr>
<td>Safe Zone</td>
<td>&gt;=0.334 to &lt;= 0.450</td>
<td>8</td>
</tr>
<tr>
<td>Revocable Trust</td>
<td>&gt;0.295 to &lt; 0.334</td>
<td>5</td>
</tr>
<tr>
<td>Irrevocable Trust</td>
<td>&lt;0.295</td>
<td>4</td>
</tr>
</tbody>
</table>

The following can be observed for TTCE.

1. There have been 4 users categorized to have irrevocable trust.
2. Out of 5 revocable cases, 3 are blocked during the TACA process.
3. However, the model provides a chance to improve upon, based on the overall trust score.
4. These 3 users trust value is now adjusted to have the min. value.
5. Previous history of trust score aids in the computation of final trust.

![Consumer Behavior at various Time Intervals](image)

**Figure 4.6: Trust score for 5 cloud consumers**

Hence to find the legitimate users, the history of trust score is included from the next iteration onwards. In Figure 4.6 a sample of around 5 users who have been assessed at 10 continuous time intervals and their corresponding value are shown. It can be observed that at the 2\textsuperscript{nd} iteration itself the value of previous trust score aids to improve the overall value. Moreover, the graph increases linearly with not many fluctuations. The reason is due to the history of trust aids the present trust value, and the basic client configurations are fixed which may not change in time.

4.7. SUMMARY

- The chapter presents detailed information on how the BTM engages the users and the services based on the QoS attribute collected for monitoring the behaviour.

- The implementation of BTM is a three stage process; with basic trust related configuration defines the criteria and behaviour of the service, followed by consumers being evaluated through evidence.
• Algorithms for evaluating the trust at each stage for client configuration using TCC, service using TTSE and user using TTCE is proposed in this chapter.

• A novel approach for login attempt using enforcement behaviour policy is developed into a formula based on trial and error approach. This effectively benefits good attempts and punishes in case the attempts frequently fails. Through this, it is possible to identify an intruder if he is planning to gain into the administrative domain or into any other persons account.

• The implementation results identify deviations based on trust categorization. Hence the BTM provides a foundation for cloud attestation process to guarantee the trustworthiness.