CHAPTER V

PRESERVE HIGHLY CORRELATED ATTRIBUTE
USING INHERITED DIVERSE CLASSIFICATION TREE

5.1 INTRODUCTION

Most of the online transaction processing system related applications are client server applications deliberately support online user’s for direct access to information. The online transaction processing system applications progresses entities of effort referred as transactions. A single transaction might demand a bank balance and another might revise that balance to replicate a deposit. In a transaction processing system, one implementation of an application processes set of specific transactions. End users gather online data to the enterprise and to organization data, and openly commence transactions. In a transaction processing system, most of the users often process parallel transactions, and needs quick response to each transaction.

Traditional online transaction processing system permits to store information into databases in a quick, safe and efficient way to deliver meaningful information. Transaction processing is very easy to access the data but it becomes increasingly difficult to access the desired information. The confront with the paradox that more data means less information but which is appropriate. Because the cost of processing power and storage has been falling, data have become very cheap. It had opened a new challenge for computer science such as the discovery of new and meaningful information. Joint Threshold Administration Model (JTAM) is based on the standard of separating the objects from at least ‘k’ Data Base Administrator (DBA) but it produces overhead during the process of transaction. Similarly, existing cluster
adaptive distance bound in design VA-File, faces a comparatively small preprocessing storage overhead.

In order to overcome the above issues, it is planned to implement Inherited Diverse Classification Tree (IDCT) algorithm to choose the tree and fulfill the transaction processing capabilities by overcoming the overhead. IDCT algorithm developed by using specialized inherited operators preserve attributes structure of the trees. It preserves membership disclosure from being destroyed. IDCT algorithm uses a stage to strengthen and find high fitness trees. The IDCT algorithm supplies the analyst with a choice of classification trees, all of those having a good explanatory value. From the set of trees, the data analyst is able to choose the tree that fulfills the requirements and does not suffer from the weaknesses of the IDCT algorithm. The IDCT algorithm uses some dedicated genetic operators that are devised to preserve the structure of the trees from being destroyed and improves the transactional processing. Experimental evaluation of JTAM and VA File are compared with the IDCT algorithm in terms of improved classification accuracy and improved search accuracy and reduced transaction overhead taken as metrics.

5.2 DATA MINING IN ONLINE TRANSACTION PROCESSING

The analysis of data in business supports further knowledge about that business by deriving rules from the data stored. The data stored is understandable with Knowledge Discovery in Databases (KDD) that holds logical benefits for any enterprise. Even non vital process used for identifying suitable, narrative, potentially useful, and eventually understandable patterns in data. Sometimes KDD is varied as a term with data mining. The KDD employed to the entire extraction of knowledge from data, while the term data mining should be used totally for the detection stage of the KDD process.
Data mining is anxious with rising algorithms and computational tools help people to extract patterns from data. Finding the patterns by identifying the underlying rules and features in the data is done in an automatic way. The JTAM is based on a cryptographic threshold signature scheme, and shows how JTAM prevents malicious modifications to policy objects from authorized users but overhead occurs on the process of transaction. The DBMS selects $p, q$ as two large prime numbers such that

\[
p = 2p' + 1
\]
\[
q = 2q' + 1
\]

Where $p'$ and $q'$ are two large prime numbers. Let $n = p \times q$ and $m = p' \times q'$, then the server chooses the private key in such a way that $d \in Z$ and also evaluates the secret share key $s_i$ such that

\[
s_i = p(i) \mod m
\]

A cryptographic hash is generated as given below

\[
H(Pol) = SHA1(PolicyID, Conditions, InitialAction(s), OptionalAction(s), k, State)
\]

The two general types of data mining problems are forecasting and knowledge discovery. Forecasting tries to find causal relationship between some fields in the database. These relations are established by finding predictor variables that explain the variation of other, and autonomous variables. If a fundamental relationship has been established, actions are undertaken to reach a specific goal e.g. reduce the number in defects of a production line, or improve the customer satisfaction. Knowledge discovery problems typically explain a stage prior to prediction, where information is inadequate for prediction. The MV algorithm and Apriori algorithm with some enhancements, that aid in the process of filling the missing value and identification.
From the research of Sharadh Ramasamy et al., developed “An Cluster Adaptive Distance Bound based on separating hyper plane boundaries of voronoi clusters” balances cluster based index. The bound enable efficient spatial filtering, with a comparatively small preprocessing storage overhead and is suitable to Euclidean and Mahalanobis similarity measures [SRK11]. The Euclidean distance over feature similarity measure using cluster adaptive distance bound was represented as

\[ (x_1, x_2) = \sqrt{\sum (x_1 - x_2)^T(x_1 - x_2)} \]

The distance \( d \) from query \( q \) for a cluster \( X_m \) is then given as

\[ d(q, X_m) = \min d(q, x) \text{ where } x \in X_m \]

For a cluster \( X_m \) with query \( q \) and hyperplane \( H \) that lies between the cluster and query is given as

\[ d(q, X_m) \geq d(q, H) + \min(d(x, H)) \text{ where } x \in X_m \]

If \( H_{sep} \) denotes finite set of separating hyperplanes then it implies that

\[ d(q, X_m) \geq \max\{ d(q, H) + d(X_m, H) \} \]

Therefore a cluster distance bounds was designed on the basis of separating hyperplane boundaries with which the search index were complemented by bounds were applicable Euclidean and Mahalanobis distance metrics as given above.

Cluster-based storage schema reduces I/O cost in Traveler algorithm, which is based on cost analysis. An optimization technique and pseudo record additionally progress the search efficiency. The top k query in the higher dimension record set is presented using the N-way Traveler algorithm according to Lei Zou et al., the work N-way Traveler algorithm does not found to be relevant DG index for relationship
analysis. With the aid of Traveler algorithm, given two records \( r \) and \( r' \) the distance between \( r \) and \( r' \) is given as follows [LLC11].

\[
\text{Dis}(r, r') = |\text{Parent}(r) \cup \text{Parent}(r')| - |\text{Parent}(r) \cap \text{Parent}(r')| \\
= |\text{Parent}(r) - \text{Parent}(r')| - |\text{Parent}(r) + \text{Parent}(r')|
\]

Provided with a disk page \( P \), the upper corner and lower corner of \( P \) using Traveler algorithm are defined as follows:

\[
P^+ = \langle \text{Max}(I_1), \ldots, \text{Max}(I_n) \rangle \\
P^- = \langle \text{Min}(I_1), \ldots, \text{Min}(I_n) \rangle
\]

**Data Mining Methods in Transaction Processing Capabilities**

Although many existing techniques from such fields as synthetic intelligence, statistics and database system have been used to effect transaction processing capabilities. The data mining becomes an independent new field of research. One of the first approaches was used multiple regressions. Regression revealed many drawbacks when applied to data mining problems. One of the shortcomings of multiple regressions is that it requires quantitative rather than qualitative data elements. The reason is that the effectiveness of regression is unable to detect interactions at more than one level.

A privacy preserving protocol for mining maintain counts, sustain high accuracy and strong privacy. Yet, a different problem is with perturbation techniques for getting better efficiency without losing any accuracy or only losing satisfactory limits of accuracy. Information theoretic inference controlled by Nan Zhang et al., with a combination of common aggregate functional bit. The information theoretic inference control is not effective on general private data query auditing. The level of privacy guarantees the level of privacy disclosure not to exceed thresholds predetermined by the data owners [NZW10].
Minimum spanning tree based clustering algorithms referred by Xiao chun Wang et al., is a source of computing that the divide and conquer approach and chiefly the whole data set cannot fit into the main memory. An efficient index called IR-tree jointly with a top-k document search algorithm assist, four main tasks in document searches, but doesn’t improve the IR tree index based on a variety of access patterns [XXD09].

Let $p$ be the probability that is designed in such a way that a random data point hits its nearest neighbor. Let $y$ denote the random variable measuring the total number of trials needed for a specific random data point that has to hit its nearest neighbor. Then, the probability of obtaining a success on trial $y$ is given by

$$p(y) = q^{y-1}p$$

From above $q = 1 - p$ represents a probability for the occurrence of failure. JTAM is based on the standard of separating the objects from at least ‘k’ database administrator (DBA) but overhead occurs on the process of transaction by the author Ashish Kamran et al., Cluster adaptive distance bound refered by the research of Sharadh Ramaswamy et al., is based on separating hyper plane boundaries of Voronoi clusters balance cluster based index [AEB11]. The bound enable efficient spatial filtering, with a comparatively small preprocessing storage overhead [SRK11]. Another class of detection problem by the author Mohamed et al., becomes more demanding in the presence of concept-drift. But that concept fails in addressing the data stream classification problem under active feature sets, when the underlying data distributions streams are developed [MJL11].
5.3 INHERITED DIVERSE CLASSIFICATION TREE (IDCT) IN PRESERVATION OF HIGHLY CORRELATED ATTRIBUTE

The Inherited Diverse Classification Tree (IDCT) algorithm operates with large samples of relational attribute in the transactional databases. IDCT divides the population of subjective attribute into binary analytical variable and dependent categorical variable. Based on statistical significance of transactional database, number of subjective attribute becomes more critical on preserving privacy of user relevant data. The IDCT considers correlated variables to evaluate the cohesiveness across the user transaction.

![Flow Diagram of IDCT Algorithm](image)

*Fig. 5.1 Flow Diagram of IDCT Algorithm*

The data stream is a mechanical and automatic system that takes off the steps by an experienced data analyst to determine strong data interaction effects. The idea behind IDCT algorithm is to explain the variance of a binary analytical variable and dependent variable through a comprehensive search of all probable relationships between predictors and the dependent variable. The criticisms of the IDCT algorithm
is that it tends to be exceedingly aggressive at finding relationships and dependencies in data and that could differentiate meaningful from meaningless relationships.

The results of the search are represented by binary tree. The nodes represent analytical variables in a binary tree split that explains the largest part of search. In a first step of IDCT algorithm, every possible binary analytical variable is tested to see which one has the strongest analytical power by classifying the binary tree. The population of IDCT algorithm is then split into two classes according to variables. The process is often for the descendent classes, until some triggering principle is met. IDCT algorithm classifies the binary tree to minimize the classification error rate in first class. And the secondary class devises the genetic operators for effective transaction processing.

**Design Notation of Inherited Diverse Binary Tree Classification**

IDCT algorithm works with categorical variables with many possible values. The assumption restricts the variables in the IDCT to analyze binary ones. In further process, the IDCT algorithm includes variables with more than two dissimilar possible values. Q is a set of elements in the binary tree. \( \hat{Q} \) is the union of elements in the binary tree and its leaf node elements \( \hat{Q} = Q \cup \{\ast\} \), where \( \ast \) is a terminal symbol. A classification tree over Q is a quadruple. The set of components in the binary tree are maintained as \( A = \{U, F, \varepsilon, \delta\} \) with, \( (U, F) \) is a finite binary tree with vertices (nodes) ‘U’ and edges ‘F’. The set of final nodes is denoted by \( T \). The number of leaves is also called the width of the tree and is denoted by \( w(A) \).

For instance, consider \( \varepsilon: (U, S) \ast (+, ) \Rightarrow F \) performs the one to one map. For \( u \in U S, \varepsilon: (u, +) \) is left edge and \( \varepsilon: (u, s) \) is right edge. \( \delta: U \Rightarrow \hat{Q} \) is a map satisfying \( \delta(S) \in \{\ast\} \) and \( \delta(U S) \in \mathcal{P} \). \( \delta \) labels the elements of ; leaves are label by \( \ast \), non final nodes are label by an element of ‘Q’. If \( A = (U, F, \varepsilon, \delta) \) and if \( u \in U \) then, \( A^u = \)
\((V_u, F_u, \epsilon_{U-S}, \delta_u)\) denotes the sub tree with top ‘u’. This exposes an inherited diverse classification tree. The graphical representation of a classification tree is a picture of the tree. IDCT algorithm represents the vertices u by their label \(\delta_u\), sketch arrows to represent the edges and label the left edges by ‘+’ and the right edges by ‘-’.

![Three Level of Complete Binary Tree](image.png)

**Fig. 5.2 Three Level of Complete Binary Tree**

A binary tree classification of height ‘n’ where the paths from the top node to every leaf node all have equal length ‘n’ a complete binary n-tree. Two different trees take up the same qualified position in the classification trees, if the path from the top to leaf node is same as determined by the labels of the edges on the path. The rank of a node in the tree is the distance end to end path from the top node to that node, plus one.

For each classification tree, a fitness value \(f (A) \in \mathcal{P}^+\) is assigned which denotes its explanatory power, i.e, classification tree. In fitness function a difference are made between virtual fitness. The virtual fitness in fitness function describes the contents of each virtual function. In IDCT algorithm research, the virtual function is used for keeping the structure of the trees constant.
The content and the internal features determine the virtual fitness of the tree. IDCT algorithm uses the virtual function by keeping the structure of the trees constant. In case the nodes represent binary tests over a feature set. The binary tests classify the subjects into classes, depending on their scores on the predictor variables. The virtual fitness \( f_r(A) \) is effective measure of classification tree to reduce error rate. If a classification is perfect in IDCT algorithm, the difference within the classes is equal to zero value, whereas the difference between the several classes obtains maximal value.

The fitness function used in the implementation of the IDCT algorithm considers the classes of subjects defined by the analyst in the tree. In the inherited derived classification tree, the classes are represented by an asterisk (*), as shown in Fig. 5.2. borrowing from one way the following virtual fitness is assigned to each binary tree classification.

\[
f_r(A) = \frac{\sum_{i=1}^{k} m_i (\bar{y}_i - \bar{y})}{\sum_{i=1}^{k} \sum_{j=1}^{m_i} (y_{ij} - \bar{y})^2}
\]

Where, \( m_i \) is the number of observations in class \( i \), ‘\( k \)’ is the number of classes \( y_{ij} \) is the \( j^{th} \) observation in class \( i \), \( \bar{y}_i \) is the class ‘\( i \)’ sample mean and \( \bar{y} \) is the overall sample mean. Calculating the fitness of a classification tree is a mathematically complex task as each subject in the population has to be classified into one of the \( width(A) \) classes. After that, the sum of squares in each class has to be calculated. Taking into account, devise genetic operators in such a way that the amount of fitness function assessment improves the transaction processing. Ensure the virtual fitness trees using the genetic operators to attain effective transaction processing.
**Genetic Operators in Transaction process**

Given a set \( Q \), assume that a fitness value is assigned to each classification tree with high fitness. Starting from an IDCT algorithm, randomly generated population of classification trees and the genetic algorithm is used to get next successive populations including new elements for effective transaction processing. The tools are called as genetic operators. In a reproduction phase, a number of classification trees are selected. Then various genetic operators such as crossover, switch, and translocation as well as definite micro operators are performed with a definite probability.

The genetic operators are performed based on priority of higher survival of fitness trees. In IDCT genetic operators, attempt to attach to the rule that high fitness is a property of trees that should be at least incompletely conserved by genetic operators. Every genetic operator defined here conserves the size and form of the tree, considering that only complete \( n \) trees are used. Performing genetic operators on several complete \( n \) trees, yield effective transaction over data stream.

The probability \( Q(A) \) that an element ‘\( A \)’ is chosen out of a given population of trees’s is proportional to its fitness

\[
Q(A) = \frac{f(A)}{\sum_{c \in S} f(B)}
\]

Naturally, more compound probability functions are devised in genetic operation, but for function, this simple function is enough. Crossover does not change the population if both selected nodes are final nodes and sub trees are the same. As a result, crossover is applied only on nodes and has diverse fundamental trees. The crossover operator in IDCT algorithm, although a bit restrictive, works very well in conserving the fitness of high fitness trees. Crossover maintains the size and shape of the trees if both trees are complete \( n \) trees.
A couple of remarkable ways for performing crossover operator devise a pair of nodes in the same virtual location with a certain probability at processing. A more aggressive approach is to provide relevant crossover for each pair of nodes that are in the same relative position at both trees with a certain probability. The transposition operator is a kind of auto crossover operator. In order to perform transposition, first select randomly two non final nodes ‘u1’ and ‘u2’ that appear on the same level of a selected tree \( A_1 = \{U, F, e, \delta\} \). Then switch the sub trees in IDCT algorithm.

The requirement for \( A_1^{u1} \) and \( A_1^{u2} \) nodes are to be on the same level of the tree. Similar level of the tree is needed to preserve the structure of the tree after transposition.

Followed by crossover operation, different versions of operator are devised, depending on the aggressiveness giving the transposition operator on a certain probability to transposition two nodes in every tree. Genetic operator allows the transposition operator to switch each pair of nodes on the same level with certain, smaller probability to perform the effective transaction processing.

When the user details are grouped and sorted based on the frequency, the frequency classification algorithm classifies the correlated attribute. These attribute are given to the IDCT algorithm for classifying less error data using binary tree and it is preceded with genetic algorithm. In the final phase, it processes the given details with genetic operators using high fitness tree. Finally the credit card transaction takes places with high security. The flow diagram of the IDCT algorithm in the credit card application is given in Fig. 5.3.
The IDCT genetic operator namely crossover and transposition is termed as micro operators, because the function of cut and paste of complete sub trees within and between trees. Considering that the fitness of a tree gets better by exchanging sub tree between or within two trees or exchanges the labels of two edges. But in many cases, the fitness of a tree is also increased with less drastic measures. Therefore, micro operators are devised. Micro operators are operators that impact openly on the contents of the nodes of the trees.

Fig. 5.3 Flow diagram of IDCT for Credit Card Application

Depending on the kind of classification tree, different micro operators are devised. In case, every non terminal node is labeled by an analyst variable. Micro operators change the analysis variable used in a certain node into another one. The operators perform only on non terminal nodes of trees selected through reproduction. The main difference between micro operators is that it does not replace entire sub
trees, only individual analytical binary variables are exchanged for processing. The algorithmic flow of inherited diverse classification tree steps are shown below

The existing JTAM algorithm is provided as the anomaly detection of JTAM provides its anomaly assessment with the help of two main categories of anomaly attributes. The first category of anomaly attribute is the contextual category that included all the attributes describing the context of the anomalous request such as user, role, source, and time. The second category of anomaly attribute also called as the structural category, consists of every attributes that convey information regarding the structure of the anomalous request such as SQL command, and accessed database objects. The task of detection engine is to submit its characterization of the anomaly with the aid of anomaly attributes.

Once a database request has been flagged off as anomalous, an action is executed by the response system to address the anomaly. The response action to be executed is specified as part of a response policy. The conservative actions are low severity actions. Such actions may log the details send as alert, but they do not proactively prevent a fault. Aggressive actions, on the other hand, are high severity responses. Such actions are capable of preventing a fault either dropping the request, disconnecting the user revoking/denying the necessary privilege. Such action may suspend or taint request. The suspended request simply put on the hold, until some specific actions are executed by the user.

As a result, the anomaly attributes acts as an interface for the response engine, thereby hiding the internals of the detection mechanism. Once a database request has been flagged off to be anomalous, then an action is responded by the response system provide immediate response to the anomaly. The detailed algorithmic description of JTAM is provided below:
Existing JTAM Algorithm

**Step 1: Policy creation**
- Prompts for password
- Encrypt the password
- Generate cryptographic hash
- Create signature

**Step 2: Authorize the response policy**
- if role = DBA and Object Type = table and Data Time Between range then NOP

**Step 3: Signature combining and verification**
- Re-authenticate unprivileged users who are logged from inside the organization's internal network for write anomalies to tables in the dbo schema. If re-authentication fails, drop the request and disconnect the user else do nothing.

**Step 5: Policy alteration**
- if role ! = DBA and Obj Type = table and Objs in dbo and SQLCmd in {Insert, Update, Delete} then SUSPEND
  - CONFIRM RE-AUTHENTICATE
  - ON SUCCESS NOP
- End if

**Step 6: Policy drop**
- If ON FAILURE ABORT,
  - DISCONNECT
- end if
Existing JTAM Algorithm with Policy Activation and Suspension

//Policy Creation
Input  : Stream of Data for processing
Output: Effective processing with minimal error rate
Begin

Step 1: Create Response Policy [Policy Data] Jointly Administered
By k Users;
  Prompts DBA1 for its password
  Decrypt the encrypted secret share of DBA1 corresponding
to the value of k =3 to get s1
  \[ H(Pol) = \text{SHA1}(\text{PolicyID}, \text{Conditions, InitialAction(s), OptionalAction(s), k, State}) \]

//Policy Activation
Step 2: Authorize Response Policy [Policy ID] Create;
  Prompts DBA3 for its password
  Decrypt the encrypted secret share of DBA3 corresponding
to k = 3 to get s3
  Creates a signature share on H(Pol)
  using the secret share s3
  Decrements the value in column r by 1

//Policy Suspension
Step 3: Suspend Response Policy [Policy ID]
  Prompts DBA2 for its password
  Decrypt the encrypted secret share of DBA2 corresponding
to k = 3 to get s2
  Creates a signature share, WDBA_1 on H(Pol) using the
  secret share s2

Step 4: Alter Response Policy [Policy ID] [Policy Data] command
Step 5: Drop Response Policy [Policy ID] command

//Signature Verification
Step 6: Signature verification performed according to Policy 1, Policy 2

End
Proposed Inherited Diverse Classification Tree Algorithm

**Input:** Stream of data for processing  
**Output:** Effective processing with minimal error rate

**Begin**

// Binary Tree Classification

**Step 1:** Analysis the variable in data stream  
**Step 2:** Classify the binary tree  
**Step 3:** \( (U S) \ast (+, ) \rightarrow F \) performs the one to one map  
**Step 4:** Relative position obtained on classification tree  
**Step 5:** Obtain error free classification tree

// Genetic Operator

**Step 6:** Initialize by choosing a random population  
**Step 7:** Calculate the virtual fitness for each member in population  
**Step 8:** Apply genetic operator for processing  
\[ i. \quad \text{Crossover takes place at randomly chosen vertices with a probability} \]  
\[ ii. \quad \text{Transposition takes place at both trees with probability} \]  
**Step 9:** Micro operator used to obtain the adjoining resultant trees

**End**

The above algorithm presented an interactive Event-Condition-Action type response policy language that helped the database administration to provide response actions for users at different circumstances but tradeoff with overhead was observed on the transaction processing capabilities of the DBMS.

The algorithmic steps are used for performing the transaction processing. The entire procedure is repeated until the specified criterion is fulfilled, usually until certain number of generations has been generated. A more aggressive IDCT algorithm is obtained by employing the effective version of the micro operators. At last, IDCT algorithm fulfills the transaction processing capabilities resolving the overhead.
5.4 EXPERIMENTAL EVALUATION

5.4.1 Performance Evaluation of IDCT for Credit Approval Data Set

Experimental evaluation is carried out to estimate the performance of classification accuracy, virtual fitness function efficiency and transaction overhead of proposed IDCT and existing JTAM and VA file. The IDCT algorithm is implemented in java and tested for credit approval data set; the results are evaluated using the above mentioned equations in 3.1, 3.4 and 3.5.

a) Classification Accuracy

In this section the quantitative evaluation for the proposed IDCT algorithm with classification accuracy is given and compared with existing JTAM and VA File.

<table>
<thead>
<tr>
<th>No. of Data Stream</th>
<th>Classification Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JTAM</td>
</tr>
<tr>
<td>50</td>
<td>79.25</td>
</tr>
<tr>
<td>100</td>
<td>75.79</td>
</tr>
<tr>
<td>150</td>
<td>83.23</td>
</tr>
<tr>
<td>200</td>
<td>75.56</td>
</tr>
<tr>
<td>250</td>
<td>75.14</td>
</tr>
<tr>
<td>300</td>
<td>82.14</td>
</tr>
</tbody>
</table>

Table 5.1 shows the classification accuracy rate using the proposed IDCT algorithm that ranges from 92.45 – 93.68 %. Similarly, the classification accuracy rate of VA-File ranges from 75.14 – 83.23 % whereas JTAM measures the classification accuracy rate as 81.23 – 81.89 %.
Fig. 5.4 shows the impact of classification accuracy. From this it is evident that the classification accuracy increases with the increase in the number of records using the proposed IDCT algorithm when compared to the two other existing methods JTAM and VA-File. The IDCT algorithm which helps in improving the classification accuracy by 17 – 18 % compared to JTAM and 11–12 % compared to VA File.

b) Virtual Fitness Function Efficiency

The virtual function efficiency for the proposed algorithm IDCT is compared with the existing JTAM and VA File. Table 5.3 shows the values obtained for virtual function efficiency using IDCT algorithm and two other methods VA File and JTAM.

Fig. 5.5 shows the impact of virtual function efficiency with respect to the number of correlated attributes in the range of 2 to 12. From the Fig, this can be observed that though with the increase in number of correlated attributes and virtual fitness function efficiency time using the proposed IDCT algorithm is lowering.
Table 5.3 Comparison Table for Virtual Fitness Function Efficiency of IDCT (Credit Approval Data Set)

<table>
<thead>
<tr>
<th>No. of Correlated Attributes</th>
<th>Virtual Fitness Function Efficiency (ms)</th>
<th>JTAM</th>
<th>VA-File</th>
<th>IDCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>200</td>
<td>190</td>
<td>180</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>280</td>
<td>260</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>360</td>
<td>320</td>
<td>315</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>470</td>
<td>440</td>
<td>425</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>530</td>
<td>510</td>
<td>480</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>640</td>
<td>600</td>
<td>570</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3 depicts the performance evaluation of virtual function efficiency for IDCT algorithm, JTAM and VA-File. The IDCT algorithm has taken less time for virtual fitness function efficiency in credit approval dataset.

Fig. 5.5 Comparison Graph for Virtual Fitness Function Efficiency of IDCT (Credit Approval Data Set)
Fig. 5.5 describes the virtual function efficiency measure based on the number of correlated attributes present in the dataset. The IDCT algorithm improved the virtual fitness function efficiency to 11 – 23% less when compared with the JTAM and nearly 9 – 11% less than VA-File.

c) **Transaction Overhead**

The transaction overhead for the proposed algorithm IDCT is compared with the existing JTAM and VA File. Table 5.4 shows the values obtained for transaction overhead using IDCT algorithm and existing methods VA File and JTAM.

Fig. 5.8 shows the impact of transaction overhead with respect to the number of data stream 50 to 300. It can be observed that with the increase in data stream transaction overhead also has to be increased, but comparatively recorded transaction overhead is lowering in the proposed IDCT algorithm.

<table>
<thead>
<tr>
<th>No. of Data Stream</th>
<th>Transaction Overhead (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTAM</td>
<td>VA-File</td>
</tr>
<tr>
<td>50</td>
<td>899</td>
</tr>
<tr>
<td>100</td>
<td>903</td>
</tr>
<tr>
<td>150</td>
<td>935</td>
</tr>
<tr>
<td>200</td>
<td>946</td>
</tr>
<tr>
<td>250</td>
<td>962</td>
</tr>
<tr>
<td>300</td>
<td>975</td>
</tr>
</tbody>
</table>

Table 5.4 depicts the performance evaluation of transaction overhead for IDCT algorithm, JTAM and VA-File. The IDCT algorithm has taken less ms for transaction overhead in credit approval data set.
Fig. 5.6 describes the transaction overhead measure based on the number of data streams present in the dataset. The IDCT algorithm reduces the transaction overhead to 11–23 % less when compared with the JTAM and nearly 9–11 % less than VA-File.

5.4.2 Performance Evaluation of IDCT for Spam Base Data Set

Experimental evaluation is carried out to estimate the performance of classification accuracy, virtual fitness function efficiency and transaction overhead of proposed IDCT and existing JTAM and VA-File. The IDCT algorithm is implemented in java and tested for spam base data set; the results are evaluated using the above mentioned equations in 3.1, 3.4 and 3.5.

a) Classification Accuracy

This provides an analysis for classification accuracy and compared with two other existing methods, Joint Threshold Administration Model (JTAM) and Vector Approximation (VA-File).
Table 5.5 Comparison Table for Classification Accuracy of IDCT (Spam Base Data Set)

<table>
<thead>
<tr>
<th>No. of Data Stream</th>
<th>Classification Accuracy (%)</th>
<th>JTAM</th>
<th>VA-File</th>
<th>IDCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>83.21</td>
<td>80.23</td>
<td>92.25</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>74.79</td>
<td>80.46</td>
<td>91.56</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>74.39</td>
<td>82.28</td>
<td>92.26</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>83.15</td>
<td>84.26</td>
<td>92.68</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>74.24</td>
<td>80.79</td>
<td>91.23</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>74.79</td>
<td>80.27</td>
<td>91.48</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5 shows the tabulation of classification accuracy of the proposed IDCT algorithm in the range of 91.23 – 92.68 %. The classification accuracy using VA-File was in the range of 80.23 – 84.62 % whereas 74.24 – 83.21 using JTAM.

Fig. 5.7 Comparison Graph for Classification Accuracy of IDCT (Spam Database)

Fig. 5.7 shows the impact of classification accuracy using the three methods IDCT algorithm and comparison is made with two other methods namely, JTAM and VA-File. From the Fig, the classification accuracy is high compared to two other
methods the effectiveness of classification accuracy by 9 – 18 % compared to JTAM and 8 – 12 % compared to VA-File.

b) Virtual Fitness Function Efficiency

This analysis has been carried out to prove virtual fitness function efficiency among the methodologies under research.

<table>
<thead>
<tr>
<th>No. of Correlated Attributes</th>
<th>Virtual Fitness Function Efficiency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JTAM</td>
</tr>
<tr>
<td>2</td>
<td>210</td>
</tr>
<tr>
<td>4</td>
<td>290</td>
</tr>
<tr>
<td>6</td>
<td>370</td>
</tr>
<tr>
<td>8</td>
<td>480</td>
</tr>
<tr>
<td>10</td>
<td>540</td>
</tr>
<tr>
<td>12</td>
<td>660</td>
</tr>
</tbody>
</table>

Table 5.7 shows the tabulation of virtual fitness function efficiency of the proposed IDCT algorithm in the range of 190 ms to 580 ms using VA-File, the virtual fitness function efficiency was in the range of 200 ms to 610 ms and JTAM recorded the virtual fitness function efficiency in the range of 210 ms to 660 ms with correlated attributes JTAM.

Fig. 5.11 shows the impact of virtual fitness function measured using the proposed IDCT algorithm for the purpose of comparative analysis. From the Figure, it is evident that the time taken to execute using the IDCT algorithm was comparatively lesser than two other methods, number of correlated attributes ranging 2 to 12. The improvement was observed in the range of 12 – 27 % compared to JTAM and 9 – 17 % compared to VA-File.
Fig. 5.8 Comparison Graph for Virtual Fitness Function Efficiency of IDCT (Spam Base Data Set)

Fig. 5.8 describes the virtual fitness function efficiency measure based on the number of correlated attributes present in the dataset. The IDCT algorithm reduces the virtual fitness function efficiency to 11 – 23 % less when compared with the JTAM and nearly 9 – 11 % less than VA-File.

c) Transaction Overhead

Table 5.8 shows the tabulation of transaction overhead of the proposed IDCT algorithm in the range of 869 ms to 940 ms. Using VA-File, the transaction overhead was in the range of 880 ms to 970 ms and JTAM transaction overhead is in the range of 900 ms to 978 ms with 60 analyst.
Table 5.8 Comparison Table for Transaction Overhead of IDCT  
(Spam Base Data Set)

<table>
<thead>
<tr>
<th>No. of Data Stream</th>
<th>Transaction Overhead (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JTAM</td>
</tr>
<tr>
<td>50</td>
<td>900</td>
</tr>
<tr>
<td>100</td>
<td>913</td>
</tr>
<tr>
<td>150</td>
<td>940</td>
</tr>
<tr>
<td>200</td>
<td>953</td>
</tr>
<tr>
<td>250</td>
<td>970</td>
</tr>
<tr>
<td>300</td>
<td>978</td>
</tr>
</tbody>
</table>

From the Fig. 5.9, it is evident that the transaction overhead using the IDCT algorithm was comparatively lesser than two other methods for number of records ranging 10 to 60. The improvement was observed in the range of 12 – 27 % compared to JTAM and 9 – 17 % compared to VA-File.

Fig. 5.9 Comparison Graph for Transaction Overhead of IDCT (Spam Base Data Set)
Fig. 5.9 describes the transaction overhead measure based on the number of data streams present in the dataset. As the above parameter, transaction overhead is evaluated based on the records. IDCT algorithm reduces the transaction overhead to 11 – 23 % less when compared with the JTAM and nearly 9 – 11 % less than VA-File.

5.4.3 Performance Evaluation of IDCT for Adult Data Set

The Experimental evaluation is carried out to estimate the performance of classification accuracy, virtual fitness function efficiency and transaction overhead of proposed IDCT and existing JTAM and VA file. The IDCT algorithm is implemented in java and tested for adult data set; the results are evaluated using the above mentioned equations in 3.1, 3.4 and 3.5.

a) Classification Accuracy

To measure the efficacy of the proposed Inherited Diverse Classification Tree, experimental analysis for classification accuracy is presented and compared with Joint Threshold Administration Model and Vector Approximation (VA-File).

<table>
<thead>
<tr>
<th>No. of Data Stream</th>
<th>Classification Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTAM</td>
<td>VA-File</td>
</tr>
<tr>
<td>50 78.35</td>
<td>83.56</td>
</tr>
<tr>
<td>100 78.56</td>
<td>83.69</td>
</tr>
<tr>
<td>150 79.26</td>
<td>84.26</td>
</tr>
<tr>
<td>200 79.56</td>
<td>85.65</td>
</tr>
<tr>
<td>250 80.12</td>
<td>85.96</td>
</tr>
<tr>
<td>300 81.32</td>
<td>86.00</td>
</tr>
</tbody>
</table>
Table 5.9 measures the classification accuracy using IDCT algorithm, JTAM and VA-File. The number of data streams used for measuring classification error rate varies in the range of 50 to 300.

Fig. 5.10 Comparison Graph for Classification Accuracy of IDCT (Adult Data Set)

Fig. 5.10 shows the impact of classification accuracy and comparison is made with two other existing methods JTAM and VA-File. From the figure, it is evident that the classification accuracy is reduced by 13 – 14 % in IDCT algorithm and compared to VA-File the error rate is reduced by 9 – 11 % in IDCT algorithm.

b) Virtual Fitness Function Efficiency

To analyse the efficiency of the proposed IDCT, experimental analysis for virtual fitness function is presented and compared with JTAM and VA-File.
Table 5.11 Comparison Table for Virtual Fitness Function Efficiency of IDCT (Adult Data Set)

<table>
<thead>
<tr>
<th>No. of Correlated Attributes</th>
<th>Virtual Fitness Function Efficiency (ms)</th>
<th>JTAM</th>
<th>VA-File</th>
<th>IDCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>230</td>
<td>220</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>310</td>
<td>290</td>
<td>265</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>400</td>
<td>350</td>
<td>330</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>500</td>
<td>480</td>
<td>440</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>560</td>
<td>530</td>
<td>500</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>680</td>
<td>630</td>
<td>590</td>
</tr>
</tbody>
</table>

Table 5.11 tabulated to depict virtual fitness function of IDCT algorithm and to compare with JTAM and VA-File.

Fig. 5.11 Comparison Graph for Virtual Fitness Function Efficiency of IDCT (Adult Data Set)
Fig. 5.11 describes the virtual fitness function efficiency measure based on the number of record correlated attribute in the data set. As the above parameter, virtual fitness function efficiency is evaluated number of record correlated attribute. IDCT algorithm reduces the virtual fitness function efficiency to 11–23 % less when compared with the JTAM and nearly 9–11 % less than VA-File.

c) Transaction Overhead

As other criteria, measuring transaction overhead also proves that the proposed method is highly effective in reducing the transaction overhead than other existing methods.

Table 5.12 Comparison Table for Transaction Overhead of IDCT (Adult Data Set)

<table>
<thead>
<tr>
<th>No. of Data Streams</th>
<th>Transaction Overhead (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JTAM</td>
</tr>
<tr>
<td>50</td>
<td>912</td>
</tr>
<tr>
<td>100</td>
<td>922</td>
</tr>
<tr>
<td>150</td>
<td>945</td>
</tr>
<tr>
<td>200</td>
<td>960</td>
</tr>
<tr>
<td>250</td>
<td>973</td>
</tr>
<tr>
<td>300</td>
<td>980</td>
</tr>
</tbody>
</table>

Table 5.12 tabulated to prove transaction overhead for IDCT algorithm also good than JTAM and VA-File. Fig. 5.15 shows the impact of transaction overhead using IDCT algorithm, JTAM and VA-File. The transaction overhead is comparatively less using IDCT algorithm than that of using JTAM and VA-file. This result in the minimization of transaction overhead by 10–27 % compared to JTAM and 6–20 % compared to VA-File respectively.
5.5 SUMMARY

Inherited Diverse Classification Tree (IDCT) algorithm achieves the transaction processing capabilities by overcoming the transaction overhead. IDCT algorithm in incorporates genetic operators that operates explicitly on the binary classification trees. The designed IDCT with genetic operators achieves high level virtual fitness trees among the population. From the classification of trees, the analyst chooses the tree that best meets every criteria. IDCT algorithm developed by using specialized inherited operators maintenance attributes structure of the trees. The IDCT algorithm supports high virtual fitness trees to have a strong base. The IDCT algorithm with dedicated genetic operators is efficiently devised to preserve the structure of the trees from being destroyed and improves the transactional processing.