Volume-1, Issue-2

SREEPATHY
JOURNAL OF COMPUTER SCIENCE & ENGINEERING

Published by

Department of Computer Science and Engineering
Sreepathy Institute of Management and Technology, Vavanoor
Palakkad - 679 533

January 2015

Contents

Optimization of Horizontal Aggregation in SQL by Using K-Means Clustering

Nisha S,
Dept. of Computer Science and Engg,
SIMAT, Vavanoor, Palakkad
nisha.s@simat.ac.in

***Abstract*—To analyze data efficiently, Data mining systems are widely using datasets with columns in horizontal tabular layout. Preparing a data set is more complex task in a data mining project, requires many SQL queries, joining tables and aggregating columns. Conventional RDBMS usually manage tables with vertical form. Aggregated columns in a horizontal tabular layout returns set of numbers, instead of one number per row. The system uses one parent table and different child tables, operations are then performed on the data loaded from multiple tables. PIVOT operator, offered by RDBMS is used to calculate aggregate operations. PIVOT method is much faster method and offers much scalability. Partitioning large set of data, obtained from the result of horizontal aggregation, in to homogeneous cluster is important task in this system. K- means algorithm using SQL is best suited for implementing this operation.**

***Keywords*—Aggregation, Data Mining, Structured query language (SQL), PIVOT, K-means algorithm.**

I. INTRODUCTION

HORIZONTAL aggregation is new class of function to return aggregated columns in a horizontal layout. Most algorithms require datasets with horizontal layout as input with several records and one variable or dimensions per columns. Managing large data sets without DBMS support can be a difficult task. Trying different subsets of data points and dimensions is more flexible, faster and easier to do inside a relational database with SQL queries than outside with alternative tool. Horizontal aggregation can be performing by using operator, it can easily be implemented inside a query processor, much like a select, project and join. PIVOT operator on tabular data that exchange rows, enable data transformations useful in data modeling, data analysis, and data presentation. There are many existing functions and operators for aggregation in Structured Query Language. The most commonly used aggregation is the sum of a column and other aggregation operators return the average, maximum, minimum or row count over groups of rows. All operations for aggregation have many limitations to build large data sets for data mining purposes. Database schemas are also highly normalized for On-Line Transaction Processing (OLTP) systems where data sets that are stored in a relational database or data warehouse. But data mining, statistical or machine learning algorithms generally require aggregated data in summarized form. Data mining algorithm requires suitable input in the form of cross tabular (horizontal) form, significant effort is required to compute aggregations for this purpose. Such effort is due to the amount and complexity of SQL code which needs to be written, optimized and tested.

Data aggregation is a process in which information is gathered and expressed in a summary form, and which is used for purposes such as statistical analysis. A common aggregation purpose is to get more information about particular groups based on specific variables such as age, name, phone number, address, profession, or income. Most algorithms require input as a data set with a horizontal layout, with several records and one variable or dimension per column. That technique is used with models like clustering, classification, regression and PCA. Dimension used in data mining technique are point dimension.

There are several advantages for horizontal aggregation. First one is horizontal aggregation represent a template to generate SQL code from a data mining tool. This SQL code reduces manual work in the data preparation phase in data mining related project. Second is automatically generated code, which is more efficient than end user written SQL code. Thus datasets for the data mining projects can be created in less time. Third advantage is the data sets can be created entirely inside the DBMS. K-means clustering algorithms are used to cluster the attribute, that attribute is the result of horizontal aggregation.

The rest of the paper is organized as follows. Next part presents clustering of aggregated dataset and different methods existing for aggregation and Conclusion.

II. RELATED WORKS

SQL extensions to define aggregate functions for association rule mining. Their optimizations have the purpose of avoiding joins to express cell formulas, but are not optimized to perform partial transposition for each group of result rows. Conor Cunningalam [1] proposed an optimization and Execution strategies in an RDBMS which uses two operators i.e., PIVOT operator on tabular data that exchange rows and columns, enable data transformations useful in data modelling, data analysis, and data presentation. They can quite easily be implemented inside a query processor system, much like select, project, and join operator. Such a design provides opportunities for better performance, both during query optimization and query execution. Pivot is an extension of Group By with unique restrictions and optimization opportunities, and this makes it very easy to introduce incrementally on top of existing grouping implementations. H Wang.C.Zaniolo [2] proposed a small but Complete SQL Extension for data Mining and Data Streams. This technique is a powerful database language and system that enables users to develop complete data-intensive applications in SQL by writing new aggregates and table functions in SQL, rather than in procedural languages as in current Object-Relational systems. The ATLaS system consist of applications including various data mining functions, that have been coded in ATLaS SQL, and execute with a modest (20 40%) performance overhead with respect to the same applications written in C/ C++. This system can handle continuous queries using the schema and queries in Query Repository . Sarawagi, S. Thomas, and R. Agrawal [3] proposed integrating association rule mining with relational database systems. Integrating Association rule mining include several method. Loose - coupling through a SQL cursor interface is an encapsulation of a mining algorithm in a stored procedure. Second method is caching the data to a file system on-the-fly and mining tight-coupling using primarily user-defined functions and SQL implementations for processing in the DBMS. Loose-coupling and Stored-procedure architectures: For the loose-coupling and Stored-procedure architectures, can use the implementation of the Apriori algorithm for finding association rules.C. Ordonez [4] proposes an Integration of K-means clustering with a relational DBMS using SQL. This technique consists of three SQL implementations. First step is a straightforward translation of K-means computations into SQL, and an optimized version based on improved data organization, efficient indexing, sufficient statistics, and rewritten queries, and an incremental version that uses the optimized version as a building block with fast convergence and automated reseeding. The first implementation is a straightforward translation of K-means computations into SQL, which serves as a framework to build a second optimized version with superior performance. The optimized version is then used as a building block to introduce an incremental K-means implementation with fast convergence and automated reseeding. G. Graefe, U. Fayyad, and S. Chaudhuri [5] introduced efficient gathering of sufficient statistics for classification from large SQL Databases. This technique use a SQL operator (Unpivot) that enables efficient gathering of statistics with minimal changes to the SQL backend. Need a set of counts for the number of co-occurrences of each attribute value with each class variable. In classification the number of attribute values is not large (in the hundreds) the size of the counts table is fairly small. Continuous - valued attributes are discredited into a set of intervals. The most familiar selection measures used in classification do not require the entire data set, but only sufficient statistics of the data. A straightforward implementation for deriving the sufficient statistics on a SQL database results in unacceptably poor performance. The problem of optimizing queries with outer joins is not new. Optimizing joins by reordering operations and using transformation rules is studied. This work does not consider optimizing a complex query that contains several outer joins on primary keys only, which is fundamental to prepare data sets for data mining. Traditional query optimizers use a tree based execution plan, but the use of hyper-graphs to provide a more comprehensive set of potential plans. J. Gray, A. Bosworth, A. Lay man, and H. Pirahesh [6] proposed a relational aggregation operator that generalizing Group-By, Cross -Tab, and Sub-Totals. The cube operator generalizes the histogram, cross tabulation, roll-up, drill-down, and sub-total constructs. The cube operator can be imbedded in more complex non-procedural data analysis programs and data mining. The cube operator treats each of the N aggregation attributes as a dimension of N-space. The aggregate of a particular set of attribute values is a point in this space and the set of points forms an N-dimensional cube. Super-aggregates are computed by aggregating the N-cube to lower dimensional spaces. Creating a data cube requires generating the power set (set of all subsets) of the aggregation columns. Since the CUBE is an aggregation operation, it makes sense to externalize it by overloading the SQL GROUP BY operator. G. Luo, J.F. Naughton, C.J. Ellmann, and M. Watzke [7] proposed Immediate materialized view introduces many lock conflicts or deadlocks. System results in low level of concurrency and high level of deadlocks. To solve the materialized view update problem V-locks (View locks) augment with a value-based latch pool. Direct Propagate Updates propagate updates on base relations directly to the materialized view without computing any join operator. Granularity and the No-Lock Locking Protocol locks have some interesting properties with respect to granularity and concurrency .Finer granularity locking results in higher concurrency. In the no-lock locking protocol, like the V locking protocol, updaters of the materialized view must get X locks on the tuples in the base relations they update and S locks on the tuples in the other base relations mentioned in the view. Xiang Lian and Lei Chen [9] analyzed cost models for evaluating dimensionality reduction in high-dimensional Spaces. This model is general cost models for evaluating the query performance over the reduced data sets by GDR, LDR, and ADR, in light of which we introduce a novel (A) LDR method, Partitioning based on Randomized Search (RANS). Formal cost models to evaluate the effectiveness and efficiency of GDR, LDR, and ADR for range queries. Furthermore, we present a novel partitioning based (A) LDR approach, PRANS, which is based on our cost model and can achieve good query performance in terms of the pruning power. Extensive experiments have verified the correctness of our cost models and indicated that compared to the existing LDR method, can result in partitions with a lower query cost .C. Ordonez [10] introduced techniques to efficiently compute fundamental statistical models inside a DBMS exploiting User-Defined Functions (UDFs). Two summary matrices on the data set are mathematically shown to be essential for all models: the linear su m of points and the quadratic sum of cross products of points. Introduce efficient SQL queries to compute summary matrices and score the data set. Based on the SQL framework, introduce UDFs that work in a single table scan. Aggregate UDFs to compute summary matrices for all models and a set of primitive scalar UDFs are used to score data sets. C. Ordonez [11] proposed two SQL aggregate functions to compute percentages addressing many limitations. The first function returns one row for each percentage in vertical form and the second function returns each set of percentages adding 100% on the same row in horizontal form. These novel aggregate functions are used as to introduce the concept of percentage queries and to generate efficient SQL code in data min ing related works . Queries using percentage aggregations are called percentage queries . Two practical issues were identified when computing vertical percentage queries. First issue is missing rows and second issue is division by zero.

III. EXECUTION STRATEGIES IN HORIZONTAL AGGREGATION

Horizontal aggregations propose a new class of functions that aggregate numeric expressions and the result are transposed to produce data sets with a horizontal layout. The operation is needed in a number of data mining tasks, such as unsupervised classification and data summation, as well as segmentation of large heterogeneous data sets into smaller homogeneous subsets that can be easily managed, separately modeled and analyzed. To create datasets for data mining related works, efficient and summary of data are needed. For that this proposed system collect particular needed attributes from the different fact tables and displayed columns in order to create date in horizontal layout. Main goal is to define a template to generate SQL code combining aggregation and transposition (pivoting). A second goal is to extend the SELECT statement with a clause that combines transposition with aggregation. Consider the following GROUP BY query in standard SQL that takes a subset L$_1$, . . . , L$_m$ from $D_1, ..., D_p$:

SELECT $L_1, .., L_m, sum(A)$
$FROM\ F_1, F_2$
$GROUP\ BY\ L_1, L_m$;

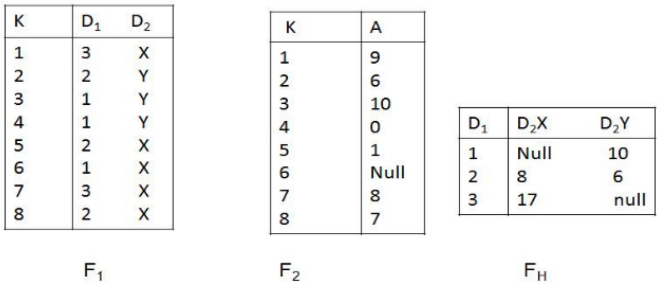
In a horizontal aggregation there are four input parameters to generate SQL code:

1) The input table F_1, F_2, F_n
2) The list of GROUP BY columns $L_1, ..., L_j$,
3) The column to aggregate (A),
4) The list of transposing columns $R_1, ..., R_k$.

This aggregation query will produce a wide table with m+1 columns (automatically determined), with one group for each unique combination of values $L_1, ..., L_m$ and one aggregated value per group (i.e., sum(A)). In order to evaluate this query the query optimizer takes three input parameters. First parameter is the input table F. Second parameter is the list of grouping columns $L_1, ..., L_m$. And the final parameter is the column to aggregate (A).

A. Example

In the Fig.1 there is a common field K in F_1 and F_2.In F_2, D_2 consist of only two distinct values X and Y and is used to transpose the table. The aggregate operation is used in this is sum (). The values with in D1 are repeated, 1 appears 3 times, for row 3, 4 and, and for row 3 & 4 value of D2 is X & Y. So D2X and D2Y is newly generated columns in F_H. Commonly using Query Evaluation methods in Horizontal aggregation functions [12] are:



K	D$_1$	D$_2$
1	3	X
2	2	Y
3	1	Y
4	1	Y
5	2	X
6	1	X
7	3	X
8	2	X

F$_1$

K	A
1	9
2	6
3	10
4	0
5	1
6	Null
7	8
8	7

F$_2$

D$_1$	D$_2$X	D$_2$Y
1	Null	10
2	8	6
3	17	null

F$_H$

Fig. 1: An example of Horizontal aggregation

1) **SPJ Method**: The SPJ method is based on only relational operators. The basic concept in SPJ method is to build a table with vertical aggregation for each resultant column. To produce Horizontal aggregation FH system must join all those tables. There are two sub-strategies to compute Horizontal aggregation .First strategy includes direct calculation of aggregation from fact table. Second one compute the corresponding vertical aggregation and store it in temporary table FV grouping by $LE_1, ..., LE_i$,$RI_1, ..., RI_j$ then F_H can be computed from F_V. To get FH system need n left outer join with n+1 tables so that all individual aggregations are properly assembled as a set of n dimensions for each group. Null should be set as default value for groups with missing combinations for a particular group.
INSERT INTO F_H
SELECT $F_0.LE_1, F_0.LE_2, ..., F_0.LE_j,, F_1.A, F_2.A, ..., F_n.A$
FROM F_0
LEFT OUTER JOIN F_1 ON $F_0.\ LE_1 = F_1.LE_1 and...and F_0.\ LE_j = F_1.LE_j$
LEFT OUTER JOIN F_2 ON $F_0.\ LE_1 = F_2.LE_1 and...and F_0.LE_j = F_2.LE_j$
. . . .
LEFT OUTER JOIN F_n ON $F_0.\ LE_1 = F_2.LE_1 and...and F_0.LE_j = F_n.LE_j$
It is easy to see that left outer join is based on same columns. This strategy basically needs twice I/O operations by doing updates rather than insertion.

2) **CASE method**:In SQL build -in case programming construct are available, it returns a selected value rather from a set of values based on Boolean expression. Queries for FH can be evaluated by performing direct aggregation form fact table F and at the same time rows are transposing to produce the FH.

SELECT DISTINCT RI_1 FROM F:
INSERT INTO F_H SELECT $LE_1, LE_2, ..., LE_j$,
V(CASE WHEN $RI_1 = v_{11} and...R_k = v_{k1} THENAELSEnullEND)$
..,
V(CASE WHEN $RI_1 = v_{1n} and...R_k = v_{kn} THENAELSEnullEND)$
FROM F GROUP BY $LE_1, LE_2, ..., LE_j$

3) **PIVOT method** Pivot transforms a series of rows into a series of fewer numbers of rows with additional columns Data in one source column is used to determine the new column for a row, and another source column

is used as the data for that new column. The wide form can be considered as a matrix of column values, while the narrow form is a natural encoding of a sparse matrix In current implementation PIVOT operator is used to calculate the aggregations. One method to express pivoting uses scalar sub queries. Each pivoted is created through a separate sub query. PIVOT operator provides a technique to allow rows to columns dynamically at the time of query compilation and execution.

SELECT *FROM (Bill Tab le PIVOT (SUM (A mount) for Month in (Jan,Feb,Mar)

This query generate table with jan,feb and mar as column attribute and the sum of the amount of particular customer that are stored inside the Bill Table. The pivot method is more efficient method than other two methods. Because the pivot operator internally calculates the aggregation operation and no need to create extra tables. So operation performed within this method is less compared to other methods.

IV. INTEGRATING K-MEANS ALGORITHM WITH HORIZONTAL AGGREGATION

Clustering methods partition a set of objects into clusters such that objects in the same cluster are more similar to each other than objects in different clusters according to some defined criteria. Data mining applications frequently involve categorical data. The biggest advantage of these clustering algorithms is that it is scalable to very large data sets.

Even though the existing system presented the computation of the values for different attributes, it has some drawbacks. In the research of the horizontal aggregation, the existing systems are not well defined for the different fact tables that need better indexing and extraction.

Multiple fact tables: Constructing new data sets within the range of a discrete set of known data points we need different attributes from different fact tables. In many applications one often has a number of data values, obtained by experimentation, which stored on limited number of databases. It is often required to extract the particular useful attributes from the different fact tables and perform aggregation.

K-means: K-means is initialized from some random or approximate solution. Each step assigns each point to its nearest cluster and then points belonging to the same cluster are averaged to get new cluster centroids. Each step successively improves cluster centroids until they are stable. This is the standard version of K-Means technique used. Optimized K-means computes all Euclidean distances for one point in one I/O, exploits sufficient statistics, and stores the clustering model in a single table. Experiments evaluate performance with large data sets focusing on elapsed time per iteration.

The main issue here addressed is how to make efficient indexing of horizontal aggregation. Initially an aggregation operation is performed horizontal layout are creating by using pivot operator. In this a k-means algorithm are implementing to create datasets with horizontal layout as input.

A. Algorithm Design

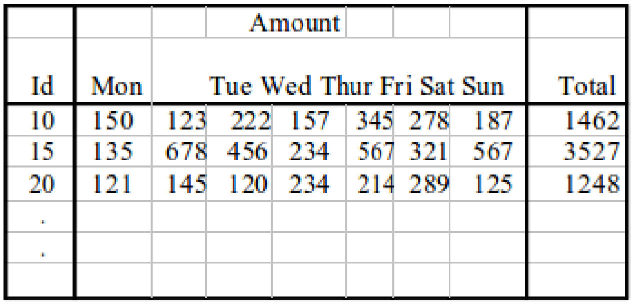
The algorithm is designed as follows: K-means algorithm based on classification technique uses horizontal aggregation as input. Pivot operator is used to calculate the aggregation of particular data values from distinct fact tables. Optimization provides for PIVOT for large number of fact table. The database connectivity and choosing different tables with .mdb extension is the first step in this system. Horizontal aggregation can be evaluated by choosing transpose column and aggregate operation .Pivot operator automatically transforms table to horizontal layout. This is the main advantage of this particular algorithm

The k-means algorithm is the best-known squared error based clustering algorithm with input as horizontal aggregation The algorithm consist of mainly four steps.1) Selection of the initial k means for k clusters from attribute of datasets obtained from horizontal aggregation operation.2) Calculation of the dissimilarity between an object and the Mean of a cluster.3) Allocation of an object to the cluster whose mean is nearest to the object.4) Re-calculation of the mean of a cluster from the objects allocated to it so that the intra cluster dissimilarity

B. Experimental Study

For the experimental studies data base file from Database are taken. Attributes having integer values along with aggregate operator are chosen. Most of the experiments are done in aggregated data set with attributes having many values . Firstly an algorithm was developed to find horizontal aggregation values for different attributes using PIVOT operator. Transposing columns and aggregation operation are chosen before the query generator. To systematically study the performance of the algorithm, calculate the aggregation manually and make sure that the query result and calculated result are same. Secondly similar algorithm using different database table is developed. Join operator is used to extract attributes from different tables. As next step, for attributes of horizontal aggregation i.e. having only integer values, a k-means clustering algorithm is developed and thereafter index for particular attributes are generated. By using clustering algorithm, system generated related aggregated result in one group and other related result

Table I. Dataset in Horizontal Layout



			Amount					
Id	Mon	Tue	Wed	Thur	Fri	Sat	Sun	Total
10	150	123	222	157	345	278	187	1462
15	135	678	456	234	567	321	567	3527
20	121	145	120	234	214	289	125	1248
.								
.								

in next group. For query optimization the distance computation and nearest cluster in the k-means are based on SQL.

V. Conclusions & future scope

This system extended the horizontal aggregations with k-means algorithm to cluster the aggregated column which help preparing datasets for data mining related work. Optimized k-means is significantly faster because of small data set run clustering outside the DBMS.Input to the system is data from multiple tables rather than single table used in traditional horizontal aggregation. Include Euclidean distance computation, pivoting a table to have one dimension value per row. Data manipulating operator Pivot is easy to compute for wide set of values. Pivot is an extension of Group By with unique restrictions and optimization opportunities, and this makes it easy to introduce incrementally on top of existing grouping implementation

In future, this work can be extended to develop a more formal model of evaluation methods to achieve better results. Also then we can be developing more complete I/O cost models .

References

[1] C. Cunningham, G. Graefe, and C.A. Galindo-Legaria. PIVOT and UNPIVOT: Optimizat ion and execution strategies in an RDBMS. In Proc. VLDB Conference, pages 9981009, 2004.

[2] H. Wang, C. Zaniolo, and C.R. Luo. ATLaS: A small but comp lete SQL extension for data mining and data streams. In Proc. VLDB Conference, pages 11131116, 2003.

[3] S. Sarawagi, S. Tho mas, and R. Agrawal. Integrating association rule mining with relational database systems: alternatives and implications. In Proc. ACM SIGMOD Conference, pages 343354, 1998.

[4] C. Ordonez. Integrating K-means clustering with a relational DBM S using SQL. IEEE Transactions on Knowledge and Data Engineering (TKDE), 18(2):188201, 2006.

[5] G. Graefe, U. Fayyad, and S. Chaudhuri. On the efficient gathering of sufficient statistics for classification fro m large SQL databases. In Proc. ACM KDD Conference, pages 204208, 1998.

[6] J. Gray, A. Bosworth, A. Lay man, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and subtotal. In ICDE Conference, pages 152 159,1996.

[7] G. Luo, J.F. Naughton, C.J. Ellmann, and M. Watzke. Locking protocols for materialized aggregate join views. IEEE Transactions on Knowledge and Data Engineering (TKDE) , 17(6):796807, 2005.

[8] C. Ordonez and S. Pitchaimalai. Bayesian classifiers programmed in SQL. IEEE Transactions on Knowledge and Data Engineering (TKDE), 22(1):139144, 2010.

[9] Xiang Lian, Student Member, IEEE, and Lei Chen, General Cost Models for Evaluating Dimensionality Reduction in High-Dimensional Spaces. IEEE Transactions on Knowledge and Data Engineering (TKDE) , 22(1):139144, 2010.

[10] C. Ordonez. Statistical model computation with UDFs. IEEE Transactions on Knowledge and Data Engineering (TKDE) , 22, 2010.

[11] C. Ordonez. Vertical and horizontal percentage aggregations. In Proc. ACM SIGMOD Conference, pages 866871, 2004.

[12] C. C. Ordonez and Zhibo Chen. Horizontal Aggregation in SQL to prepare Data Sets for Data Mining Analysis. . IEEE Transactions on Knowledge and Data Engineering (TKDE) , 1041- 4347/ 11/ $26.00 ,2011

Secure Technique to Block Misbehaving users in Anonymous Networks

Ashmy Antony,
Dept. of Computer Science and Engg,
SIMAT, Vavanoor, Palakkad
ashmyantony@simat.ac.in

***Abstract*—Anonymizing network is a network that allows the user to access internet privately, using a series of routers. The peculiarity of such networks is that, it allows the client to hide its IP address from the server. One of the limiting factors of anonymizing networks is that, the users of such networks misuse their anonymity for abusive purposes such as defacing a website. Website defacement is an attack that changes the visual appearance of a particular website. In ordinary networks, website administrators prevent a misbehaving user from accessing their website by blocking its IP address. This is not practical in case of anonymizing networks since the client is accessing a website anonymously without providing its IP address to the server. An alternative technique to prevent website defacement through an anonymizing network is to block all the exit nodes of the anonymizing network. This method is not feasible since it will be blocking both misbehaving users and genuine users. To overcome all these limitations, a secure system, based on the concept of nymble is presented. This system allows the server to blacklist a misbehaving user without compromising its anonymity. Server can blacklist users for whatever reason. The privacy of the blacklisted users is maintained. This system provides anonymous authentication where, a user can access web without prompting username and password. Proposed system is capable of performing fast authentication and it also provides revocation auditability where a user can check whether it is blacklisted or not.**

***Keywords*—Anonymous blacklisting, Privacy, Revocation.**

I. INTRODUCTION

ANOMALIZING networks such as Tor [18] route traffic through independent nodes in separate administrative domains to hide a clients IP address. Unfortunately, some users have misused such networksunder the cover of anonymity, users have repeatedly defaced popular Web sites such as Wikipedia. Since Web site administrators cannot blacklist individual malicious users IP addresses, they blacklist the entire anonymizing network. Such measures eliminate malicious activity through anonymizing networks at the cost of denying anonymous access to behaving users. In other words, a few bad apples can spoil the fun for all. (This has happened repeatedly with Tor.1). There are several solutions to this problem, each providing some degree of accountability. In pseudonymous credential systems [14], [17], [23], [28], users log into Web sites using pseudonyms, which can be added to a blacklist if a user misbehaves. Unfortunately, this approach results in pseudonymity for all users, and weakens the anonymity provided by the anonymizing network. Anonymous credential systems [10], [12] employ group signatures. Basic group signatures [1], [6], [15] allow servers to revoke a misbehaving users anonymity by complaining to a group manager. Servers must query the group manager for every authentication, and thus, lacks scalability. Traceable signatures [26] allow the group manager to release a trapdoor. that allows all signatures generated by a particular user to be traced; such an approach does not provide the backward unlinkability [30] that we desire, where a users accesses before the complaint remain anonymous. Backward unlinkability allows for what we call subjective blacklisting, where servers can blacklist users for whatever reason since the privacy of the blacklisted user is not at risk. In contrast, approaches without backward unlinkability need to pay careful attention to when and why a user must have all their connections linked, and users must worry about whether their behaviors will be judged fairly. Subjective blacklisting is also better suited to servers such as Wikipedia, where misbehaviors such as questionable edits to a Webpage, are hard to define in mathematical terms. In some systems, misbehavior can indeed be defined precisely. For instance, double spending of an e-coin is considered a misbehavior in anonymous e-cash systems [8], [13], following which the offending user is deanonymized. Unfortunately, such systems work for only narrow definitions of misbehaviorit is difficult to map more complex notions of misbehavior onto double spending or related approaches [32]. With dynamic accumulators [11], [31], a revocation operation results in a new accumulator and public parameters for the group, and all other existing users credentials must be updated, making it impractical. Verifier-local revocation (VLR) [2], [7], [9] fixes this shortcoming by requiring the server (verifier) to perform only local updates during revocation. Unfortunately, VLRrequires heavy computation at the server that is linear in the size of the blacklist. For example, for a blacklist with 1,000 entries, each authentication would take tens of seconds,2 a prohibitive cost in practice. In contrast, our scheme takes the server about one millisecond per authentication, which is several thousand times faster than VLR. These low overheads will incentivize servers to adopt such a solution when weighed against the potential benefits of anonymous publishing (e.g., whistle-blowing, reporting, anonymous tip lines, activism, and so on.).

II. SOLUTION

We present a secure system called Nymble, which provides all the following properties: anonymous authentication, backward unlinkability, subjective blacklisting, fast authentication speeds, rate-limited anonymous connections, revocation auditability (where users can verify whether they have been blacklisted), and also addresses the Sybil attack[19] to make its deployment practical. In Nymble, users acquire an ordered collection of nymbles, a special type of pseudonym, to connect toWebsites. Without additional information, these nymbles are computationally hard to link,4 and hence, using the stream of nymbles simulates anonymous access to services. Web sites, however, can blacklist users by obtaining a seed for a particular nymble, allowing them to link future nymbles from the same user those used before the complaint remain unlinkable. Servers can therefore blacklist anonymous users without knowledge of their IP addresses while allowing behaving users to connect anonymously. Our system ensures that users are aware of their blacklist status before they present a nymble, and disconnect immediately if they are blacklisted. Although our work applies to anonymizing networks in general, we consider Tor for purposes of exposition. In fact, any number of anonymizing networks can rely on the same Nymble system, blacklisting anonymous users regardless of their anonymizing network(s) of choice.

III. CONTRIBUTIONS OF THIS PAPER

Our research makes the following contributions: . Blacklisting anonymous users. We provide a means by which servers can blacklist users of an anonymizing network while maintaining their privacy. Practical performance. Our protocol makes use of inexpensive symmetric cryptographic operations to significantly outperform the alternatives. Open-source implementation. With the goal of contributing a workable system, we have built an open-source implementation of Nymble, which is publicly available.5 We provide performance statistics to show that our system is indeed practical. Some of the authors of this paper have published two anonymous authentication schemes, BLAC [33] and PEREA [34], which eliminate the need for a trusted third party for revoking users. While BLAC and PEREA provide better privacy by eliminating the TTP, Nymble provides authentication rates that are several orders of magnitude faster than BLAC and PEREA (see Section 6). Nymble thus represents a practical solution for blocking misbehaving users of anonymizing networks. We note that an extended version of this paper is available as a technical report [16].

IV. OVERVIEW OF ARCHITECTURE

We now present a high-level overview of the Nymble system, and defer the entire protocol description and security analysis to subsequent sections.

A. Resource-Based Blocking

To limit the number of identities a user can obtain (called the Sybil attack [19]), the Nymble system binds nymbles to resources that are sufficiently difficult to obtain in great numbers. For example, we have used IP addresses as the resource in our implementation, but our scheme generalizes to other resources such as email addresses, identity certificates, and trusted hardware. The address the practical issues related with resource-based blocking in Section 8, and suggest other alternatives for resources.

B. The Pseudonym Manager

The user must first contact the Pseudonym Manager (PM) and demonstrate control over a resource; for IP-address blocking,the user must connect to the PM directly (i.e., not through a known anonymizing network), as shown in Fig. 1. Its assume that the PM has knowledge about Tor routers, for example, and can ensure that users are communicating with it directly.6 Pseudonyms are deterministically chosen based on the controlled resource, ensuring that the same pseudonym is always issued for the same resource. Note that the user does not disclose what server he or she intends to connect to, and the PMs duties are limited to mapping IP addresses (or other resources) to pseudonyms. The user contacts the PM only once per linkability window (e.g., once a day).

C. The Nymble Manager

After obtaining a pseudonym from the PM, the user connects to the Nymble Manager (NM) through anonymizing network, and requests nymbles for access to a particular server (such as Wikipedia). A users requests to the NM are therefore pseudonymous, and nymbles are generated using the users pseudonym and the serversidentity. These nymbles are thus specific to a particular user-server pair. Nevertheless, as long as the PM and the NM do not collude, the Nymble system cannot identify which user is connecting to what server; the NM knows only the pseudonym-server pair, and the PM knows only the user identity-pseudonym pair. To provide the requisite cryptographic protection and security properties, the NM encapsulates nymbles within nymble tickets. Servers wrap seeds into linking tokens, and therefore, we will speak of linking tokens being used to link future nymble tickets. The importance of these constructs will become apparent as we proceed.

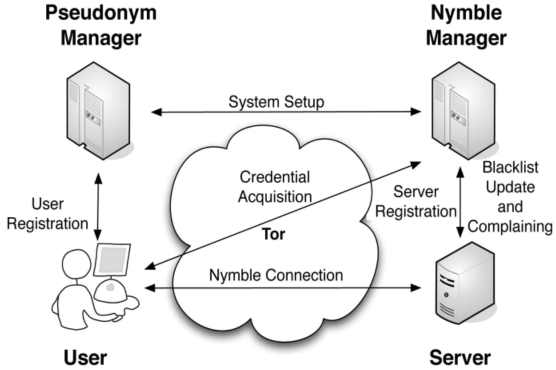


Fig. 1: The Nymble system architecture showing the various modes of interaction.

D. Time

Nymble tickets are bound to specific time periods. As illustrated in Fig. 2, time is divided into linkability windows of duration W, each of which is split into L time periods of duration T (i.e., W 14L _ T). We will refer to time periods and linkability windows chronologically as t1; t2; . . . ; tL and w1; w2; . . . , respectively. While a users access within a time period is tied to a single nymble ticket, the use of different nymble tickets across time periods grants the user anonymity between time periods. Smaller time periods provide users with higher rates of anonymous authentication, while longer time periods allow servers to rate-limit the number of misbehaviors from a particular user before he or she is blocked. For example, T ould be set to five minutes, and W to one day (and thus, L 14 288). The linkability window allows for dynamism since resources such as IP addresses can get reassigned and it is undesirable to blacklist such resources indefinitely, and it ensures forgiveness of misbehavior after a certain period of time. We assume all entities are time synchronized (for example, with time.nist.gov via the Network Time Protocol (NTP)), and can thus calculate the current linkability window and time period.

E. Blacklisting a User

If a user misbehaves, the server may link any future connection from this user within the current linkability window (e.g., the same day). Consider Fig. 2 as an example: A user connects and misbehaves at a server during time period t_ within linkability window w_. The server later detects this misbehavior and complains to the NM in time period tc (t_ < tc _ tL) of the same linkability window w_. As part of the complaint, the server presents the nimble ticket of the misbehaving user and obtains the corresponding seed from the NM. The server is then able to link future connections by the user in time periods tc; tc 1; . . . ; tL of the same linkability window w_ to the complaint. Therefore, once the server has complained about a user, that user is blacklisted for the rest of the day, for example (the linkability window). Note that the users connections in t1; t2; . . . ; t_; t_ 1; . . . ; tc remain unlinkable (i.e., including those since the misbehavior and until the time of complaint). Even though misbehaving users can be blocked from making connections in the future, the users past connections remain unlinkable, thus providing backward unlinkability and subjective blacklisting.

F. Notifying the User of Blacklist Status

Users who make use of anonymizing networks expect their connections to be anonymous. If a server obtains a seed for that user, however, it can link that users subsequent connection. It is of utmost importance then that users be notified of their blacklist status before they present a nimble ticket to a server. In our system, the user can download the servers blacklist and verify her status. If blacklisted, the user disconnects immediately. Since the blacklist is cryptographically signed by the NM, the authenticity of the blacklist is easily verified if the blacklist was updated in the current time period m (only one update to the blacklist per time period is allowed).

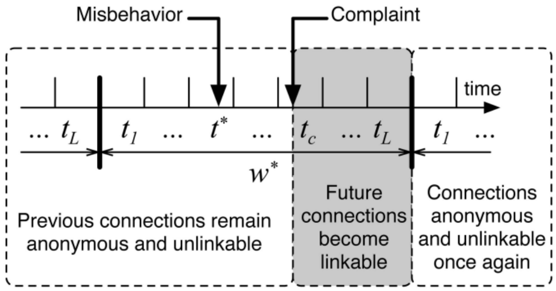

Misbehavior
Complaint
time
... t_L
t_1 ...
t^*
...
t_c ... t_L
t_1 ...
w^*
Previous connections remain anonymous and unlinkable
Future connections become linkable
Connections anonymous and unlinkable once again

Fig. 2: The life cycle of a misbehaving user. If the server complains in time period tc about a users connection in t_, the user becomes linkable starting in tc. The complaint in tc can include nymble tickets from only tc_1 and earlier.

If the blacklist has not been updated in the current time period,theNMprovides servers with daisies every time period so that users can verify the freshness of the blacklist (blacklist from time period told is fresh as of time period tnow). As discussed in Section 4.3.4, these daisies are elements of a hash chain, And provide a lightweight alternative to digital signatures. Using digital signatures and daisies, we thus ensure that race conditions are not possible in verifying the freshness of a blacklist. A user is guaranteed that he or she will not be linked if the user verifies the integrity and freshness of the blacklist before sending his or her nymble ticket.

V. PRELIMINARIES

Notation The notation a 2R S represents an element drawn uniformly at random from a nonempty set S.NN0 is the set of nonnegative integers, andNNis the setNN0nf0g. s12i_ is the ith element of list s. skt is the concatenation of (the unambiguous encoding of) lists s and t. The empty list is denoted by ;. We sometimes treat lists of tuples as dictionaries. For example, if L is the list ((Alice, 1234), (Bob, 5678)), then L[Bob] denotes the tuple (Bob, 5678). If A is an (possibly probabilistic) algorithm, then Ax denotes the output when A is executed given the input x. a :14 b means that b is assigned to a.

VI. CRYPTOGRAPHIC PRIMITIVES

If Nymble uses the following building blocks (concrete instantiations are suggested in Section 6) Secure cryptographic hash functions. These are oneway and collision-resistant functions that resemble random oracles [5]. Denote the range f the hash functions by H. Secure message authentication (MA) [3]. These consist of the key generation (MA.KeyGen), and the message authentication code (MAC) computation (MA.Mac) algorithms. Denote the domain of MACs by M.

Secure symmetric-key encryption (Enc) [4]. These consist of the key generation (Enc.KeyGen), encryption (Enc.Encrypt), and decryption (Enc.Decrypt) algorithms. Denote the domain of ciphertexts by _. Secure digital signatures (Sig) [22]. These consist of the key generation (Sig.KeyGen), signing (Sig.Sign), and verification (Sig.Verify) algorithms. Denote the domain of signatures by _.member list.

VII. DATASTRUCTURES

Nymble uses several important data structures:

A. Pseudonyms

The PM issues pseudonyms to users. A pseudonym pnym has two components nym and mac: nym is a pseudorandom mapping of the users identity (e.g., IP address),7 the linkability window w for which the pseudonym is valid, nd the PMs secret key nymKeyP ; mac is a MAC that the NM uses to verify the integrity of the pseudonym. Algorithms 1 and 2 describe the procedures of creating and verifying pseudonyms.

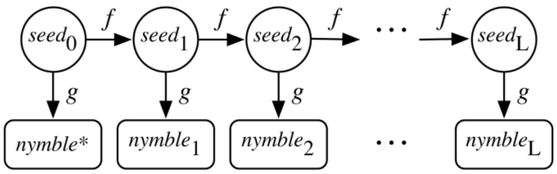


Fig. 3: Evolution of seeds and nymbles

B. Seeds and Nymbles

A nymble is a pseudorandom number, which serves as an identifier for a particular time period. Nymbles (presented by a user) across periods are unlinkable unless a server has blacklisted that user. Nymbles are presented as part of a nymble ticket, as described next. As shown in Fig. 4, seeds evolve throughout a linkability window using a seed-evolution function f; the seed for the next time period (seednext) is computed from the seed for the current time period (seedcur) as seednext 14 fseedcur: The nymble (nymblet) for a time period t is evaluated by applying the nymble evaluation function g to its corresponding seed (seedt), i.e. nymblet 14 gseedt: The NM sets seed0 to a pseudorandom mapping of the users pseudonym pnym, the (encoded) identity sid of the server (e.g., domain name), the linkability window w for which the seed is valid, and the NMs secret key seedKeyN. Seeds are therefore specific to user-server-window combinations As a consequence, a seed is useful only for a particular server to link a particular user during a particular linkability window. In our Nymble construction, f and g are two distinct cryptographic hash functions. Hence, it is easy to compute future nymbles starting from a particular seed by applying f and g appropriately, but infeasible to compute nymbles otherwise. Without a seed, the sequence of nymbles appears unlinkable, and honest users can enjoy anonymity. Even when a seed for a particular time period is obtained, all the nymbles prior to that time period remain unlinkable.

C. Nymbles and ticket credentials

A credential contains all the nymble tickets for a particular linkability window that a user can present to Particular server. Algorithm 3 describes the following procedure of generating a credential upon request: A ticket contains a nymble specific to a server, time period, and linkability window. ctxt is encrypted data that the NM can use during a complaint involving the nymble ticket. In particular, ctxt contains the first nymble (nymble_) in the users sequence of nymbles, and the seed used to generate that nymble. Upon a complaint, the NM extracts the users seed and issues it to the server by evolving the seed, and nimble helps the NM to recognize whether the user has already been blacklisted.

The MACs macN and macNS are used by the NM and the server, respectively, to verify the integrity of the nimble ticket, as described in Algorithms 4 and 5. As will be explained later, the NM will need to verify the tickets integrity upon a complaint from the server.

D. Blacklists

A servers blacklist is a list of nymble_s corresponding to all the nymbles that the server has complained about. Users can quickly check their blacklisting status at a server by checking to see whether their nymble_ appears in the servers blacklist (see Algorithm 6).

E. Blacklist integrity

It is important for users to be able to check the integrity and freshness of blacklists, because, otherwise, servers could omit entries or present older blacklists and link users without their knowledge. The NM signs the blacklist (see Algorithm 7), along with the server identity sid, the current time period t, current linkability window w, and target (used for freshness, explained soon),using its signing key signKeyN. As will be explained later, during a complaint procedure, the NM needs to update the servers blacklist, and thus needs to check the integrity of the blacklist presented by the server. To make this operation more efficient, the NM also generates an MAC using its secret key macKeyN (line 7). At the end of the signing procedure, the NM returns a blacklist certificate (line 7), which contains the time period for which the certificate was issued, a daisy (used for freshness, explained soon), mac, and sig. Algorithms 8 and 9 describe how users and the NM can verify the integrity and freshness of blacklists. Blacklist freshness. If the NM has signed the blacklist for the current time period, users can simply verify the digital signature in the certificate to infer that the blacklist is both valid (not tampered with) and fresh (since the current time period matches the timeperiod in The blacklist certificate). To prove the freshness of blacklists every time period, however, the servers would need to get the blacklists digitally signed every time period, thus imposing a high load on the NM. To speed up this process, we use a hash chain [20], [29] to certify that blacklist from time period t is still fresh.

VIII. OUR NYMBLE CONSTRUCTION

A. System Setup

1) The NM executes NMInitState (see Algorithm 10) and initializes its state nmState to the algorithms output.
2) The NM extracts macKeyNP from nmState and sends it to the PM over a type-Auth channel. macKeyNP is a

shared secret between the NM and the PM, so that the NM can verify the authenticity of pseudonyms issued by the PM.

3) The PM generates nymKeyP by running Mac.KeyGen() and initializes its state pmState to the pairnymKeyP macKeyNP .
4) The NM publishes verKeyN in nmState in a way that the users in Nymble can obtain it and verify its integrity at any time (e.g., during registration).

B. Server Registration

To participate in the Nymble system, a server with identity sid initiates a type-Auth channel to the NM, and registers with the NM according to the Server Registration protocol below. Each server may register at most once in any linkability window.

C. User Registration

A user with identity uid must register with the PM once in each linkability window. To do so, the user initiates a type-Basic channel to the PM, followed by the User Registration protocol described below. The PM checks if the user is allowed to register. In our current implementation, the PM infers the registering users IP address from the communication channel, and makes sure that the IP address does not belong to a known Tor exit node. If this is not the case, the PM terminates with failure. Otherwise, the PM reads the current linkability window as wnow, and runs pnym :14 PMCreate Pseudonympm Stateuid;wnow: The PM then gives pnym to the user, and terminates with success. The user, on receiving pnym, sets her state usrState to pnym; ;, and terminates with success.

D. Credential Acquisition

To establish a Nymble connection to a server, a user must provide a valid ticket, which is acquired as part of a credential from the NM. To acquire a credential for server sid during the current linkability window, a registered user initiates a type-Anon channel to the NM, followed by the Credential Acquisition protocol below.The user extracts pnym from usrState and sends the pair pnym; sid to the NM. The NM reads the current linkability window as wnow. It makes sure the users pnym is valid M Verify Pseudonym returns false, the NM terminates with failure; it proceeds otherwise.The NM runs NM Create Credentialnm Statepnym which returns a credential cred. The NM sends cred to the user and terminates with success.The user, on receiving cred, creates usrEntry:14 sid; cred; false, appends it to its state usrState, and terminates with success.

E. Nymble Connection Establishment

To establish a connection to a server sid, the user initiates a type-Anon channel to the server, followed by the Nymble connection establishment protocol described below. Blacklist Validation

1) The server sends hblist; certi to the user, where blist is its blacklist for the current time period and cert is the certificate on blist. (We will describe how the server can update its blacklist soon.)
2) The user reads the current time period and linkability window as tU now and wU now and assumes these values to be current for the rest of the protocol.
3) For freshness and integrity, the user checks if Verify BLusrState_sid; tU now; wU now; blist; cert_ 14 true If not, she terminates the protocol with failure.

F. Blacklisting a User

For each user a unique seedkey is generated and it is stored in the BL if they misbehave On misbehavior, server send complaint token to NM Token contains nymble ticket of the user at the time of misbehavior NM extract ctxt from ticket to obtain seed NM send seed to server & it update the BL Server compute future seeds and future tickets of the misbehaved user, & prevent user from accessing with these tickets. NM has the right to access and update BL. On the next attempt to connect to server, user has to first obtain tickets from NM NM will check BL and ensure whether seedkey of user is in BL If so user status will be shown as black listed User disconnect automatically without connecting to server

IX. Conclusions

We have proposed and built a comprehensive credential system called Nymble, which can be used to add a layer of accountability to any publicly known anonymizing network. Servers can blacklist misbehaving users while maintaining their privacy, and we show how these properties can be attained in a way that is practical, efficient, and sensitive to the needs of both users and services. We hope that our work will increase the mainstream acceptance of anonymizing networks such as Tor, which has, thus far, been completely blocked by several services because of users who abuse their anonymity.

References

[1] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, A Practical and Provably Secure Coalition-Resistant Group Signature, Scheme, Proc. Ann. Intl Cryptology Conf. (CRYPTO), Springer, pp. 255-270, 2000

[2] M. Bellare, R. Canetti, and H. Krawczyk, Keying Hash Functions for Message Authentication, Proc. Ann. Intl Cryptology Conf.(CRYPTO), Springer, pp. 1-15, 1996

[3] T. Nakanishi and N. Funabiki, Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps, Proc. Intl Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), Springer, pp. 533-548, 2005.

[4] LNguyen,Accumulators from Bilinear Pairings and Applications,Proc. Cryptographers Track at RSA Conf. (CT-RSA), Springer pp. 275-292 2005.

[5] P.P. Tsang, M.H. Au, A. Kapadia, and S.W. Smith, Blacklistable Anonymous Credentials: Blocking Misbehaving Users without TTPs, Proc. 14th ACM Conf. Computer and Comm. Security (CCS 07), pp. 72-81, 2007.

Negative Representations of Information

Rabeena T M,
Dept. of Computer Science and Engg,
SIMAT, Vavanoor, Palakkad
rabeena.tm@simat.ac.in

Abstract—**Data can be positive or negative. Database store positive data(original information). Negative Database brings about the concept of exact negative representations of information, discuss some possible implementations, and explore its attributes and applications. The concept is summarized by the phrase ”everything except....” What follows-the exceptions-are the negative image of the idea being conveyed. That is, NDB store everything except data of interest(negative data). Here a compact representation is used to accomplish this, and discuss the properties of the arrangement. A negative representation can be used to constrain the knowledge gained regarding the positive image because the amount of information per item is generally lower in a negative representation and that the way in which answers are inferred using a positive or negative set is fundamentally different. Some operations that take advantage of this change of perspective and help address some of the privacy concerns of the day.**

Keywords—**Negative Database(NDB).**

I. INTRODUCTION

INFORMATION is and always has been a valuable commodity. It reduces the uncertainty about a specific domain and facilitates the achievement of objectives. With the advent of computers and digital storage devices, the amount of information that is being collected and that can potentially be exploited has grown dramatically. Data is being gathered from every imaginable source and ranges from the scientific (e.g. genome sequences, particle collision streams etc.) to the sociological, where information about individuals such as their demographics, preferences, and spending habits are being amassed. The nature and volume of data pose novel challenges in terms of how it should be used. Central to this, is the question of how it is to be represented, as data representation has immense impact on how it can be utilized. Take, for instance, the data concerning hourly fluctuations in the price of some stock. Suppose a year's worth of data is at our disposal for analysis. What we can feasibly learn from these data depends at least on whether we have a printout or a digital representation. With the latter, a computer can be readily used to produce aggregate and statistical values, plot graphs and charts, and search the data for patterns, whereas a paper printout will render all such examinations impractical.

The importance of data representation is well understood in the sciences like cryptography where the aim is to find data representations from which meaningful information can only be extracted when knowledge of the secret used to create them is provided.

This report is about a new way to represent data; one in which everything except the items of interest is depicted. Information expressed in this way is referred to as negative information. The concept is a familiar one; consider for instance a political speech or an activity report: Omissions are often considered as compelling (substantial) as the items that are actually discussed (the phrase ”reading between the lines” suggests there is something being deliberately omitted that is of importance to the discourse). Artists sometimes portray everything but the subject of interest as a means to more meaningfully convey a message, and statisticians often use the information of everything but the subset of interest as a more tractable way to compute some value (e.g. one minus the probability of something not happening is the probability that it occurs). An example of special significance, one that inspired the present work, comes from the field of immunology, in particular, the method by which pathogens are identified; the immune system keeps a ”negative” image of self-the normal constituents of the body-and uses it to recognize foreign material.

Examples are plentiful and ubiquitous; however, this report focuses on a specific domain, one amenable to rigorous analysis, and one which studies the following: How can information be represented negatively; what are some of the properties of a negative encoding; and, how can these properties be used to our advantage. For instance, you are asked for all the information you have in your address book because there are some specific items a third party wants to verify. You have reservations about distributing your entire directory, and decide to turn in an alternative address book containing all the names, addresses, etc. that are not in your ”confidential” black book. Strictly speaking you have not been deceitful since both books contain the same information, but how can someone make use of it? What questions can be asked of it in an efficient manner? How voluminous is this so called address book, and how long did it take you to generate it? Finally, let us suppose this complementary address book does not have the same information content and is missing some of the ”negative” data. Is it still useful? What can you infer from the answers it provides? This report addresses questions such as these, studying the feasibility of negative representations, drawing out their distinctive properties and discussing their potential benefits for security and privacy.

Section 2 reviews related work, specifically in the fields of artificial immune systems, databases and areas associated with safeguarding sensitive information. Section 3 studies how the negative image of a set of strings can be represented exactly; it introduces negative databases as a means to this end, and provides algorithms for creating and updating them efficiently. The Section examines some distinctive properties of negative databases; in particular, it shows that they are easy to create, easy to query in certain ways, but that it is very hard to derive their entire positive image, making them a natural candidate for certain privacy preserving operations.NDB creation by using Prefix algorithm is covered in section 4.Section 5 clearly explains the different operators on NDB and their applications.Querying concepts are covered in Section 6.Another method for creating NDB which is hard-to-reverse is illustrated in section 7.An application of distributed NDB can be viewed in section 8.Differences between DB and NDB are explained in section 9 before conclusion.The final Section concludes the report summarizing the representations and providing avenues for future work.

II. Related Works

This section reviews some of the work done in three major areas that relate to the present work: Artificial Immune Systems (AIS), Databases, Data security and privacy. Other disciplines may come into play, such as information theory and statistics, but here we concentrate on the fields that have tackled related problems and/or that provide significant inspiration.

AIS is reviewed since it has been the primary source of inspiration and because the findings herein will be translated into contributions to that research area. One goal of this report is to establish a connection between databases and the representations presented herein, for this reason a concise account of some of the relevant database concepts and resources is given. Finally, a review of techniques for safeguarding sensitive data is given to illustrate some of the distinctive characteristics of the schemes proposed herein.

The Immune System and Artificial Immune Systems:
Studying biological systems has often been recourse for solving engineering problems. Recently (1990's) the immune system (IS) has come under focus for this purpose, giving birth to the field of artificial immune systems (AIS). Here, discuss the aspects of the adaptive IS that have served as inspiration to this field, and a brief review of the systems that have resulted from these insights. This is not a complete review of immunology nor is it a complete account of the AIS field.

One of the primary functions, of the Immune System (IS) is to keep the organism healthy, in particular to prevent and defuse illnesses induced by foreign agents known as pathogens. The task of identifying a pathogen is complicated by the possibility of never having encountered it before and by the fact that pathogens are subject to evolution (driven in part to avoid recognition by the IS) and change their form. One strategy thought to be employed by the IS to discriminate between self-the normal ”components” of the organism-and nonself-everything else, including potential pathogens. The theory is known as the self-nonself discrimination paradigm and forms the basis of much AIS related work.

The main focus of the AIS community has been on the adaptive IS of which lymphocytes, T-cell and B-cells, are primary actors. Lymphocytes are distributed throughout the body and are individually capable of binding foreign antigen. The collective actions of lymphocytes ultimately determine whether a bound antigen will be eliminated or not. AIS are thus composed of a collection of agents, that mimic lymphocytes, known as detectors.

One of the most popular applications of AIS has been to intrusion detection, owing to the parallel between protecting a computer from intruders and protecting an organism from pathogens. In addition, AIS have been used for varied applications such as color image classification, fault detection, recommender systems and even some hardware implementations.

Particularly relevant to the present work are the representation of self, nonself and immune cells as binary strings and the use of partial matching to establish detection.

The work proposed here shares with the above approaches the self vs. nonself distinction, the representation of these sets as fixed length strings, and the use of detectors to recognize members of such sets. However, this work is more concerned with the theoretical aspects and tradeoffs between representing data positively and negatively and with studying the relationship of these representations with database theory.

Databases:
One of the objectives of this work is to construct a connection between IS/AIS and database theory. Database theory has studied how, in a relational scheme, a table (a set of records) can be decomposed or broken up into sub tables while preserving certain properties. Decomposing a table requires knowledge about the semantics of the data such as dependencies among attributes. In the case where such information is not known a priori, data mining techniques such as association mining and frequent pattern discovery can provide useful guidance.

Another major goal of the present research is to study the feasibility and properties of representing a database negatively, how regular operations such as updates are to take place and what the query system will look like. The present work introduces algorithms for creating, querying and updating negative databases.

Sensitive Data:
The main object of this report is to study the properties of negative representations of data. Storing the negative image of a set rather than the set itself immediately suggests that such a strategy might be useful for protecting sensitive information, being that it is everything but the data we care about which is being stored.

An obvious starting point for protecting sensitive data is the large body of work on cryptographic methods. Some researchers have investigated how to combine cryptographic methods with databases, for example, by encrypting each record with its own key. These techniques, however, are intended to conceal all information about the encrypted data, and it is therefore not appropriate to situations in which some queries should be efficiently supported without revealing the entirety of the records.

Negative databases as described in Chapter 3 have the property that it is NP-hard to recover their positive image, i.e. the items of interest. By representing data negatively, as described in Chapter 3, a single message has many possible encodings, an idea that is also exploited by probabilistic encryption. Multi-party computation schemes, in which complex operations across databases can be performed privately are related to the applications discussed in this report, in particular when they involve operations such as set intersection.

Secret sharing is a technique whereby data is protected by splitting it into several pieces. This primitive is related to the setup of chapter 8 where the privacy of certain operations relies on negative information being divided into several subsets. Also relevant to the discussion of Chapter 8 is the area of private information retrieval which focuses on protecting the privacy of the entities consulting the database, rather than the contents of the database itself. The applications of negative information outlined in chapter 3 are also related to query restriction where the query language is designed to support only the desired classes of queries. Although query restriction controls access to the data by outside users, it cannot protect an insider with full privileges from inspecting individual records to retrieve information. In summary, the existence of sensitive data requires some method for controlling access to individual records. The overall goal is that the contents of a database be available for appropriate analysis and consultation without revealing information inappropriately. Satisfying both requirements usually entails some compromise, such as degrading the detail of the stored information, limiting the power of queries, or database encryption.

III. Exact Representation

Negative Database(NDB) store everything except the data of interest(negative data).It is a condensed representation of the compliment of DB. It is used to constrain the knowledge gained regarding the positive image(original data). Only a limited set of information is available to the external user, about the original data. In order to create a database NDB that is reasonable in size, it is necessary to compress the information contained in U-DB. To this end an additional symbol is introduced, known as a ”don’t care ”,written as ” * ”. The entries in NDB will thus be l-length strings over the alphabet {0, 1, *}. The don’t-care symbol has the usual interpretation, matching either a one or a zero at the bit position where the * appears. Positions in a string that are set either to one or zero are referred to as ”defined” or ”specified” positions, and locations where a * appears are referred to as ”unspecified” positions. With this new symbol large subsets of U-DB can be represented with just a few entries. For example, the set of strings U-DB can be exactly represented by the NDB set shown below: in which U consists of all binary strings of length 3 and DB is defined as 000, 111 .

DB	U-DB	NDB
000	001	0*1
111	010	*10
	011	10*
	100	
	101	
	110	

The convention is that a binary string s is taken to be in DB if and only if it fails to match each of the entries in NDB. This condition is fulfilled only if for every string tj NDB, s disagrees with tj in at least one defined position.

IV. NDB Creation

There are several algorithms for creating a negative database given as input DB. The main differences between these algorithms are the size of the resulting NDB and the ease with which DB can be retrieved from NDB.The first algorithm developed for this purpose, prove that negative databases can be created in polynomial time, that is in reasonable time and of reasonable size, is the Prefix Algorithm. The prefix algorithm introduced here is deterministic and reversible, which has consequences for the kinds of inferences that can be made efficiently from NDB.This algorithm creates a NDB in $O(l\|DB\|)$ time with $O(l\|DB\|)$ entries. A salient feature of these NDBs is that every string in these is matched by a single negative record and that some strings in DB can be retrieved using only a small subset of NDB can be recovered in polynomial time using all of NDB,refer [4].

A. Prefix Algorithm:

Let wi denote an i-bit prefix and Wi a set of i-length bit patterns.

```
1) $i \leftarrow 0$
2) Set Wi to the empty set.
3) Set Wi+1 to every pattern not present in DB’s wi+1 but
   with prefix in Wi.
4) For each pattern Vp in Wi+1
5) Create a record using Vp as its prefix and the remaining
   positions set to the don’t care symbol.
6) Add record to NDB.
7) Increment i by one
8) Set Wi to every pattern in DB’s wi
9) Return to step 3 as long as $i \leq l$.
```

1) Sample Creation: Consider a universe U consist of 16 elements,0000–1111. DB stores only 4 elements. So U-DB consists of all the elements in U which are not in DB. That is the negative image of DB is in U–DB. NDB is the compressed form of U–DB. Here the NDB is created by Prefix algorithm. **Through the algorithm:**

1) Start with $i = 1$.
   a) Bit combinations possible by one bit is only 0 and 1.
   b) Check the first bit of every records in DB.
   c) 0 and 1 are there in DB, so there is no need to insert these in NDB.
2) Increment i by 1.
   a) Bit combinations possible by two bits are 00,01,10,11.
   b) Check the first two bits of every records in DB.
   c) 11 is not in DB. So insert 11** in NDB.
3) Increment i by 1.

   a) Bit combinations are 000–111
   b) Check first three bits of every records in DB.
   c) 001,011,110,111 are not in DB, so insert in NDB as 001*,011*. 110* and 111* is already in NDB as 11**

4) Increment i by 1, $i = 4$, now it equals l,the bit length.
   a) Do like the previous steps.
   b) Insert corresponding elements in NDB.

5) Prefix coded NDB is now obtained.

Table shown below is the final NDB.

DB	U-DB	NDB
0001	0000	11**
0100	0010	001*
1000	0011	011*
1011	0101	0000
	0110	0101
	0111	1001
	1001	1010
	1010	
	1100	
	1101	
	1110	
	1111	

Col 1:Example DB, Col 2:Corresponding U–DB and Col 3:Corresponding NDB generated by prefix algorithm.

The NDB produced by the prefix algorithm has some interesting properties. For example, each record of NDB uniquely covers a subset of U–DB. This nonoverlapping property allows NDB to support more powerful queries than simple membership. Questions like ”Are there any engineers in DB?” can be answered by finding all records that match ”engineer” in the corresponding field of NDB and simply counting whether these records completely represent the subset of U that contains the engineers.

V. Operators And Their Applications

What if the created DB changes over time and its changes are to be reflected in NDB? In [2] three basic operations are presented. They are Insert, Delete and Morph. In general, insert to NDB means, it should be removed from DB and delete from NDB results in insertion of that element to DB.

A. Insertion

Insert takes as input a NDB and string x in $\{0, 1\}^l$ and outputs a NDB' that matches string matched by NDB and every string matched by x. The purpose of the insert operation is to introduce a subset of strings into the negative database while safeguarding its irreversibility properties. Lines 1 and 2 enable the procedure to create several entries in NDB portraying x. Likewise steps 3 and 4 set some of the unspecified positions of x (if any) so that it may be possible for a set of strings representing x, that exhibit bits not found in x, to be entered in NDB. Finally the call to Negative Pattern Generate produces y that represents x; y is then inserted in NDB.

Insert(x,NDB)

```
1) Randomly choose $1 \leq j \leq l$
2) for $k = 1$ to j do
3)    Randomly select from x at most n distinct unspecified bit positions
4)    for every possible bit assignment Bp of the selected positions
5)       $x^{|} \leftarrow x \times Bp$
6)       $y \leftarrow Negative - Pattern - Generate\{x^{|}, NDB\}$
7)       add y to NDB.
```

Negative-Pattern-Generate(x, NDB)

```
1) Create a random permutation $\pi$.
2) for all specified bits bi in $\pi(x)$,
3)    Let x' be the same as $\pi(x)$ but with bi flipped.
4)    if x' is subsumed by some string in $\pi(NDB)$,
5)       $\pi(x) \leftarrow \pi(x)$ -ith bit (set value to *)
6)    Keep track of the ith bit in a set indicator vector (SIV).
7) Randomly choose $0 \leq t \leq \|SIV\|$
8) $R \leftarrow t$ randomly selected bits from SIV.
9) Create a pattern Vk using $\pi(x)$ and the bits indicated by R.
10) Return $\pi'(Vk)$ where $\pi'$ is the inverse permutation of $\pi$
```

Negative-Pattern-Generate take as input a string x defined over 0, 1,* and a database NDB and outputs a string that matches x and nothing else outside of NDB.

Example: Insertion of 1*1 in NDB is shown below.

DB	NDB	DB-1*1	NDB+1*1
000	01*	000	1**
101	100		01*
111	001		001*
	110		

B. Deletion

Delete takes the same input as insert but outputs a NDB' that matches every string matched by NDB minus those matched by x. This operation aims to remove a subset of strings from being represented in NDB. This operation cannot simply be implemented by looking for a particular entry in NDB and removing it, since it may be the case that a string is represented by several entries in NDB and an entry in NDB can in turn represent several strings.

Delete(x,NDB)

```
1) Let Dx be all the strings in NDB that match x.
2) Remove Dx from NDB.
3) for all y in Dx,
4)    for each unspecified position qi of y,
5)       Create a new string y' using the specified bits of y and the complement of the bit specified at the ith position of x.
6)       Insert(y', NDB)
```

The algorithm takes the current NDB and the string to be removed x as input, line 1 identifies the subset, Dx, of NDB that matches x and removes it. Removing an entry that matches x might also unintentionally delete some additional strings.

Lines 3-6 reinsert all the strings represented by Dx except x. For each string y in Dx that has n unspecified positions (don't care symbols) there are n strings to be inserted into NDB that match everything y matches except x. Each new string y' is created by using the specified bits of y and the complement of the bit specified at the ith position of x as the following example illustrates:
Let $x = 101001$
And $Dx = 1 * 1 * 0 *$
Then all but x will be 111*0*, 1*110*, 1*1*00
Each new string $y^{'}$ by construction, differs from x in its ith position therefore none of the new strings match x.
Example: Deletion of 101 from NDB is shown below.

DB	NDB	DB+101	NDB-101
000	01*	000	01*
111	10*	101	100
	001	111	001
	110		110

C. Morph Operation

The morph operation [1] takes as input a NDB and outputs a NDB' that matches exactly the same set of binary strings; however, NDB and NDB' are different, that means some NDB records are not in NDB' and vice-versa. This illustrates the ability of the negative database scheme to have different representations for the same data. A property that makes it difficult to determine if two negative databases are equivalent (match exactly the same set of binary strings) without reversing them. Uses of the Morph operation include removing superfluous entries from the negative database and obscuring the history of operations performed on it.

D. Application Of Operators

Imagine a goods company that wishes to keep a list of risky credit-card numbers: credit-cards that have been involved in suspicious activities. A hard-to-reverse negative database can be created such that the ability to retrieve individual numbers is hindered, preventing a malicious party from stealing any numbers, and the capability to validate specific numbers is preserved.Suppose that the department that holds the negative database discovers that some number in the list is not delinquent. The Insert operation allows that party to remove the entry without finding out anything about any other record in the database. Likewise, if a new number comes along that is regarded as suspicious, it can be safely included in the watch list via the Delete operation.

VI. Querying NDB

Queries are also expressed as strings over the same alphabet; when a string, Q, consists entirely of defined positionsonly zeros and onesit is interpreted as "Is Q in DB?", and we refer to it as a simple membership or authentication query. Answering such a query requires examining NDB for a match and can be done in time proportional to $\|NDB\|$.Take, for example, a negative database of the tuples {name, address, profession}. The query "Is { Tintan, 69 Pine Street, Plumber } in DB?" (written as a binary string Q) would be easily answered, while retrieving the names and addresses of all the engineers in DB (expressed as a query string with the profession field set to the binary encoding of engineer and the remaining positions to *) would be intractable,[3]. Note that it is possible to construct NDBs, with specific structures, for which complex queries can be answered efficiently [4], [5]. Indeed, creating negative databases that are hard to reverse in practice can be a difficult task;the next chapter addresses this issue and present an algorithm for creating negative databases that only support authentication queries efficiently.

VII. Hard-to-Reverse Negative Databases

The creation of negative databases has been previously addressed in [4], [2], [5], where several algorithms are given that either produce NDBs that are provably easy to reverse, i.e., for which there is an efficient method to recover DB, or that have the flexibility to produce hard-to-reverse instances in theory,but have yet to produce them experimentally. It was shown in [4] that reversing a NDB is an NP-hard problem, but this, being a worst case property, presents the challenge of creating hard instances in practice.

In this section, we focus on a generation algorithm that aims at creating hard-to-reverse negative databases in practice [3]; we take advantage of the relationship negative databases have with the boolean satisfiability problem (SAT).The resulting scheme has two important differences with the algorithms of Refs. [4], [2], [5]besides the ability to produce hard instances: first, it generates an NDB for each string in DB, and second, it creates an inexact representation of U-DB, meaning that some strings in addition to DB will not be matched by NDB. In what follows we present the generation algorithm, outline how the problem of extra strings can be dealt with.

A. Using SAT Formulas as a Model for Negative Databases

Reference [6] presents an algorithm for creating SAT formulas which we use as the basis for our negative database construction. Their objective is to createa formula that is known to be satisfiable, but which SAT-solvers are unable to settle. The approach is to take an assignment A (a binary string representing the truth values for the variables in the formula), and create a formula satisfied by itmuch like the algorithms in [4], [2], [5],except that the resulting formula might be satisfied by other unknown assignments. Given the assignment A, the algorithm randomly generates clauses with $t \succ 0$ literals satisfied by it withprobability proportional to qt for $q \prec 1$ (q is an algorithm specific parameter used to bias the distribution of clauses within the formula). The purpose of the method is to balance the distribution of literals in such a way as to make formulas indistinguishable from one another in this respect. The process outputs a collection of clauses, all satisfied by A, which can be readily transformed into a negative database.Refer [3].

Given a database (DB) of size at most one, containing a l-length binary string A, we create a negative database (NDB) with the following properties:

1) Each entry in the negative database has exactly three specified bits.
2) A is not matched by any of NDBs entries.
3) Given an arbitrary l-bit string, it is easy to verify if it belongs to NDB or not (in time proportional to the size of NDB).
4) The size of NDB is linear in terms of the length of A. Let l be the number of bits in A and m the number of strings in NDB; the tunable parameter $r = m \div l$ determines the size of the database and its reversal difficulty.
5) The size of NDB does not depend on the contents of DB, i.e., it has the same size for $\|DB\| = 1$ and $\|DB\| = 0$.
6) A is ”almost” the ”only” string not matched by NDB, i.e., almost the only string contained in the positive image DB’ of NDB. The other entries in DB’ are close in hamming distance to A.
7) The negative database NDB is very hard to reverse, meaning no known search method can discover A in a reasonable amount of time.

Properties one through five follow from the isomorphism of negative databases with a 3-SAT formulae and the characteristics of the algorithm.

Boolean Formula	NDB
$(x1orx2orx\bar{5})$ and	00**1
$(x\bar{2}orx3orx5)$ and	*10*0
$(x2orx\bar{4}orx\bar{5})$ and	*0*11
$(x\bar{1}orx\bar{3}orx4)$	1*10*

Mapping SAT to NDB: In this example the boolean formula is written in conjunctive normal form (CNF) and is defined over five variables x1, x2, x3, x4, x5. The formula is mapped to a NDB where each clause corresponds to a record, and each variable in the clause is represented as a 1 if it appears negated, as a 0 if it appears un-negated, and as a * if it does not appear in the clause at all. It is easy to see that a satisfying assignment of the formula such as x1= FALSE, x2= TRUE, x3= TRUE, x4= FALSE, x5= FALSE corresponding to string 01100 is not represented in NDB and is therefore a member of DB.

1) Superfluous Strings: A consequence of the above method for generating negative databases, is the inclusion of extra strings in the corresponding positive database. That is, DB’the reverse of NDBwill include strings that are not in the original DB from which it was created; we refer to these strings as superfluous.$(DB \subset DB')$ To address the incidence of superfluous strings, we introduce a scheme that allows us to distinguish, with high probability, the true members of DB from the artifacts. Rather than creating a NDB using A as input, we construct a surrogate string A’appending to A the output of some function F of Aand use it to generate NDB. The membership of an arbitrary string B is established by computing F(B) and testing whether B concatenated with F(B) is represented in NDB. The purpose of the function is to divide the possible DB’ entries into valid and invalidvalid strings having the correct output of F appended to themand reduce the probability of including any unwanted valid strings in DB’.

The choice of function impacts both the accuracy of recovery (avoidance of superfluous strings) and the performance of the database: the more bits appended to A, the less likely to mistake a false string for a true one (assuming areasonable code) and the larger the resulting NDB. There is a wide variety of codes that can be used for this purpose: parity bits, checksums, CRC codes, and even hash functions like SHA or MD5 with upwards of a 100 bits.

B. Multi-record Negative Databases

The preceding section explored how to create a hard-to-reverse negative representation of a DB with zero or one entries; now, we briefly outline how this can be extended for DBs of an arbitrary sizethe work in [4], [2]is concerned with creating negative databases for any DB, regardless of its size, but does not show that the instances they output are hard to reverse in practice.

Our scheme can be used to generate the negative representation of any set of strings DB by creating an individual NDBAi for each string Ai in DB, i.e., each record in the resulting NDB is itself some negative database. It is important to point out that all NDBAis are the same size (and are thus indistinguishable by this measure) and that some may represent the empty (positive) set.

DB	NDB0	NDB4	NDB5	$NDB\oslash$
000	*1*	*1*	0**	0**
100	1**	**1	**0	*1*
101	0*1	000	*11	10*

A sample DB with possible NDBAi ($NDB\oslash$ represents the empty set). The final NDB collects all NDBAis.

Compare this scheme to the method described in [4], [2].First, there is additional information leakage, as the size of the underlying DB can be bounded by the number of records (NDBAis) in NDBa bound, since NDB may contain any number of records that represent the empty set. Second, a NDB created in this manner is much easier to update: inserting a string Ai into DB is implemented as finding which records in NDB represent Ai and removing them; deleting Ai from DB amounts to generating its corresponding NDBAi and appending it as a record to NDB. The result is a database in which updates take linear time (or better as discussed below) and whose size remains linear in $\|DB\|$. Moreover, our scheme allows many operations to be parallelized, given that the database can be safely divided into subsets of records and the results easily integrated. This contrasts with the databases and update operations presented in [2], where a single ”insert into DB” requires access to all of NDB, runs in $O(l^4\|NDB\|^2)$ time, and may cause the database to grow exponentially when repeatedly applied. Finally, the nature of updates remain ambiguous to an observer, given that a record can represent the empty set and that different records (different NDBAis) can stand in for the same DB entry. Distributed Negative Databases Consider a negative database NDB whose records are distributed among n subsets NDB1 . . . NDBn. Each NDBi matches only a fraction of U–DB and is the negative image of some set $GDBi = DB$ [Gi, where Gi are those strings in U–DB not matched by NDBi, and act as noise that obfuscates the composition of DB]. The union of the NDBi yields the negative image of DB; the intersection of the GDBis produces DB.

Imagine a scenario in which each NDBi is assigned to an agent Ai and that the agents are independent of each other, able to consult only the database they manage. Some other entity J wishes to ascertain whether item x is in DB and is allowed to pose a query to each Ai. By the set intersection property [1], J can determine the membership of x by consulting the individual agents. Note however, that if it is the case that x is a member of DB all agents need be queried to confirm it as a fact. This brings out an important characteristic of the setup: Only J can establish with certainty that x is in DB and J has to consult all Ai's to do so.

Suppose a list of names has been drawn as a result of a lottery. The agency that holds the event wishes to communicate to the winners their good fortune. However, they want to avoid publishing the clear text of the list to avoid impersonators, thieves, and to protect the privacy of the lucky ones. After reading this dissertation the board of directors has decided to create a negative database of the list, and distribute it among n parties. Each party may provide answers regarding their share but must disallow any single entity from making several consultations (allowing too many queries opens the door for a dictionary attack). Also, they must protect their NDBi from getting stolen. Any person interested in learning whether they've won will need to consult the n agents, if their name is not in any of the negative databases, they may go to the benevolent agency and ask for their winnings. A variant of this might ask the holders of the partial databases to issue a certificate: "Name Y is not on NDBi", the possession of the n certificates with your name, together with an Id, allows you to claim the prize. The benefit of the setup is that only the winners (and the agency) can be absolutely sure that they've won. The administrator of each NDBi knows the name of some of the people that did not win, but cannot be certain of the name of anyone who actually did. The security of this scheme relies on the fact that the negative database is distributed, each agent having a limited amount of information, rather than on having a negative database that is hard to reverse. Also, the distributed strategy allows for more complicated queries than simple membership; say the positive database DB has the tuples {name, profession, address} and the negative database NDB all possible tuples not in DB. If NDB is divided into n subsets, a query such as "What are the names and addresses of the engineers contained in DB" can be answered by issuing "Give me all the tuples {name, address} of the records having engineer in the profession field" to each NDBi and compiling a new NDB0 out of all the replies. Reversing NDB0 will yield the desired answer. Note that NDB0 has all the tuples {name, address} of all the engineers that are not in DB, complementing it (assuming NDB0 is easy to reverse) will leave those that are.

This setup can be used in concert with other encryption techniques to yield some interesting applications, for instance: Some agency wants to communicate a message to a group of people. They want the message to be secret, the identity of the recipients to remain ambiguous, and to conceal which messages have been collected.

One option is to encrypt the message using the name of the recipient as key, producing E(message), and create a database DB with the tuples {H(name), E(message)} where H(name) is the hash of the name with a public one–way function. Now instead of publishing DB, which is susceptible to a dictionary attack and lets the entity in charge of managing it know which messages have been retrieved, we can use the construct described above and create a negative database NDB and divide it amongst n parties. To see if you have a message simply hash your name to create H(name) and ask each NDBi for all records with H(name) in their name field (concealing your identity). Inverting the union of the tuples {E(message)} reported by all NDBis will result in a database with your messages. Only you know which records it contains, only you know if it is empty or not. If it's not, the messages may be decrypted using your name.

A further characteristic of distributing a negative database worth pointing out is that inserting an item x into the definition of DB, i.e. deleting it from NDB, is a more difficult task than removing it, since all NDBis have to be inspected and perhaps modified to assure x is subtracted from all of them (recall that there is a lot of redundancy in a NDB and x may be present in several NDBis). Removing x from DB, on the other hand, is simple as it will suffice to append it to any NDBi. Contrast this to performing the same operations on a distributed positive database; inserting x into DB is accomplished by inserting it into any subset, while removing it will entail making sure it is absent from all of its subsets.

Resilience to deletions is a desirable property for data integrity. However, there are settings where it is useful to easily remove an item, such as when the database stores a checklist or when it keeps perishable messages. Resilience to insertion is advantageous in scenarios like the one described herein where you wish to prevent people from including their names in the list of winners. Similar strategies can be devised to perform secret sharing that exhibits some of the nuances of using negative databases.

VIII. DB VS NDB

A finite universe of items can be partitioned into two sets the positive DB or just DB, and the U-DB whose compressed form is the NDB. The DB is deemed to contain the items of interest, and in general, is considered to be smaller than its image. The characteristics examined pertain to the fact that the amount of information per string is lower in a negative set and increases as the identity of the item that it represents is revealed. Inferring answers from either set differs in that to establish that x belongs to DB, the entire of NDB must be inspected, while the search may stop as soon as x is encountered in a the search may stop as soon as x is encountered in the DB. The converse is true when x is a member of NDB. The most important advantage of using NDBs that these provide very little information about the DB. Also, it is NP-Hard to reverse NDB to get DB.

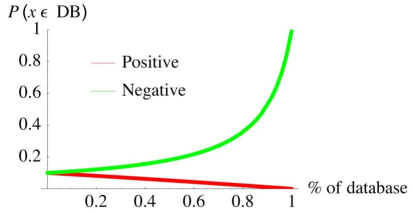


Fig. 1: DB contains 10% of all possible L-length strings (formulas)

IX. Conclusion

The seminar report has focused on two questions regarding such a convoluted way of describing sets, namely: Can it be done efficiently, and what can be gained from it? Theoretically, as long as the universe is finite, it is always possible to list all the objects not in some arbitrary subset but, as the crow example suggests, the list might be too big and it might take too long a time to generate. Therefore, can it be done efficiently? The second concern is more open-ended and at first glance, hard to imagine.

The chief contribution of the report has been to draw attention to an alternate way to represent data, one that is often alluded to in our everyday discourse but hasn't been studied in a rigorous fashion. The study has been limited to specific kinds of sets (consisting of finite length strings) and showed that, even here, there are some distinctive characteristics in representing information negatively and that it is actually feasible to create and store them.

Research is going on to develop the existing representations and make them more flexible. Negative data mining will be possible in a few years. Also, it is being researched whether it would be feasible to expand the universe beyond the finite set of strings. There are many venues for future work beyond refining the current analysis.

It would be of interest to study alternative encodings of negative information and expand their scope beyond finite sets of strings. Finally, can these representations be used to shed light on the working of intricate biological and immune systems? Can the immune system be thought of as a negative database that is hard to reverse, as a security system that co-evolves with the pathogens that try to subvert it?

References

[1] F.Esponda,*"Everything that is not important: negative databases"*,IEEE Computational Intelligence may 2008.

[2] F.Esponda,E.S.Ackley,S.Forrest,P.Helman,*"Online negative databases"*,International Journal of Unconventional Computing,vol 1 no:3,pp 403-416,2005.

[3] F.Esponda, E.S.Ackley, S.Forrest, H.Jia, P.Helman,*"Protecting Data privacy through Hard to reverse negative databases"*, International Journal of Information and Security, vol 6, pp 403-416, Oct 07.

[4] F.Esponda, Stephanie Forrest and Paul Helman,*" Enhancing Privacy through Negative Representations of Data"*, UNM Computer Science Technical Report TR-CS-2004-18, March 2004.

[5] F.Esponda, S. Forrest, and P. Helman,*"Negative representations of information"*, Submitted to International Journal of Information Security, 2004.

[6] H. Jia, C. Moore, and D. Strain,*"Generating hard satisfiable formulas by hiding solutions deceptively"*, In AAAI, 2005.

[7] F. Esponda, E.D. Trias, E.S. Ackley, and S. Forrest,*"A Relational Algebra for Negative Databases"*,UNM Computer Science Technical Report TR-CS-2007-18, November 2007.

[8] http://www.cs.unm.edu/ forrest/projects/ndb/index.html

[9] http://www.ieeexplore.org/

Notes on O(n) Median Finding Algorithm

Manu Madhavan,
Dept. of Computer Science and Engg,
SIMAT, Vavanoor, Palakkad
manumadhavan@simat.ac.in

***Abstract*—This paper gives an overview of design of a divide and conquer algorithm to find the kth smallest element of an unsorted array. Traditional algorithms, find this by sorting the array. But, the following algorithm use a linear time procedure to solve the problem. The base idea of this algorithm is taken from Quick sort logic- i.e dividing array based on pivot. A pseudo median of the array elements are selected as pivot. The construction and analysis of the algorithm is also explained.**

***Keywords*—Pseudo median, Pivot, Divide and Conquer, Linear Algorithm.**

I. INTRODUCTION

A selection algorithm is an algorithm for finding the k^{th} smallest number in a list or array; such a number is called the k^{th} order statistic. This includes the cases of finding the minimum, maximum, and median elements. There are O(n) (worst-case linear time) selection algorithms, and sub-linear performance is possible for structured data; in the extreme, O(1) for an array of sorted data. Selection is a subproblem of more complex problems like the nearest neighbor problem and shortest path problems. Many selection algorithms are derived by generalizing a sorting algorithm, and conversely some sorting algorithms can be derived as repeated application of selection.

The median of an array of size n (with integers) can be find by sorting the array in ascending order and returning $\lceil(n/2)^th$ element. We could simply sort the entire array A the $(n/2)^{th}$ element of the resulting array would be our answer. If we use MERGE-SORT to sort A in place and we assume that jumping to the i^{th} element of an array takes O(1) time, then the running time of our algorithm is $\theta(n\ lg\ n)$ for sorting and $\theta(1)$ for returning the median. So, this is not linear.

We are thinking about a linear algorithm, to find the median element from an unsorted list. Here, we can use divide and conquer algorithm design to get the desired algorithm.

To do this, we divide the array A into two sub-arrays A_L and A_R based on an approximate median x (like pivot in quick sort). The x is select such a way that, both $|A_L|$ and $|A_R|$ are at least $n/4$. Now, if $|A_L| > n/2$ the median will be in right part, otherwise it will be in left part. We do the above search recursively until reach the solution.

Since the above method have only partitioning, which can be done in $O(n)$ time. But the unknown thing is how to find approximate median in linear time. In fact, a specialized median-selection algorithm can be used to build a general selection algorithm, as in median of medians. The best-known selection algorithm is Select, which is related to quicksort; like quicksort, it has (asymptotically) optimal average performance, but poor worst-case performance, though it can be modified to give optimal worst-case performance as well.

II. PROBLEM

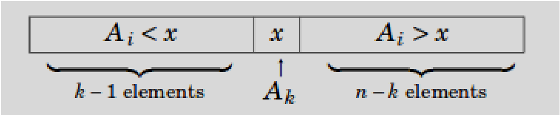


Fig. 1: Dividing the Array into left and right sub arrays.

Consider the above figure. If $|A_L| < n/2$ the median will be in $|A_R|$. The index of the median in $|A_R|$ will be $n/2 - |A_L| - 1$. So, the index of r^{th} ranked document in such an array will be $r - |A_L| - 1$. Based on the above observation, we can generalize the median finding problem as follows:

Given an array $A[1, ..., n]$ of numbers and an index r $(1 \leq r \leq n$ find the r^{th} smallest element of A.

The median fining will be a special case, where $r = \lceil(n/2)$.

III. SOLUTION

The solution as discussed in Introduction is based on Quick sort algorithm. An approximate median of the array elements, called pseudo median is considered as pivot element. To find the pseudo median, the array is first grouped into (n/5) groups, each with 5 elements. Then we will construct a new array with the (exact)median of each group. Then B have (n/5) elements. Then this procedure repeated on B to find its median. So, this recursive call will returns the pivot, which is the median of median. Then array is divided into left and right sub arrays, based on the pivot.

IV. ALGORITHM

Now we can design a divide and conquer algorithm to find the r^{th} element from an unsorted list using approximate median.

Algorithm 1: Select(A,r)

Input : An Sorted list A of size n and an integer r such that $1 \leq r \leq n$.
Output: r^{th} smallest element in A
Steps:

1) Partition A into (n/5) groups, with 5 elements each

```
    2) Find median (exact median) of each group and put it into array B. So Size of B will be (n/5).
    3) x=Select(B, n/10) (i.e find median of B, since $|B| = n/5$, its median will be $(n/10)^{th}$ smallest element. )
    4) Partition A into $A_L$ and $A_R$ based on x, such that elements less than x will be in $A_L$ and elements grater than x will be in $A_R$.
    5) If $|A_L| = r - 1$, return x.
    6) If $|A_L| > r - 1$, Select($A_L$,r-1)
    7) Else, if $|A_L| < r - 1$, Select($A_R$, $r - |A_L| - 1$)
end
```

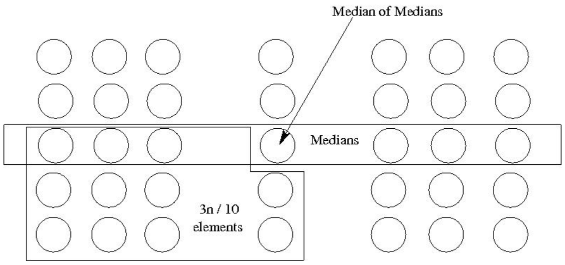


Fig. 2: Concept of Median of Median.

V. Analysis

Now we have to prove that the above algorithm take linear time to calculate the r^{th} smallest element. Let $T(n)$ be the total time taken to find this element on array $A[1...n]$.

- *Step 1: Divide array into (n/5 groups, each of size 5*
  Clearly, this can be done in $O(n)$ steps.
- *Step 2: Find median of each group*
  To find exact median of a group, we have to sort the elements and return the middle element. In worst case, this will take $O(n^2)$. Here, each group have size 5. So, this will take 25 steps in all time, which is a constant. Then, there will be (n/5) such groups. So, the total time will be $25 \times (n/5) = 5n$. So it is also linear.
- *Step 3: Find median of median, recursive call on array B*
  Since B contain median of each group, its size will be (n/5). So the time taken to find median of B, by recursive call is T(n/5).
- *Step 4; Partitioning A into A_L and A_R*
  This involves comparison of $n - 1$ elements with x. So, the time complexity will be $O(n)$.
- *Step 5: If $|A_L| = r - 1$ return x.*
  In this step, the algorithm finds the solution. This will take O(1) time.
- *Step 6 and 7: Recursive call on Select with A_L or A_R*
  As we stated earlier, size of both A_L and A_R should be at least n/4. Sinxe we are dividing A into group of 5 elements, and partitioning A is based on the median of these group, this condition will be maintained. Similarly, the number of elements will be at most 3n/4.
  ⇒ $|A_L| \geq n/4 \; and |A_R| \geq n/4$
  Similarly, $|A_L| \leq 3n/4 \; and |A_R| \leq 3n/4$
  Therefore the worst case complexity of these calls will be T(3n/4) at most.

So, the total complexity of Select algorithm will be
$T(n) \leq Cn + T(n/5) + n + T(3n/4)$
$T(n) \leq (C + 1)n + T(n/5) + T(3n/4)$
Now we have to prove that this can be equal to linear time. T(n) can be solved by substitution. Assume $T(n) \leq cn. \implies (cn + 1)n + (cn/5) + (3cn/4) \leq cn$
$(c + 1) + c/5 + 3c/4 \leq c$
⇒ There exists a constant c such that, $T(n) \leq c(n)$
$\implies T(n) = O(n)$

References

[1] "Lecture notes for January 25, 1996: Selection and order statistics", ICS 161: Design and Analysis of Algorithms, David Eppstein

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001.

[3] Design and Analysis of Algorithm, Video Lecture by Prof. Prof.Sunder Vishwanathan, NPTEL

Effective Data Mining Using Neural Network Towards Business Forecasting

N. M. Midhun, K. Najmathunnisa, T. M. Swathy, Vinitha Mathew, Vishnu Venugopal,
Dept. of Computer Science and Engg,
SIMAT, Vavanoor, Palakkad
vishnuvenugopal15@gmail.com

***Abstract*—Companies have been collecting data for decades, building massive data warehouse to store it. Even though this data is available, very few companies has been able to realize the actual value stored in it. The question these companies are asking is how to extract this value. The answer is data mining. There are many technologies available for data mining like Artificial Neural networks, regression and decision trees. Many companies are vary of neural networks due to their black box nature, Even though they have proven themselves in many situations. The neural networks exhibit mapping capabilities, they can map input pattern to their associated output pattern. Neural network architectures can be trained with known examples of a problem before they are tested for their influence capacity on unknown instance of the problem. In this paper it is also proposed to have study on business forecasting based on neural networks. It focus on use of business intelligence (BI) in neural. This paper propose an implementation of neural networks, for effective data mining with special care for BI. BI on the cloud technology make it affordable and easily available as compared to traditional BI.**

Keywords*—Dataminig, Neural Networks(NN), Artificial Neural Networks(ANN), Business forecasting strategy, Influencing factors, Marketing neural forecasting, Financial neural forecasting, Operational neural forecasting, Risk assessment neural forecasting

I. Introduction

DATAMINIG is the term used to describe the process of extracting value from the database.An enterprise data warehouse(EDW),is the system used for reporting and data analysis.Integrating data from one or more disparate source creates a repository of data,a Datawarhouse(DW).Data warehouse store current and historical data and are used for creating trending reports.Mainly it required four things to perform effective datamining:good-quality,accurate data,an adequate sample size and perfect tool.There are many dataminig tool availableto practitioner such as decision tree,various types of regression and neural network[1].

A. Artificial Neural Network

An artificial neural network(ANN),often just called a ”neural network”(NN) is a mathematical or computational models. The term neural network traditionally refers to a network of biological neurons. The modern usage of the term refers to artificial neural networks, which are composed of artificial neurons. Thus the term has two distinct usages, Biological neural networks and Artificial Neural Networks. Biological neural networks are made up of real biological neurons that are connected in the peripheral nervous system or central nervous system. In the field of neuroscience, they are identified as a group of neurons that performs a specific physiological function in the laboratory analysis. Artificial neural networks are composed of interconnecting artificial neurons. This network may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems[1].

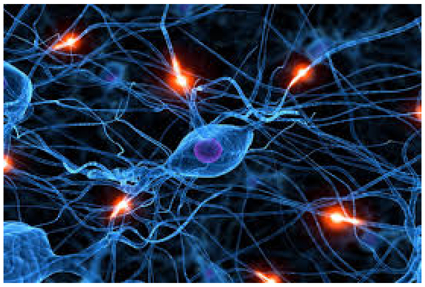


Fig. 1: Biological Neural Networks

The neural networks exhibit mapping capability, they can map input pattern to their associated output patterns. They can learn by examples. Neural network architectures can be trained with known examples of a problem before they are tested for their inference capacity on unknown instances of the problem. They can identify new objects previously untrained. They possess the capacity to generalize. Thus, they can predict new outcomes from past trends. They are robust systems and fault tolerant. They can recall full patterns from incomplete, partial or noisy patterns. They can also process information in parallel, at high speed and in a distributed manner. In most case an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase[1].

B. Neural Network Topologies :

1) Feedforward neural network: The feedforward neural network was the first and arguably simplest type of artificial neural network devised. Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down

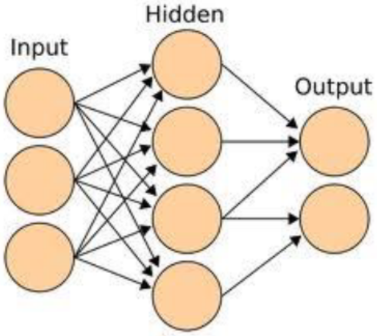


Fig. 2: Neural Networks

2) Feedback networks/Recurrent networks: Recurrent neural networks that do contain feedback connection.Feedback networks are bi-directional data flow.While a feedforward network propagates data linearly from input to output,RNs also propagate data from later processing stages to earlier stages. Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. These networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.

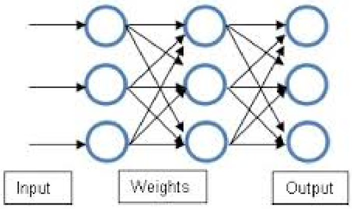


Fig. 3: Feed-Forward Networks

C. Training of Artificial Neural Network

A neural network has to be configured such that the application of the a set of inputs produces the desired set of outputs.Various methods were there for strengths of the connection exits.One method is set the weights explicitly to each connection.And other method is'train the neural network' by feeding it teaching patterns and letting it change its weights according to some learning rule[1].We can categorize it as

1) Supervised learning: Supervised learning or Associative learning in which the network is trained by providing it with input and matching output patterns. These input-output pairs can be provided by an external teacher, or by the system which contains the neural network (self-supervised).

2) Unsupervised learning: Unsupervisedlearning or Self-organization in which an (output) unit is trained to respond to clusters of pattern within the input. In this paradigm the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli.

3) Reinforcement Learning: This type of learning may be considered as an intermediate form of the above two types of learning. Here the learning machine does some action on the environment and gets a feedback response from the environment. The learning system grades its action good (rewarding) or bad (punishable) based on the environmental response and accordingly adjusts its parameters.

D. Types of Algorithms

1) Perceptron: The perceptron can be trained by adjusting the weights of the inputs with Supervised Learning. In this learning technique, the patterns to be recognised are known in advance, and a training set of input values are already classified with the desired output. Before commencing, the weights are initialised with random values. Each training set is then presented for the perceptron in turn. For every input set the output from the perceptron is compared to the desired output. If the output is correct, no weights are altered. However, if the output is wrong, we have to distinguish which of the patterns we would like the result to be, and adjust the weights on the currently active inputs towards the desired result[2].

Perceptron Convergence Theorem
The perceptron algorithm finds a linear discriminant function in finite iterations if the training set is linearly separable.
The learning algorithm for the perceptron can be improved in several ways to improve efficiency, but the algorithm lacks usefulness as long as it is only possible to classify linear separable patterns.

2) The multilayer perceptron (MLP) or Multilayer feedforward network: Building on the algorithm of the simple Perceptron, the MLP model not only gives a perceptron structure for representing more than two classes, it also defines a learning rule for this kind of network[1]. The MLP is divided into three layers: the input layer, the hidden layer and the output layer, where each layer in this order gives the input to the next. The extra layers gives the structure needed to recognise non-linearly separable classes[5].

Algorithm
The threshold function of the units is modified to be a function that is continuous derivative, the sigmoid function(The Sigmoid Function). The use of the Sigmoid function gives the extra information necessary for the network to implement the

back-propagation training algorithm. Back-propagation works by finding the squared error (the Error function) of the entire network, and then calculating the error term for each of the output and hidden units by using the output from the previous neuron layer. The weights of the entire network are then adjusted with dependence on the error term and the given learning rate. (MLP Adapt weights) Training continues on the training set until the error function reaches a certain minimum. If the minimum is set too high, the network might not be able to correctly classify a pattern. But if the minimum is set too low, the network will have difficulties in classifying noisy patterns.

II. Neural Network in Datamining

Neural networks essentially comprise three pieces: the architecture or model; the learning algorithm; and the activation functions.Neural networks are programmed or trained to store, recognize, and associatively retrieve patterns or database entries; to solve combinatorial optimization problems; to filter noise from measurement data; to control ill-defined problems; in summary, to estimate sampled functions when we do not know the form of the functions. It is precisely these two abilities (pattern recognition and function estimation) which make artificial neural networks (ANN) so prevalent a utility in data mining. As data sets grow to massive sizes, the need for automated processing becomes clear. With their model-free estimators and their dual nature, neural networks serve data mining in a myriad of ways[2].

Data mining is the business of answering questions that youve not asked yet. Data mining reaches deep into databases. Data mining tasks can be classified into two categories: Descriptive and predictive data mining. Descriptive data mining provides information to understand what is happening inside the data without a predetermined idea. Predictive data mining allows the user to submit records with unknown field values, and the system will guess the unknown values based on previous patterns discovered form the database. Data mining models can becategorized according to the tasks they perform: Classification and Prediction, Clustering, Association Rules. Classification and prediction is a predictive model, but clustering and association rules are descriptive models.

The most common action in data mining is classification[6]. It recognizes patterns that describe the group to which an item belongs. It does this by examining existing items that already have been classified and inferring a set of rules. Similar to classification is clustering. The major difference being that no groups have been predefined. Prediction is the construction and use of a model to assess the class of an unlabeled object or to assess the value or value ranges of a given object is likely to have. The next application is forecasting. This is different from predictions because it estimates the future value of continuous variables based on patterns within the data. Neural networks, depending on the architecture, provide associations, classifications, clusters, prediction and forecasting to the data mining industry[5].

A. The Back Propagation Algorithm

Backpropagation, or propagation of error, is a common method of teaching artificial neural networks how to perform a given task.The back propagation algorithm is used in layered feedforward ANNs. This means that the artificial neurons are organized in layers, and send their signals forward, and then the errors are propagated backwards. The back propagation algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated. The idea of the back propagation algorithm is to reduce this error, until the ANN learns the training data[1].

III. Business Intelligence (BI)

Human beings are continuously engaged in some activity or other in order to satisfy their unlimited wants. Every day we come across the word business or businessman directly or indirectly. Business has become essential part of modern world. Business is an economic activity, which is related with continuous and regular production and distribution of goods and services for satisfying human wants[3].

In an increasingly competitive world, business intelligence (BI) informs and guides your decision making to keep one's products and services competitive[3].

There are typically three phases in building a BI system:

- Consolidation
- Discovery
- Sharing

Phase 1: Consolidation

Step 1: Map transactional sources to a target data warehouse
Step 2: Generate the code to extract, transform and load data
Step 3: Generate the business area

Phase 2: Discovery

Now that the data is located in one place, delivers insight associated with the data captured about your products, customers, and marketplace, and allows you to quickly disseminate this information across the enterprise.

Phase 3: Sharing Data

Often much time and effort by the users are expended in the discovery phase where just the right query and analysis is performed. Once the right information is generated, sharing the information across the enterprise before it becomes stale can be challenging.

A. Business Forecasting

Business forecasting is a process used to estimate or predict future patterns using business data. Some examples of business forecasting include estimating quarterly sales, product demand, customer lifetime value and churn potential, inventory and supply-chain reorder timing, workforce attention, website traffic, and predicting exposure to fraud and risk.Several powerful estimation functions are commonly used to perform business forecasting: time series analysis, causal models, and regression analysis. Business forecasting supports executives, analysts and end users in decision-making using decision support systems such as business intelligence

1) Factors Influencing Business Forecasting: The factors which influence business forecasting are

- Preparation of Budget- Decision-making involves budget allocation i.e., resource allocation to various aspect of decision. Budget may be allocated to various factors of production[7].
- Future Development- Strategic plans are usually expected to have a significant future prosperity of the organization. This is because there is a long-term commitment. In case of absence of long-term commitment the firm cannot achieve future development.
- Orientation- Strategic planning should keep in view of the competition existing in the market. Sometimes firms have to face non-price competition.
- Factors of Environment- Plans are always influencing factor for decision-making. There are external or internal factors that influence business. Buyers, Suppliers, government and competitors are likely to react in accordance with changes in environment. Thus business also should act in the same passion.
- Risk- Strategic plans mostly face the problem of risk. The plans should able to tackle the risk bearing capacity. Risk and uncertainty are two important aspects, . which cannot be expected by business man[11].

IV. Cloud in BI

The use of Business Intelligence (BI) in the cloud is a game-changer, as it makes BI affordable and easily available as compared to traditional BI[6].
Cloud deployment strategies are often categorized either as Infrastructure-as-a-Service (IaaS), Platformas- a-Service (PaaS), or Software-as-a-Service (SaaS).

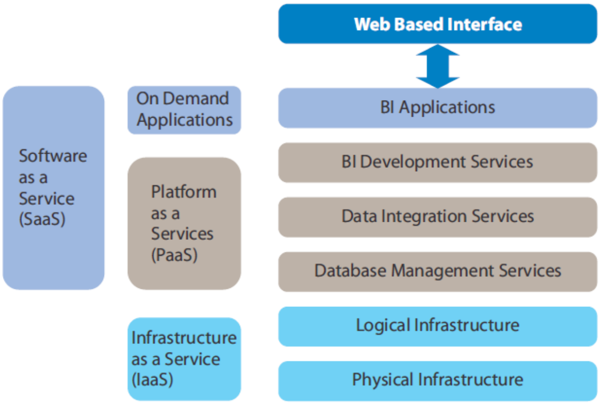

Web Based Interface
Software as a Service (SaaS)
On Demand Applications
BI Applications
Platform as a Services (PaaS)
BI Development Services
Data Integration Services
Database Management Services
Infrastructure as a Service (IaaS)
Logical Infrastructure
Physical Infrastructure

Fig. 4: Cloud BI Models

A. Business Intelligence on the Cloud: Drivers

There are several operational and financial factors that work in favor of Cloud Business Intelligence (BI), the key being[4]

- Speed of Implementation and Deployment: Immediate availability of environment without any dependence on the long periods associated with infrastructure procurement, application deployment, etc. drastically reduces the BI implementation time window.
- Elasticity: Leverage the massive computing power available on the Web, scale up and scale down based on changing requirements[9].
- Focus on Core Strength: Outsource running of BI apps to professionals and focus on their core capabilities.
- Lower Total Cost of Ownership: Convert some part of capital expenditure (capex) to operational expenditure (opex), cost-effective pricing models, pay per use model, etc.
- On-demand Availability: Support mobile and remote users, Browser-based access to control everything from the cloud platform to database management, from the data warehouse layer to the analytics platform.

Fig. 5: Business Cloud

B. BI On the Cloud: Concerns

There are also many inhibitors which have resulted in a very slow adoption rate to Cloud BI so far. A few common and leading concerns are mentioned here, along with recommendations on addressing the concerns:

- Data Security: Security concerns including confidentiality, integrity and availability of the data continues as the top concern for utilizing the Cloud. For some organizations, the concerns over security may be a barrier that is impossible to overcome today. However, as more organizations move to the Cloud it is expected that the concerns will lessen. In many cases, the Cloud vendors provide a more secure environment than what exists at customer sites.
- On premise Integration: Sudden movement to cloud is not feasible and a phased approach is usually recommended. There will be a co-existence model until the cloud BI market is more mature.
- Lack of control: Tough to get Service Level Agreements (SLAs) from cloud providers. Data control and data ownership, reliability of service challenges are some of

the main reasons for client concern. To mitigate this, organizations should already have in place thorough IT governance and service delivery standards and models.

- Vendor Maturity: Too many cloud BI vendors, hosting providers with varying offerings, etc. makes it confusing to choose the right vendor based on required needs and vendor capabilities.
- Performance: Limits to the size and performance of data warehouses in the Cloud, significant latency if BI applications exist in the Cloud but the data exists at a client site, especially when processing and returning large amounts of data.
- Pricing models: Lack of standardized pricing models makes it difficult for customers to select the right one

V. Conclusion

In this paper we present a neural network based approach to mining classification rules from given databases.There is rarely one right tool to use in data mining; it is a question as to what is available and what gives the best results. The use of neural networks in data mining is a promising field of research especially given the ready availability of large mass of data sets and the reported ability of neural networks to detect and assimilate relationships between a large numbers of variables. In most cases neural networks perform as well or better than the traditional statistical techniques to which they are compared. Due to design problems neural systems need further research before they are widely accepted in industry[9]. Cloud is a big part of future Business Intelligence and offers several advantages in terms of cost benefits, flexibility of implementation, availability and speed of implementation. Although initially cloud-based solutions were designed for small-to mid-size companies that did not have available IT resources or capital to spend on creating and managing a software and hardware infrastructure, today, many large companies are investigating the cloud as a way to add new business solutions quickly and augment existing data center capacity. While considering Cloud BI, organizations are recommended to follow a few risk mitigation steps and strategies:

- Perform due diligence for security, backup, and disaster recovery: Check whether the BI SaaS provider complies with emerging SaaS standards such as the SAS 70 Type II audit.
- Thoroughly understand the BI SaaS pricing and contract matters: Understand the various pricing models offering by the vendor and choose the one that is definitely needed, study the service-level agreements (SLAs) agreed upon by the vendor and keep track of actual application usage.
- Evaluate true long-term total costs of ownership: Perform a detailed RoI calculation to calculate longterm total ownership costs based on the specific environment and requirements.
- Double-check whether any additional source data licenses are needed: Understand if any additional licenses need to be procured for other enterprise applications that the SaaS BI application would need to interface with.
- Plan for the worst: Have a detailed cloud to on-premise migration strategy in place in case the Cloud Vendor fails to perform according to desired expectations.

References

[1] Hongjun Lu, Rudy Setiono and Huan Liu, *Effective Data Mining Using Neural Networks*, IEEE Transactions on Knowledge and Data Engineering, VOL. 8, NO. 6, December 1996.

[2] Prof. A. Maithili, Dr. R. Vasantha Kumari, Mr. S. Rajamanickam, *Neural Network towards Business Forecasting*, IOSR Journal of Engineering, Vol. 2(4), Apr. 2012.

[3] Stevan Mrdalj, *Would Cloud Computing Revolutionize Teaching Business Intelligence Courses?*, Issues in Informing Science and Information Technology, Volume 8, 2011.

[4] John J. Prevost, Kranthi, Manoj Nagothu, Brian Kelley and Mo Jamshidi, *Prediction of Cloud Data Center Networks Loads Using Stochastic and Neural Models*

[5] Amrendr Kumar, *Artifical Neural Networks for Data Mining*

[6] Dr. Yashpal Singh, Alok Singh Chauhan, *Neural Networks in Data Mining*, Journal of Theoretical and Applied Information Technology, 2005 - 2009 JATIT.

[7] S. W. Changchien, M. C. Lin,*Design and implementation of case based reasoning system for marketing plans, Expert Systems with Applications*,vol. 28, pp. 4353, 2005.

[8] Suefert Andhreas and Schiefer Josef.,*Enhanced Business Intelligence-Supporting Business Processes with Real-Time Business Analytics*,16th international workshop on Database and Expert System applications. Retrieved 19 June 2006 from www.ieee.org.

[9] Zeng, L., Xu, L., Shi, Z., Wang, M. and Wu, W.(2007), *Techniques, process, and enterprise solutions of business intelligence*,2006 . IEEE Conference on Systems, Man, and Cybernetics October 8-11, 2006, Taipei, Taiwan, Vol. 6, pp. 4722.

[10] Farhi Marir and Selma ,Liman Mansar, *An Adapted framework and Case-based Reasoning for Business Process*,An Adapted framework and Case-based Reasoning for Business Process.

[11] K. Kim, I. Han, *Maintaining case based reasoning systems using agenetic algorithms approach, Expert Systems with Applications*, vol.21, pp. 139-145, 2001.

Free Softwares and Free License

Gadheyan T S,
Dept. of Computer Science and Engg,
SIMAT, Vavanoor, Palakkad
gadheyansivakaran@gmail.com

***Abstract*—This paper discuss the various technical and philosophical aspects of free softwares. Free software is one that provides the users with basic freedoms to run, copy, distribute and study. These freedoms are important for developing innovations and creativity, especially in academic-research areas. This paper also touches the details of free licensing and open licensing.**

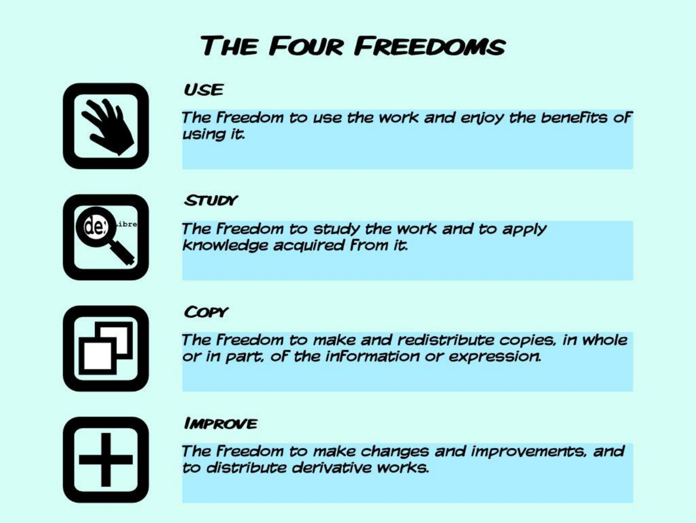
***Keywords*—Free Softwares, Free Licenses, Creative Common, Freewares, FOSS.**

I. What is Free software

A free software is any software that provides users with freedom to run, copy, distribute, study or change the software.There is a clear distinction between free software and gratis softwares which are generally called freewares.The term 'free' in free software emphasizes on user's freedom rather than price of the software.Freewares are proprietary softwares that are available free of cost. Inspite of the fact that freewares are gratis softwares,they process a danger.Since their source code is hidden,user cannot know what the software does on his/her computer.This means that the software controls the user rather than user controlling software.Free software is often ascribed as open source software.Although both philosophies have similar ideologies,open source philosophy does not talk about users freedom.Free software is a political movement while open source is a development model.The difference can be clearly seen in many softwares.Android is an open source project.Android is very different from the GNU/Linux operating system which is a free operating system,The only component in common between Android and GNU/Linux is its kernel,Linux.So Using the term free software is more appropriate if we are talking about freedom.

II. Why free softwares

Usage of free software is a political as well as an ethical choice.A person who opts this choice enjoys all freedom.Freedom to run,copy,modify,distribute or study the software.A fine print of the license agreement of a proprietary software contains the list of all freedoms that you have been denied.There are many other issues a society must face if it supports the use of proprietary software.Students who are interested in learning a software cannot do so because the source code is hidden away from them.Proprietary softwares raises big issues on privacy.Due to the fact that we are not able to see source code of the software,we cannot figure out what the software does on our computer.Because our computers control much of our personal information and our activities,proprietary softwares possess danger on our privacy.When government organization use proprietary softwares,whole society is under risk. History of software development started by sharing the software,modifying and redistributing it.In this case everyone could contribute to a software.This is not possible when source code is hidden.To really say that a software is 'available' to a user,it should be a free software.Every information of a software should be available to the user which only a free software can provide.



The Freedom to use the work and enjoy the benefits of using it.

The Freedom to study the work and to apply knowledge acquired from it.

The Freedom to make and redistribute copies, in whole or in part, of the information or expression.

The Freedom to make changes and improvements, and to distribute derivative works.

Fig. 1: The Four Freedom.

III. Free Software Movement

The Free Software Movement was kickstarted in 1985 by software freedom activist Richard Stallman.The idea of the Free Software Movement is that computer users deserve the freedom to form a community.In 1983 Richard Stallman announced the GNU project to create a free operating system.By 1990 the project was almost over but it missed an important part,the kernel.In 1991 Linus Torvalds released his kernel linux as free.This created the GNU/Linux operating system.GNU/Linux now powers almost all servers. Stallman in his catchy words justifies his support for free software.In his words-*"I could have made money this way, and perhaps amused myself writing code. But I knew that at the end of my career, I would look back on years of building walls to divide people, and feel I had spent my life making the world a worse place."*

According to stallman proprietary software is something that divides people.Free software is the only alternative for the issue.So he started the Free software movement and now it has captured the attention of many people around the world.Every activity in the free software community can be viewed in the website www.openhub.net.

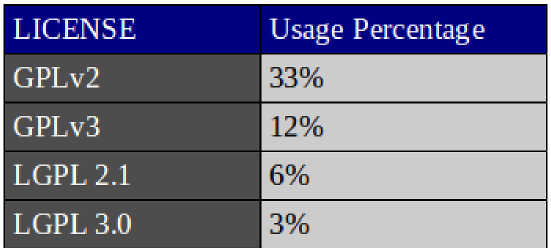
Fig. 2: Dr. Richar Stallman.

IV. Free Software Licenses

In the current scenario licensing is a marketing tool that is used by everyone from major corporations to the smallest of small business.They use the license to legally own and protect their products like softwares and their graphic design, trademark, logo, slogan, signature, name etc.Proprietary software licenses are used to restrict the use of a product by the consumer.Free software license differ from proprietary license in the sense that free software license provide freedom to the consumer.It grants the recipient of the software with freedom to copy,distribute,modify,redistribute and study the piece of software.The current copyright law restricts these actions,but the rights-holder can remove these restrictions by implicitly declaring it in the software license.Some free software licenses encompass a copyleft provision.This requires the future modified versions to provide all freedoms that the original software has provided.The most popular free license is the GNU general public license.It was first used in 1987 for the GNU Compiler collection.Now there are number of free licenses that are compatible with GNU GPL.

V. Various Free software License

All licenses that are accepted by Free Software Foundation are considered to be free software licenses.The most common and widely used free software license is the GNU General Public License.It guarantees the end user the freedom to copy,modify,distribute and study the software.The GPL demands the software to be copylefted.Following table describes the usage percentage of various versions of GPL license. Another popular license is the Apache License.It is a free software license written by Apache Software Foundation.This license allows the user to use the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software. Modified BSD license is another free software license that is generally accepted.The modified BSD removed the advertising clause from original BSD(Berkeley Software Distribution) license.FreeBSD is also a free software license that is modified by removing an advertising and another clause from original BSD. The CeCILL is a free software license, explicitly compatible with the GNU GPL.The CeCILL(CEA CNRS INRIA Logiciel Libre) was jointly developed by number of French agencies.There are many variants for this license.



LICENSE	Usage Percentage
GPLv2	33%
GPLv3	12%
LGPL 2.1	6%
LGPL 3.0	3%

Fig. 3: Free License usage.

VI. Other open licenses

Creative Commons license is a license that gives the user rights to use,share and modify a creative work by any author.It is widely used in many fields like licensing books, plays, movies, music, articles, photographs, blogs, and websites. Some other open licenses include Eclipse Public License,Mozilla Public License,Common Development and Distribution License,MIT License. Any of the above license can be used in your work by attaching the boilerplate text to your work thereby declaring it as a free or open source software.

A. Popular softwares and their licenses

Wordpress uses GPLv2 license.The popular operating system GNU/Linux uses GPL license.The 3d modeling software Blender uses GPL license.The image editing software GIMP also uses GPL license. Julia a high-level, high-performance dynamic programming language for technical computing,uses MIT license.Scilab a numerical computation software,uses CeCILL license.

References

[1] https://www.gnu.org/philosophy/free-sw.html

[2] http://fsfe.org/about/basics/freesoftware.en.html

[3] http://www.fsf.org/

[4] http://www.licensingexpo.com/licensing-expo/education/what-licensing.

[5] https://www.gnu.org/licenses/license-list.html

[6] http://www.fsf.org/licensing http://opensource.org/licenses

[7] http://opensource.org/licenses

Optical Character Recognition For Malayalam

E.P.Anjali, K.Rohini, S.Sarika, M.Uma, K.V.Vivek
Seventh Semester , 2011 Admission
Department of Computer Science and Engineering,
Sreepathy Institute of Management & Technology, Vavannoor, Palakkad, India-Pin 679533
E-mail:anjali.ep@gmail.com, rohu.kannalath@gmail.com, sarikaanugraha123@gmail.com
umadath94@gmail.com, vivurevathi@gmail.com

Abstract— **Optical Character Recognition is the process of automatic conversion of scanned images of printed or handwritten documents into computer processable codes. Most of the current OCR systems are designed for European languages using Roman script. We are presently developing an OCR system for Malayalam script, by combining the techniques of OCR and Language Synthesis. By using mobile camera or scanner, directly copy the Malayalam text from print media as an image. Then it is converted into a text document on the computer, so that it can be edited or searched. It combines the techniques of feature extraction and computer vision. It is implemented as a sequence of 3 stages - preprocessing, recognition and postprocessing.**

Keywords—**Recognition, Segmentation, Neural Networks.**

I. INTRODUCTION

Computers understand alphanumeric characters as ASCII code where each character or letter represents a recognizable code. However, computers cannot distinguish characters and words from scanned images of documents. Therefore, where information must be retrieved from scanned images such as government documents, tax returns and passport applications characters must first be converted to their ASCII equivalents before they can be recognized as readable text. Optical character recognition system (OCR) allows us to convert a document into electronic text, which we can edit and search etc. Thus OCR can be defined as the process of converting images of printed or handwritten material into a text document on the computer. OCR systems are already developed for European languages like English, German etc. It is difficult for the machine to recognise south-indian languages like Telugu, Kannada, Tamil and Malaylam because of their complex curves. Our aim is to develop a Character Recognition system for Malayalam that combines the techniques of OCR and Feature Extraction.

A. *Project Area*

- Optical Character Recognition: OCR has evolved since it was initially introduced into the computer industry. Fields like hand printing, printed text, and cursive handwriting recognition are the current focus of research in OCR technology. OCR systems scan the documents printed or handwritten on a paper as an image and recognize the characters in image to form a text document, which can be edited. The current active research areas in OCR include handwriting recognition, and also the printed versions of non-Roman scripts (especially those with a very large number of characters).
- Malayalam Language Synthesis: The modern Malayalam script consists of 13 vowel letters, 36 consonant letters, and a few other symbols. Consonants in Malayalam are known as Vyanjanaksharam and Vowels are known as Swarakshram. There are mainly 2 types of vowels: Independent vowels and dependent vowels. Another set of characters in the Malayalam script is the conjunct consonants. These characters are formed by the combination of more than one consonants and were widely used in old scripts.
- Image Processing: In imaging science, image processing is any form of signal processing for which the input is an image, such as a photographor video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. A detailed study of image processing is given by Rapheal and Richard[3]. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. The acquisition of images (producing the input image in the first place) is referred to as imaging. In our project, image processing is used for converting scanned image to binary image. Binary images are images whose pixels have only two possible intensity values. They are normally displayed as black and white. Numerically, the two values are often 0 for black, and either 1 or 255 for white.
- Neural Networks: ANN Training and Classification: Before the character recognition can take place, the ANN is trained, so that it can develop the capability of mapping various inputs to the required outputs and effectively classify various characters. For training the ANN, we use the Vectors generated by the Database Templates using Feature Extraction techniques. It may be noted that the ANN uses Back propagation algorithm for Learning. The Target values are specified by the system programmer to accommodate for small recognition errors, which may be changed from application to application. The ANN was trained for 1000 iterations, which took around 21 seconds

to complete. The Training Function was set to use Sum Squared Error rather than Mean Squared Error, because the system need to calculate the effect of joint errors in all the parameters, rather than overall error. An error goal of 0.0001 or 0.01

B. Scope and Application

- Make textual versions of printed documents quickly
- Make electronic images of printed documents searchable.
- Converts handwriting to control a computer.
- Automatic insurance documents key information extraction
- Assistive technology for blind and visually impaired users
- Automatic number plate recognition
- Extracting business card information into a contact list
- Defeating CAPTCHA anti-bot systems.

II. Literature Survey

A. Theoretical Background

OCR research and development can be traced back to the early 1950s, when scientists tried to capture the images of characters and texts.

1) First generation OCR system: The first generation machines are characterized by the constrained letter shapes which the OCRs can read. These symbols were specially designed for machine reading, and they did not even look natural. The first commercialized OCR of this generation was IBM 1418, which was designed to read a special IBM font 407. The recognition method was template matching, which compares the character image with a library of prototype images for each character of each font.

2) Second generation OCR system: Next generation machines were able to recognize regular machine-printed and handprinted characters. The character set was limited to numerals and a few letters and symbols. Such machines appeared in the middle of 1960s to early 1970s. The methods were based on the structural analysis approach.

3) Third generation OCR system: For the third generation of OCR systems, the challenges were documents of poor quality and large printed and hand-written character sets. Low cost and high performance were also important objectives. Commercial OCR systems with such capabilities appeared during the decade 1975 to 1985.

4) Fourth generation OCR system: The fourth generation can be characterized by the OCR of complex documents intermixing with text, graphics, tables and mathematical symbols, unconstrained handwritten characters, color documents, low-quality noisy documents, etc.

B. Components of OCR

The principal task of OCR is to develop computer algorithms to identify the characters in the text. All OCR implementations consist of a number of preprocessing steps followed by the actual recognition of text as shown in Figure 1. Each of the exising methods differ mainly in the classification techniques they are using.

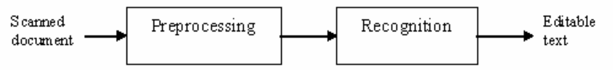


Fig. 1. OCR Process

Classification is one of the commonly used data mining technique, that uses a set of pre-classified examples to develop a model that can classify the population of records at large. The classification of data involves learning and classification. The training data are analyzed by classification algorithm in the learning phase. In classification test data are used to find the accuracy of the rules used. If the accuracy is acceptable the rules can be applied to the new data tuples[12]. Some of the classification techniques are:

- Bayesian Classification
- Classification by decision tree
- Neural Networks
- Support Vector Machines (SVM)
- Fuzzy Logic

1) Bayesian Classification: Bayesian classification is based on Bayes Theorem. Bayesian classifiers are the statistical classifiers. They are able to predict the class membership probabilities such as the probability that a given tuple belongs to a particular class. Bayes Theorem is named after Thomas Bayes. There are two types of probability as follows:

- Posterior Probability [P(H/X)]
- Prior Probability [P(H)]

Where, X is data tuple and H is some hypothesis. According to Bayes Theorem,

$$P(h/D) = \frac{P(D/h)\,P(h)}{P(D)}$$

Where, P(h) is Prior probability of hypothesis h. P(D) is Prior probability of training data D. P(h/D) is Probability of h given D. P(D/h) is Probability of D given h.

A Bayesian classifier is based on the idea that the role of a class is to predict the values of features for members of that class. The idea behind a Bayesian classifier is that, if an agent knows the class, it can predict the values of the other features. If it does not know the class, Bayes rule can be used to predict the class given the feature values. In a Bayesian classifier, the learning agent builds a probabilistic model of the features and uses that model to predict the classification of a new example.Bayesian approaches are a fundamentally important data mining technique. Given the probability distribution,Bayes classifier can provably achieve the optimal result.It is based on the probability theory.

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for Bayes models uses the method of maximum likelihood.

An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.One limitation that the Bayesian approaches can not cross is the need of the probability estimation from the training dataset. It is noticeable that in some situations, such as the decision is clearly based on certain criteria, or the dataset has high degree of randomality, then Bayesian approaches will not be a good choice.

2) Decision tree classifier: A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm. Given two algorithm to count the number of vertical and horizontal lines in a character this feed to the tree. Algorithm to count the number of vertical lines in a character is as follows:

- Initially, the height and width of the character is calculated.
- The character box is scanned horizontally from the top left position to find the left most black pixel.
- When a black pixel is found, count is incremented and the pixels which are present at the top, bottom and diagonal to it are also checked.
- If any of those pixels are black, the count is again incremented. This is done to check for slanted or curved vertical lines.
- Else if none of these pixels are black, check whether the count has reached the character height. If so a vertical line has been found and the vertical line count is incremented.
- Repeat the above steps by incrementing the x coordinate till the width of the character.
- The above process is repeated for each character in the text.

Algorithm to calculate the number and position of horizontal lines character is as follows:

- Initially, the height and width of the character is calculated.
- The character box is scanned vertically from the top left position to find the left most black pixel.
- When a black pixel is found, count is incremented and the pixels which are present at the left, right and diagonal to it are also checked.
- If any of those pixels are black, the count is again incremented. This is done to check for slanted or curved vertical lines.
- Else if none of these pixels are black, check whether the count has reached around the character width. If so a horizontal line has been found and the horizontal line count is incremented.
- Divide the character box into 3 vertical segments in order to calculate the position of the horizontal lines.

The figure 2 & figure 3 represents the decision trees formulated based on the vertical and horizontal line position analyzer algorithm. The following acronyms are used for the decision trees.

VL is Vertical Lines, HL is Horizontal Lines, M is Middle Line, T is Top Line, B is Bottom Line.

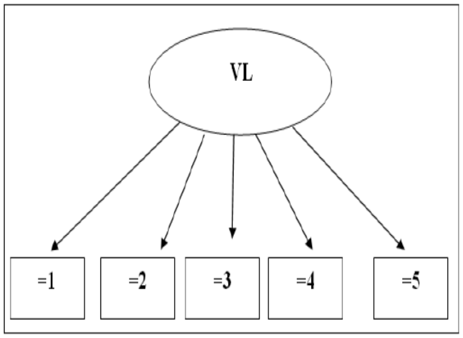


Fig. 2. Decision tree Classification based on number of vertical lines

3) Neural Networks: Artificial neural networks (ANNs) are computational models inspired by an animal's central nervous systems, and are used to estimate outputs that can depend on a large number of inputs which are generally unknown. Artificial neural networks are represented as systems of interconnected neurons which can compute values from inputs, and are capable of machine learning as well as pattern recognition. It has qualities such as adapting to changes and learning from prior experiences.

Figure 4 represents an artificial neural network interconnected by group of nodes. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another.

Since its more adaptive in nature we are using neural networks for character recognition here. It is defined by a set of input neurons which may be activated by the pixels of an input image. After being weighted and transformed by

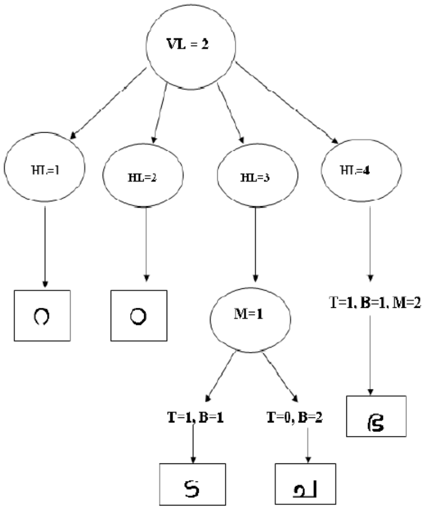


Fig. 3. Decision tree Classification of characters by horizontal lines where number of vertical lines equals 2.

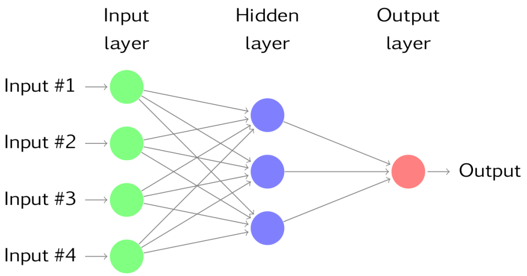


Fig. 4. Neural Network model

a function, the activations of these neurons are then passed on to other neurons. This process is repeated until finally, an output neuron is activated. This determines which character was read.

The neural net approach utilized three separate steps. The first step simply translated the binary character data into a friendlier form. The second step took the output of the first and trained a backpropagation network on it, outputting all the resulting weights and general network information. The third step took the output of the second and created a network. It then ran a full character set through the network and output identification information for all the characters the set contained.

Image digitization, is an essential step prior to neural networking. In this process, the input image is sampled into a binary window which forms the input to the recognition system. We assign a value +1 to each black pixel and 0 to each white pixel. Digitization of an image into a binary matrix of specified dimensions makes the input image invariant of its actual dimensions. Hence an image of whatever size gets transformed into a binary matrix of fixed pre-determined dimensions. This makes uniformity in the dimensions of the input and stored patterns as they move through the recognition system.

Backpropogation method of training artificial neural networks used with an optimization method. The method calculates the gradient of a loss function with respects to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function.

4) Support Vector Machines: Support vector machines are computational algorithms that construct a hyperplane or a set of hyperplanes in a high or infinite dimensional space. SVMs can be used for classification, regression, or other tasks. a separation between two linearly separable classes is achieved by any hyperplane that provides no misclassification on all data points of any of the considered classes, that is, all points belonging to class A are labeled as +1, for example, and all points belonging to class B are labeled as -1.

There are many hyperplanes that might classify the same set of data as can be seen in the figure 5 below.The objective of SVM is to find the best separation hyperplane, that is, the hyperplane that provides the highest marginal distance between the nearest points of the two classes. The main advantage of SVM as compared to other classifiers are they are robust, accurate and very effective even in cases where the number of training samples is small. SVM technique also shows greater ability to generalize and greater likelihood of generating good classifiers.

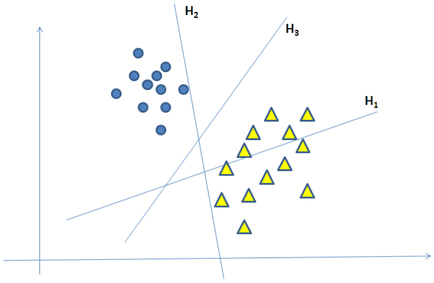


Fig. 5. SVM

In the above figure, H1 does not separate the two classes; H2 separates but with a very tiny margin between the classes and H3 separates the two classes with much better margin than H2.

5) Fuzzy Logic: The main idea behind a fuzzy system is to use the concept of linguistic variables to make decisions based on fuzzy rules and thereby get a better response compared to a system using crisp values. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The process of fuzzy inference involves: membership functions, fuzzy logic operators and if-then rules. Membership function is the mathematical function which defines the degree of an element's membership in a fuzzy set. For fuzzification we need to have fuzzy set for each input. These set is kind of linguistic variables such as relative terms: far, close, cold, warm, hot, small, normal, high, etc. Once when fuzzy set is defined, for each of the variables there has to be calculated degree of membership for every input. Minimum degree of membership is 0, and maximum is 1 or 100%. An example of membership function is shown in figure 6.

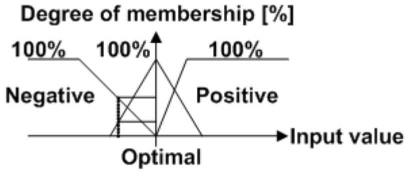
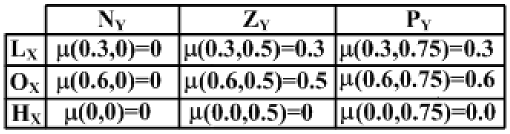


Fig. 6. Membership function

For example, if we have two inputs, x and y, and each of the inputs has 3 different fuzzy sets say low, optimal and high for x and negative, zero and positive for y. Now let x is 0.3 low and 0.6 optimal and 0 high. For y, let's say that y is 0 negative, 0.5 zero and 0.75 positive. Let's choose minimal value as "if-then rule". Then fuzzy-inference step for given data is shown in figure 7.



	N_Y	Z_Y	P_Y
L_X	μ(0.3,0)=0	μ(0.3,0.5)=0.3	μ(0.3,0.75)=0.3
O_X	μ(0.6,0)=0	μ(0.6,0.5)=0.5	μ(0.6,0.75)=0.6
H_X	μ(0,0)=0	μ(0.0,0.5)=0	μ(0.0,0.75)=0.0

Fig. 7. Fuzzy inference

Singleton constant values is given below.

L	L	M
L	M	M
L	M	H

So crisp output can be calculated when singleton table is mapped over fuzzy-inference data using the formula,

$$\text{out}=\frac{L\cdot 0.3+M\cdot 0.3+M\cdot 0.5+M\cdot 0.6}{0.3+0.3+0.5+0.6}$$

C. Existing System

Due to the complexity of the Malayalam character set, an efficient method for the recognition for handwritten characters has not been proposed till now. Based on Ostus algorithm for binarization, an OCR system called NAYANA was developed at C-DAC, Thiruvananthapuram [1]. It enables the user to convert printed Malayalam documents to editable files. In this system, projection profile method is used for skew detection and correction of image; and in the recognition phase linguistic rules are applied. An accuracy of 97 percentage was reported in this method. Its main features are:

- Images in BMP and TIFF formats are supported.
- Image should have more than 300dpi resolution.
- -5 to +5 is the skew correction angle.
- Output can be saved as TXT, HTML of RTF formats.
- Recognition speed of characters is 50 char/sec.

It's main applications are:

- Conversion of printed documents to editable text
- OCR combined with Text-To-Speech technology, can be used for text reading system.

Using wavelet based feature extraction and neural network based recognition, a new work was reported by M Abdul Rahiman and Rajasree[4]. Another work was reported by G Raju, [5] in which the daubechie wavelets (db4) were used for recognition. Another OCR system was proposed by Lajish V L, Suneesh T K and Narayanan N K [6] which was based on statistical classification. Recently a method for Handwritten Character Recognition was devised by Lajish V L [7].D. Trier, A.K. Jain and T. Taxt proposed a method for extracting features for Character Recognition[9].

The OpenBook is an optical character recognition system which converts text pages into an electronic text file and reads them with a voice synthesizer. The system consists of a scanner, processing unit with the DECtalk speech synthesizer, and a 17-key keypad. Features of the system include automatic page orientation, automatic contrast adjustment, decolumnization of multicolumn documents and the ability to recognize a wide variety of type faces and font sizes. Text scanned with the OpenBook can be saved in the system's document library for future use. The OpenBook is available in Standard, Deluxe, and Special models. This system is hardware independent of computer type. All versions of the OpenBook can be upgraded to a full featured IBM 386 or 486 computer. The OpenBook software is also available separately.

Amritha Sampath, Tripti C and Govindaru V [8] presented a Neural Network based model for handwritten recognition. The direction information of the written character is recorded based on the 8 connected Freeman chain code, one of the shape representation technique. The direction of the Pen movement from the contact is recorded as feature vector. Back propagation Neural Network is used for classifying characters. Additional disambiguation technique is used in post processing stage to identify confusing pairs.

III. Problem Definition

The input to the OCR system will be the scanned image of handwritten or printed text. There are different formats for representing images. Some of them are:

- TIFF
  - Tagged image file format
  - No Compression
  - Supported by image manipulation application
- BMP
  - No compression
  - Native file format of windows platform
  - Rich in colour
- JPEG
  - Joint photographic experts group
  - Supports 16 million colours
  - Information is lost from the original image due to compression
- GIF
  - Graphics Interchange Format
  - Limited to 256 colours
  - Allow compression

OCR is implemented in 3 basic steps. They are:

- Pre-processing
- Recognition
- Post-processing

Pre-processing consists of removal of noise and skew correction. Segmentation classifies the image into paragraphs, lines, words, and characters. The segmented characters are then fed to the recognition system. It performs feature extraction followed by classification using neural network. Thus, the code corresponding to the character is generated. Post processing includes error correction using linguistic rules to improve the accuracy of the generated code sequence. Its output will be the corresponding sequence of Unicodes.The sub-modules are described in detail in the later section.

A. *Algorithm*

- The scanned image is pre-processed before segmentation.
  - The grayscale image is converted into a binary (two-tone) image by applying threshold.
  - Noise is then removed from the image by applying Gaussian filter.
  - Skew detection and correction is performed.
    - Find the no.of white lines for every angle.
    - Find angle for which max white lines are produced.
    - Rotate the image by theta if it is less than ninety. Otherwise by 180 minus theta.
    - Output the skew corrected image.
- After pre-processing, the characters have to be segmented. First the lines are separated. It is followed by word and character separation.
  - Line separation
    - Image is scanned from top left pixel till the image width by changing x-coordinate.
    - If any black pixel is found, x and y positions are stored. Line is drawn to indicate the start of the line.
    - Repeat scanning by incrementing y-value to find width of the line.
    - Process is repeated till none of the pixels are black. Line is drawn to mark the end of line.
    - Repeat the above steps till image length.
  - Character separation
    - Scanning starts at top most pixel position of line document.
    - Pixel positions are scanned from top to bottom till end of line by changing y co-ordinate.
    - If any black pixel is found, x andy positions are stored.
- Each character is represented as an mxn matrix.
- This matrix is given to the neural network during the training phase.
- Each character will be mapped to corresponding Unicode.
- In post-processing, linguistic rules are applied to correct errors.
- The final text is obtained in TXT or RTF format.

IV. System Architecture

The block diagram of OCR system is shown in Figure 8.

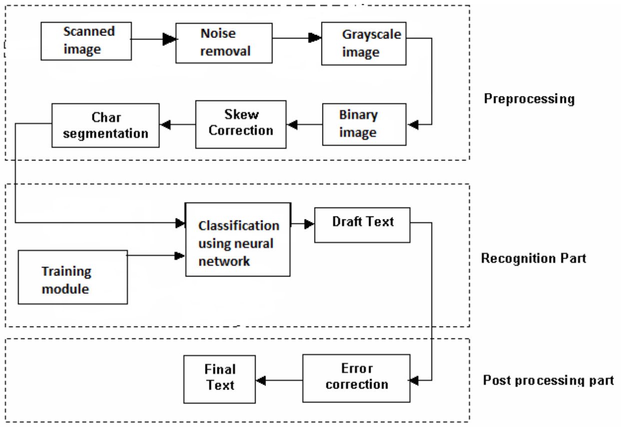


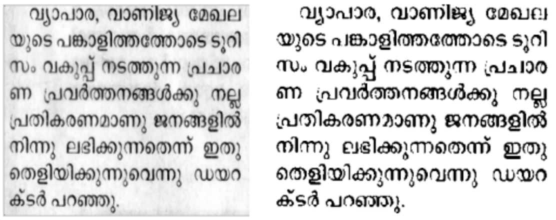
Fig. 8. Steps in OCR

A. *Pre processing*

The first stage pre-processing includes the following steps:

- Binarization
- Noise removal
- Skew correction

1) Binarization: A printed document is first scanned and is converted into a gray scale image. Binarization is a technique by which the gray scale images are converted to binary (two-tone) images. It separates the foreground and background informations. The most common method for binarization is to select a proper intensity threshold for the image and then convert all the intensity values above the threshold to one intensity value, and to convert all intensity values below the threshold to the other chosen intensity. Thresholding are of 2 types- global threshold and local threshold. Global method apply one threshold value to the entire image. Local threshold method apply different threshold values to different regions of the image. Figure 9 shows grayscale and binary image.



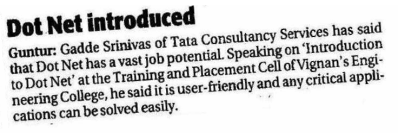
വ്യാപാര, വാണിജ്യ മേഖല
യുടെ പങ്കാളിത്തത്തോടെ ടൂറി
സം വകുപ്പ് നടത്തുന്ന പ്രചാര
ണ പ്രവർത്തനങ്ങൾക്കു നല്ല
പ്രതികരണമാണു ജനങ്ങളിൽ
നിന്നു ലഭിക്കുന്നതെന്ന് ഇതു
തെളിയിക്കുന്നുവെന്നു ഡയറ
ക്ടർ പറഞ്ഞു.

വ്യാപാര, വാണിജ്യ മേഖല
യുടെ പങ്കാളിത്തത്തോടെ ടൂറി
സം വകുപ്പ് നടത്തുന്ന പ്രചാര
ണ പ്രവർത്തനങ്ങൾക്കു നല്ല
പ്രതികരണമാണു ജനങ്ങളിൽ
നിന്നു ലഭിക്കുന്നതെന്ന് ഇതു
തെളിയിക്കുന്നുവെന്നു ഡയറ
ക്ടർ പറഞ്ഞു.

Fig. 9. Grayscale and Binary image

2) Noise removal: Scanned documents often contain noise that arises due to printer, scanner, age of the document, etc. Therefore, it is necessary to filter this noise before we process the image. Commonly used approach is to process the image through a low-pass filter. Here we are using a gaussian filter to remove noise.

3) Skew correction: When a document is fed to the scanner either mechanically or by a human operator, a few degrees of skew (tilt) is unavoidable. Skew angle is the angle that the lines of text in the digital image make with the horizontal direction. Figure 10 shows an image with skew. Projection profile based technique is one of the popular skew estimation technique. A projection profile is a one-dimensional array with a number of locations equal to the number of rows in an image. Each location in the projection profile stores a count of the number of black pixels in the corresponding row of the image.



Dot Net introduced

Guntur: Gadde Srinivas of Tata Consultancy Services has said that Dot Net has a vast job potential. Speaking on 'Introduction to Dot Net' at the Training and Placement Cell of Vignan's Engineering College, he said it is user-friendly and any critical applications can be solved easily.

Fig. 10. An image with skew

4) Character Segmentation: Once the document image is binarized and skew corrected, actual text content must be extracted. Horizontal scanning of the document image is done to extract the lines from the document. If the lines are well separated, the horizontal projection will have separated peaks and valleys, which serve as the separator of lines. Similarly, gaps in the vertical projection of a line image are used in separating words in a line, as well as for separating individual characters from the word[10].

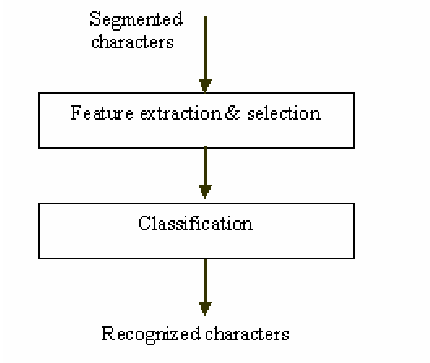


Fig. 11. Recognition Process

B. Recognition

By character recognition, the character symbols of a language are transformed into symbolic representations such as ASCII, or Unicode[2]. The basic problem is to assign the digitized character into its symbolic class. This is done using training and classification. Figure 11 shows the basic recognition process.

1) Classification: Classification process recognizes individual characters and outputs them in machine editable form. Classification can be done using decision tree, neural network, svm, bayesian classifier or fuzzy logic. Before that each character has to be represented in binary form. This is given as the input to neural network. Figure 12 shows the training phase. The training set involves the binary codes of alphabets. Considering a back-propagation neural network, the network is composed of several layers of interconnected elements. A feature vector enters the network at the input layer. Each element of the layer computes a weighted sum of its input and transforms it into an output by a nonlinear function[11].During training, the weights at each connection are adjusted until a desired output is obtained. Following are the major steps of Back-propagation algorithm.

- Randomly choose the initial weights
- While error is too large
- For each training pattern (presented in random order)
- Apply the inputs to the network
- Calculate the output for every neuron from the input layer, through the hidden layer(s), to the output layer
- Calculate the error at the outputs
- Use the output error to compute error signals for pre-output layers

- Use the error signals to compute weight adjustments
- Apply the weight adjustments

A problem of neural networks in OCR is their limited predictability, while an advantage is their adaptive nature. Figure 13 shows the unicode for Malayalam.

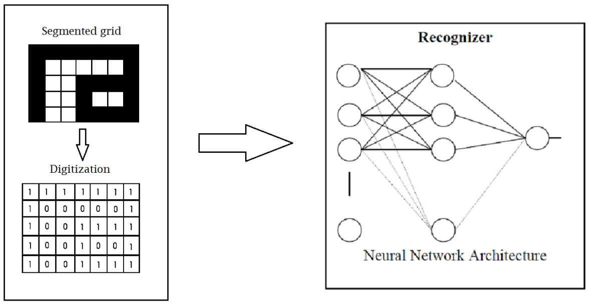


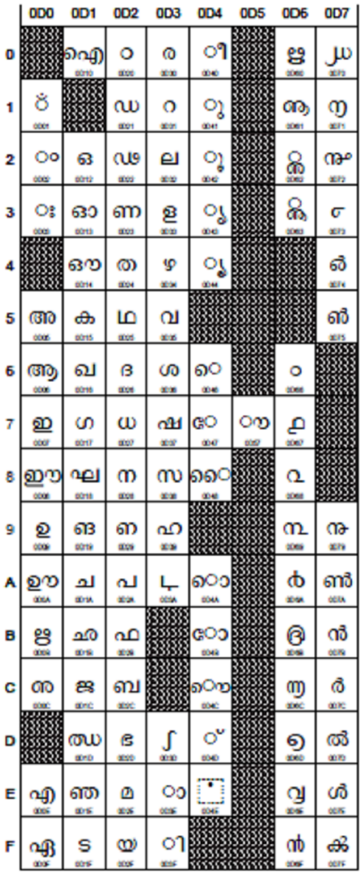
Fig. 12. Training

C. Post processing

Linguistic rules are applied to the recognized text in the post processing module to correct classification errors i.e, certain characters never occur at the beginning of a word and if found so, they are remapped appropriately. Similarly dependent vowels will appear only with consonants and independent vowels occur only at the beginning of a word.

V. Conclusion and Future Work

In this paper, we have proposed an algorithm to recognize the printed and handwritten Malayalam characters on a scanned text to generate an editable document. Various available techniques are studied to find a best technique. The techniques which provide beter results are slow in nature while fast techniques mostly provide inefficient results. It is found that the OCR techniques based on neural network provide more accurate results than any other techniques. In the proposed work, the image of the text to be recognized is given as input. The system separates it into lines, words and then characters. Pre-processing is done to improve the accuracy of recognition of characters. After recognition, post-processing is done to correct errors caused if any. The major factor to be taken care of is the similarity in character shapes of certain Malayalam characters. The future scope of our work lies in the recognition of compound Malayalam characters, extraction of text from video images and processing of historical documents.



	0D0	0D1	0D2	0D3	0D4	0D5	0D6	0D7
0		ഐ	ഠ	ര	◌ീ		ൠ	൰
1	◌ഁ		ഡ	റ	◌ു		ൡ	൱
2	◌ം	ഒ	ഢ	ല	◌ൂ		ൢ	൲
3	◌ഃ	ഓ	ണ	ള	◌ൃ		ൣ	൳
4		ഔ	ത	ഴ	◌ൄ			൴
5	അ	ക	ഥ	വ				൵
6	ആ	ഖ	ദ	ശ	◌െ		൦	
7	ഇ	ഗ	ധ	ഷ	◌േ	◌ൗ	൧	
8	ഈ	ഘ	ന	സ	◌ൈ		൨	
9	ഉ	ങ	ഩ	ഹ			൩	൹
A	ഊ	ച	പ	ഺ	◌ൊ		൪	ൺ
B	ഋ	ഛ	ഫ		◌ോ		൫	ൻ
C	ഌ	ജ	ബ		◌ൌ		൬	ർ
D		ഝ	ഭ	ഽ	◌്		൭	ൽ
E	എ	ഞ	മ	◌ാ	ൎ		൮	ൾ
F	ഏ	ട	യ	◌ി			൯	ൿ

Fig. 13. Malayalam Unicode

References

[1] K.Jithesh,K.G.Sulochana,R.Ravindra Kumar,*“ Optical Character Recognition for Malayalam ”*

[2] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh,*“ Text Detection and Character Recognition in Scene Images ”*

[3] Rafael C Gonzalez, Richard E Woods,*“Digital Image Processing”*

[4] M Abdul Rahiman and M S Rajasree, *“Printed Malayalam Character Recognition Using Back propagation Neural Networks”* Proc.of IEEE International Advance Computing Conference (IACC 2009), Patiala, March 2009.

[5] G Raju, *“Recognition of unconstrained handwritten Malayalam characters using zero crossings of wavelet coefficients”* Proc. of International Conference on Advanced Computing and Communications, ADCOM, pp 217-221, Dec 2006.

[6] Lajish V L,Suneesh T K K and Narayanan N K, *“ Recognition of Isolated handwritten images using Kolmogorov-Smirnov Statistical classifier and K nearest neighbor classifier”* Proc. Of International Conference on Cognition and Recognition, Mandya, Karnataka, December, 2005.

[7] Lajish V L, “ Handwritten Character Recognition using perpetual Fuzzy zoning and Class modular Neural Networks” Proc. of fourth International Conf on Innovations in IT, 2007.

[8] Amritha Sampath, Tripti C and Govindaru V, *“Freeman code based online handwritten character recognition for Malayalam using Back Propagation Neural Networks”* Advanced Computing: An International Journal, Vol.3 No.4, July 2012.

[9] D. Trier, A.K. Jain and T. Taxt, *“Feature Extraction Methods for Character Recognition-A Survey”* Pattern Recognition, vol. 29, no. 4, June 1996.

[10] Abdul Rahiman M, M S Rajasree, Masha N, Rema M , Meenakshi R, Manoj Kumar G,*“Recognition of Handwritten Malayalam Characters using Vertical & Horizontal Line Positional Analyzer Algorithm”* IEEE, May 2011.

[11] B. Hussain, and M. R. Kabuka, *"A novel feature recognition neural network and its application to character recognition"*, IEEE Transactions of Pattern Recognition and Machine Intelligence, Vol.16, No. 1, 1994.