



Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

**ScienceDirect**

Procedia Computer Science 57 (2015) 1104 – 1114

**Procedia**  
Computer Science

3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)

**COMBINING BITEMPORAL CONCEPTUAL DATAMODEL WITH MULTIWAY  
JOIN RELATIONS FOR FORECASTING**

**A.Christy<sup>a</sup>, MeeraGandhi.G<sup>b</sup>**

<sup>a</sup> Professor, Faculty of Computing, Sathyabama University , Jeppiaar nagar, Rajiv Gandhi Salai, Chennai , India

<sup>b</sup> Professor, Faculty of Computing, Sathyabama University , Jeppiaar nagar, Rajiv Gandhi Salai, Chennai, India

---

**Abstract**

The Conceptual Modelling deals with representing an application domain in a descriptive and consistent manner without any computer metaphor. In this paper, we have modelled forecasting a temporal database adopting the concept of Bitemporal Conceptual Data Modelling considering the domain of Stock Exchange. The Novelty of this work is to optimize and execute data from multiple heterogeneous data sources by organizing the data through various transformations for preprocessing. We propose an algorithm called enhanced 2P-TO (2 Phase Transformation Ordering) for efficient organization of non-overlapping sets of transformations. The salient feature of this approach lies in the consideration of direct, heterogeneous links among transformations and multiple data resources. Experiments are conducted using NASDAQ data which has 33280 tuples with 6 major attributes. The efficiency of the algorithm is studied using two parameters: cardinality and number of correlated attributes. The correlation between the attributes is studied graphically using descriptive analysis. Forecasting of share trend is done for various attributes of the dataset and the best attribute for forecasting is found by adjusting its smoothing factors.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of organizing committee of the 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)

*Keywords:* BiTemporal Conceptual Data Model; Multi-way join; Forecasting; Temporal databases; Transformations; Optimization; 2-Phase Transformation Ordering ; Execution Plan ; etc

---

## 1. Introduction

Conceptual data models describe an application domain in a declarative and reusable way by optimizing the set of data required for that particular domain. E-R diagrams for Relational Database Management Systems, UML, and ODMG for Object Oriented data model and RDF and OWL for Web Ontology are some of the representations used for Conceptual data modelling. This paper deals with developing a declarative bitemporal conceptual data model considering the domain of Stock Exchange.

Temporal Data Mining plays an important role in applications related to Stock exchange, Weather data analysis, Weblog data analysis, Customer Transaction data, Clinical Data Analysis, etc. Temporal data have a unique structure of High Dimension and High feature correlation in capturing time dependent data. In Temporal Conceptual Data modelling, the attribute Time plays a major role as it records the history of database states rather than the entire history.

## 2. Related Works

Optimization of complex queries for distributed applications using filter transformation[2]. Bayesian Probabilistic reasoning apparatus used for removing concepts such as outlier, noise, distribution drift, novelty detection, rare events and unexpected events in a video annotation system[3]. Remi Emonet et al developed a model which represents unsupervised discovery of recurrent temporal patterns in time series using Bayesian methods[4]. Yongnian Zhang et al has shown dealing with multiple primitive events happening in parallel or sequentially over a period of time. The authors have defined a framework that combines the Bayesian network with the interval algebra to model temporal dependencies and advanced machine learning methods are introduced to learn the model structure and parameters over time periods[5].

Bin Liu and Elke A. Rundensteiner represented data in materialized view after extracting from heterogeneous data sources[6]. Yanchang Zhao and Shichao proposed dimension reduction framework for time series data[7]. Jun-ichi takeuchi and Kenji Yamanishi has defined an unifying framework for detecting outliers and changepoints for time series data[8]. Carlos Ordonez proposed integrated K-Means clustering with relational DBMS for SQL Database[9]. Christian S. Jensen et al has discusses dependency theory to Temporal Databases[10].

Byunk Suk Lee and Gio wiederhold defined an approach for integrating object-oriented programs with databases by evaluating view queries[11]. The query is used to materialize the necessary data into a relation from database and the function is used to restructure the materialized relation into objects. While instantiating objects and dealing only with inner join, it often happens that some tuples may get lost, whereas outer joins do not discard any null values. Outer join, retrieve data from the outer table padding the missing data with Null values. Sometimes the table will be overloaded with Null values. Without optimization declarative approaches such as SQL queries and views are not practical. For eg) Consider two users, user1 and user2 who define Inventory object differently as:

```
Inventory (itemname, itemdescription) // defined by user1
Inventory(itemname, itemprice) // defined by user2
```

In the object-storage approach, the two objects are stored separately. To provide sharing requires a separate mechanism for identifying the owners.

In the relation storage approach, this problem does not occur, because the information to support the two Inventory objects is stored in a single relation.

Inventory (itemname, itemdescription, itemprice) and their owners are distinguished by the database view mechanism. Morgan D. Monger et al proposed temporal data management in NoSQL Databases[12].

E. Malinowskia et al introduces a temporal extension of the MultiDim model. This extension is based on research realized in temporal databases. It discusses different temporality types: valid time, transaction time, and lifespan, which are obtained from source systems, and loading time, which is generated in the data warehouse[13]. K.Murugan et al proposes intelligent query processing using context free grammar[14].

In this digital world, the number of shares and number of users trading online increases day by day. Accumulating data, preprocessing and organizing the required data for forecasting is an important challenge. The application is modelled using bitemporal conceptual model considering time and facts. The model is verified with enhanced 2P-TO algorithm which organizes required data from multiple heterogeneous datasources. Forecasting is done with various attributes and the attribute which matches the best criterion is further to be suggested among these attributes for forecasting.

## 2.1 Associating Time with Facts

Facts are associated with time, i.e., with one or more dimensions of valid time and transaction time. One valid time and one transaction time may be related in various ways, resulting temporal specialization. Multiple transaction times arises when a fact is stored in one database, then later replicated or transferred to another database. By retaining the transaction times as temporal generalizations, the original relation can be effectively queried by referencing only the final relation.

Time-varying data looks simple, but instead of having one value, there is a value for each instant of time. Researchers have found that Events and states can also be associated with time. A *state* is something that has extent over time. An *event* is instantaneous, which happens being true over time. Events delimit state. The occurrence of an event brings a fact into true. On the other hand, the occurrence of another event can generalize that fact no longer valid. Hence, events and states are duals; states can be represented by their delimiting events, and events are implied by states. The database facts have at least two relevant temporal aspects *valid time* indicating when the fact was true and *transaction time* specifying when a fact was current in the database. The time slice at any time yields the conventional relation at that time.

The temporal data mining inclines towards finding the interaction between valid and transaction time. In the End of Day dataset collected from NASDAQ, we had seven attributes as (Identifier, Date, Open, High, Low, Close and Volume) the relation is updated precisely (at a granularity of days) when reality changes. Certain days when there are no changes in the value of share, the data is assumed to be always up to date at the end of each day otherwise, the time slice might not yield the correct result. A time slice at any time in the past yields what the database stored as current at that time.

A temporal relation is *retroactive* if the facts stored by the tuples are valid before they are entered into the relation, i.e., the facts became true before they were stored. Retroactive relations are common in monitoring situations where attributes from the dataset such as *Volume* and *High* are periodically sampled and stored in a database for subsequent analysis.

The advantage of using Binary Conceptual Data Model is to retain the simplicity of the relational model with temporal aspects of the facts stored in the database. This could be achieved by relating with each relational data tuple, a region in the space spanned by transaction time and valid time that defines the temporal concepts of the tuple. The Temporal Modelling is distinguished into two main categories as *Time Stamping* and *Evolution Constraints*. Time Stamping deals with the discrimination at the schema level between elements of the model that change over time while the remaining elements are time invariant. Time Stamping can be applied to Classes, Attributes and Relationships. For example, in Stock Exchange data, the value of a share will be between a range for a particular period and it may increase or drop after certain time slot. Evolution Constraints control the mechanism that rules dynamic aspects, i.e., the possible transitions from one state of the database to the next one. Object migration deals with evolution of an object from being member of a class to being member of another class.

A time interval is defined as the time between two instants, a starting and a terminating instant. A time interval is then represented by a sequence of consecutive chronons, where each chronon represents all instances that occurred during the chronon. A *chronon* is the shortest duration of time supported by a temporal DBMS. It is a non-decomposable unit of time. A sequence of chronons can be represented simply by the pair of the starting and terminating chronon. Unions of intervals are termed *temporal elements*

The domain of valid times is given as  $D_{VT} = \{c_1^v, c_2^v, \dots, c_k^v\}$  and the domain of transaction times may be given as  $D_{TT} = \{c_1^t, c_2^t, \dots, c_k^t\}$ . A valid-time chronon  $c^v$  is thus a member of  $D_{VT}$ , a transaction-time chronon  $c^t$  is a member of  $D_{TT}$ , and a bitemporal chronon  $c^b = (c^t, c^v)$  is an ordered pair of a transaction time chronon and a valid-time chronon.

A set of explicit attributes are defined by  $D_A = \{A_1, A_2, \dots, A_{nA}\}$  and a set of domains for this attributes are defined by  $D_D = \{A_1, A_2, \dots, A_{nD}\}$ . For these domains, the symbols  $\perp_i$ ,  $\perp_u$  and  $\perp$  are defined in this domain to represent inapplicable, unknown and inapplicable or unknown null values. The schema of a bitemporal conceptual relation,  $R$ , consists of an arbitrary number, e.g.,  $n$ , of different explicit attributes from  $DD_A$  with domains in  $DD_D$  and an implicit timestamp attribute  $T$  with domain  $2^{(D_{TT} \cup \{UC\})} * D_{VT} \setminus \Phi$ , where  $UC$  (“time changed”) is a special transaction time maker. A value  $(UC, c^v)$  in a timestamp for a tuple indicates that the tuple being valid at time  $c^v$  is current in the database. A tuple  $(a_1, a_2, \dots, a_n | t^b)$  in a bitemporal conceptual relation instance,  $r(R)$  consists of a number of attribute values associated with a bitemporal timestamp value. Depending on the extent of decomposition, such a tuple may be thought of as encoding an atomic or a composite fact.

In graphical representations of bitemporal space, we choose the  $x$ -axis as the transaction-time dimension and the  $y$ -axis as the valid-time dimension. Hence, the ordered pair  $(C^t, C^v)$ . This represents the bitemporal chronon with transaction time  $C^t$  and valid time  $C^v$ . Valid-time relations and transaction-time relations are special cases of bitemporal relations that support only valid time or transaction time, respectively. Thus a valid-time tuple has associated a set of valid-time chronons (termed a *valid time element* and denoted  $t^v$ ), and a transaction-time tuple has associated a set of transaction-time chronons (termed a *transaction-time element* and denoted  $t^t$ ). For clarity, we use the term *snapshot relation* for a conventional relation. Snapshot relations support neither valid time nor transaction time. In general, we use the term *snapshot relation* for a conventional relation. Snapshot relations support neither valid time nor transaction time.

## 2.2 Temporal Database Transformation for Multi Join relations

In temporal data mining related to Stock market data, two types of temporal rules identified are snapshot and aggregate. Snapshot temporal rules are formulated by associating a current decision attribute instance to the relevant past condition attribute instances. Aggregate temporal rules are formulated by associating a current decision attribute instance to the total change of relevant past condition attribute instances. Any chronologically based data can be discretized in this manner to produce snapshot and aggregate temporal rules.

The Multi Join relations are preprocessed by cleaning and equalizing Sampling cycle. In the Stage of Preprocessed pattern extraction, attribute selection is applied for the purpose of aggregate and filter transformations. Temporal Databases are based on the concept of translating temporal queries into standard relational queries both logically and physically. Removing the noise in the data, identifying the abnormal patterns and predicting the patterns having desired feature is one of the important challenge of Temporal Data Mining. For instance, the volume of transactions occurring on a supermarket will be high during a specific period of time on a day and will be more during certain days on a month (probably weekends). Identifying the periodic patterns could reveal some interesting information regarding the behavior and future trends of the case which can help in efficient decision making. Faraz Rasheed et al describes the periodicity detection is a process for finding temporal regularities within the time series and the goal of analyzing a time series is to find whether and how frequent a periodic pattern is repeated within the series. This paper deals with NASDAQ data collected from eoddata.com for 15 days. Forecasting is done after preprocessing through a predefined set of transformations in work flow management.

## 2.3 Optimization of Join Relations

The problem deal with the ordering of transformations which minimizes the response time of the multiway JOINS relations. The motivation of joining relations is based on the fact that vast number of applications of today is based on combining data from heterogeneous databases. For example, an application that performs meteorological studies

may have to combine data from different sensor deployments, where each sensor deployments forms various types of measures such as Filter, Indirect Load, Aggregate, Router, Expression and so on. A filter may be implemented using filter condition or it is performed by look up services. A look up service takes one or more data values as input and performs Key-Foreign Key joins with a base data source, i.e., it checks whether a input parameterizing with a time period [  $\zeta_l, \zeta_u$  ], we get all related information[2]. Temporal databases are based on the concept that they are clustered [physically and logically] as they filter out portion of the stored data based on some criteria.

**2.3.1 Algebraic Operators in the Bitemporal Conceptual Data Modelling**

Define a relation schema  $R = (A_1, \dots, A_n \mid T)$  and let  $r$  be an instance of this schema. Let  $D$  be an arbitrary set of explicit (i.e., non-timestamp  $\pi^B_D r$  is defined as follows :

$$\begin{aligned} \pi^B_D r = \{ & Z^{(D)+1} \mid \exists x \in r (Z [D]) = x [D] \wedge \\ & \forall y \in r (y [D]) = z [D] \rightarrow y [T] \subseteq Z [T] \wedge \\ & \forall t \in Z [T] \exists y \in r ((y [D]) = z [D] \wedge t \in y [T]) \} \end{aligned} \tag{1}$$

As an example, our NASDAQ dataset had data for 15 days for 2863 companies. The entire dataset should be optimized for efficient storage. Thus, we have grouped the identifiers for various days under one category and only one entry is maintained for each identifier which indicates Maximum High and Maximum Volume. All of the bitemporal chronons associated with a tuple with a identifier for different dates are merged into one bitemporal element. Each chronon of this bitemporal element must be in the timestamp of atleast one among Maximum High or Maximum Volume in the underlying relation.

Let  $P$  be a predicate defined on  $A_1, \dots, A_n$ . The selection  $P$  on  $r, \sigma^B_D (r)$  is defined which stores the aggregated value of the instance

$$\sigma^B_D (r) = \{ Z \mid Z \in r \wedge P(Z[A]) \} \tag{2}$$

$\sigma^B_D (r)$  simply performs the familiar snapshot selection, with the addition that each selected tuple carries along its timestamp,  $T$ . A new tuple variable,  $z$ , is used for holding the projected result tuples. It also ensures that no chronon in any value-equivalent tuple of  $r$  is left unaccounted for, and the second line ensures that no spurious chronons are introduced in  $Z$ 's timestamp. A bitemporal relation thus results.

There are two operators that select on valid time and transaction time. They have no counterparts in the snapshot relational algebra. Let  $c^v$  denote an arbitrary valid-time chronon and let  $c^t$  denote a transaction-time chronon. The *validtimeslice* operator ( $\tau^B$ ) yields a transaction-time relation and the *transaction-timeslice* operator ( $\rho^B$ ) evaluates to a valid-time relation

$$\tau^B_{c^v} = \{ Z^{n+1} \mid \exists x \in r (Z[A] = x[A]^{Z[T]} = \{ C^t \mid (C^t, C^v) \in X[T] \} \wedge X[T] \neq \Phi) \} \tag{3}$$

$$\rho^B_{c^t} = \{ Z^{n+1} \mid \exists x \in r (Z[A] = x[A]^{Z[T]} = \{ C^v \mid (C^t, C^v) \in X[T] \} \wedge X[T] \neq \Phi) \} \tag{4}$$

Thus  $\tau^B_{c^v}$  simply returns all tuples in  $r$ , that were valid during the valid-time chronon  $c^v$ . The timestamp of a returned tuple is all transaction-time chronons associated with  $c^v$ . Similarly,  $\rho^B_{c^t}$  performs the same operation, except the selection is performed on the transaction time  $C^t$ . The type of the result is different from the type of the argument relation. Here, a time dimension has been removed while for projection, a number of explicit attributes were removed. Thus, the definitions must again employ a separate tuple variable (i.e., variable  $z$ ) in order to construct the result tuples

## 2.4 Cost-Based Analysis

The optimization plan can be enhanced by incorporating relevant cost information in terms of total processing time. This can be done by incorporating two factors :

1. Cardinality of the source relation  $R_i$
2. Number of attributes in the source relation  $R_i$ .

Let  $T_{ij}$  (or  $T_{ji}$ ) estimates the processing time for the join condition  $\bowtie_{ij}$  at the data source  $R_i$  and let  $\sigma_{ij}$  estimate the selectivity of the join operations between data sources  $R_i$  and  $R_j$ .

## 3. System Description

Consider  $L$  data sources. Each data source produces a data stream  $X^j$ ,  $j = 1, 2, \dots, L$  of finite length. It is mandatory initially that the data from each source should be joined together i.e., using aggregator by grouping based on identifier. The processing cost of each tuple in  $X^j$  corresponds to a query call. Let  $W^j$  be the set of functions that process data from  $X^j$ . The average cost to process an input tuple is  $C_i^j$ .

For the data source services, this cost may correspond to the cost of producing tuples, which is the inverse of the tuple production rate. For the join relation, the average processing cost may differ for tuples originated from different sources. Costs and selectivities are independent of each other, for example, the selectivity of a filtering or a join relation. Goal is to build a plan  $P$  that has the minimum response time.

Table 1 2P-TO algorithm

---

```

Inputs :  $W^j, j = 1, \dots, L$  : The set of transformations that process the data of  $X^j$ 
Outputs : An optimized multi-way Join Plan  $P$  which removes bottle neck
exit  $\leftarrow$  false
let  $W^k$  be the set of alternate transformations that contains the multiway join  $T_{join}^k$ 
 $C \leftarrow \phi$  ; {  $C$  is the linear sub plan of  $P$  consisting of transformations placed after the multiway join  $T_{join}^k$  }
for  $j = 1, \dots, L$  do
if  $j \neq k$ , i.e., the transformation set  $W^j$  does not contain the multiway join  $T_{join}^k$  then
build a linear transformation plan  $L^j$  utilizing both the transformations in  $W^j$  and  $T_{join}^k$  imposing the constraint that  $S_{join}^k$  is always
placed after any transformation in  $W^j$  in the resulting plan, i.e.,  $T_i^j < T_{join}^k$ 
let  $L^j$  be the resulting linear transformation plan without the multiway join transformation  $T_{join}^k$ 
else
build a linear transformation plan utilizing the services in  $W^k$  without imposing any additional precedence constraint
let  $L^k$  be the part of the ordering of  $W^k$  before  $T_{join}^k$ 
if one or more transformations of  $W^k$  are placed after  $T_{join}^k$  in the output linear plan then
 $C$  is assigned to the part of the ordering of  $W^k$  after the multiway join transformation
end if
end if
end for
Create a multiway join plan  $P$  using the sub plans  $L^j, j = 1, \dots, L$  and  $C$  by connecting the output of the last transformation of every  $L^j$ 
with one of the inputs of the multiway join

```

---

Efthymia Tsamoura and Anastasios Gounasis (2012) have proposed an algorithm called 2P-SO (2 Phase-service ordering) for building multi-way join plans in services algorithm. The 2P-TO(2 Phase-Transformation Ordering) algorithm presented in table 1 is the modified version concentrated on transformations on a single phase. Given NASDAQ data, attempt is made to define the transformations required. Informatica ETL tool is taken for evaluating the available plans. Given the Input as the set of available transformations, the algorithm extracts the various available plans as discussed in table 2 and the plan with the minimum execution time is extracted as the optimal plan. Bottleneck occurs when data input is taken from a relational writer.

Table 2. Various execution plans

---

Input ( $X^i$ ) : Set of available transformations like (Source Qualifier, Filter, Router, Expression, Aggregator, Joiner, Lookup, Union)

Three different plans are suggested includes

Plan 1 :  $X^i$  ..... Union ..... Aggregator ..... Filter .... Filter ...  
 Plan2 :  $X^i$  ..... Source Qualifier ..... Lookup ... Expression .....  
 Plan 3 : Indirect Load ..... Merged flat file  
           Merged Flat file .... Filter ..... Expression ..... Aggregator

---

Under Plan 1, using UNION operator the master file and detail file are taken for union. As the scalability of the relations increases, UNION operators make the system down to a crawl and increases bottleneck. To decrease bottleneck, consider the relation with less number of tuples as the master file, so that the file with less number of tuples are stored in cache. The contents of detail file are compared with cache contents. If the tuple already exists, it is ignored else it is written to the cache. After comparing with all contents of detail relation, the contents of the cache is then written back to master file.

Under Plan 2, the filter condition is given in the Source Qualifier. Using a Connected Lookup transformation, the contents of the relation are retrieved and passed through an expression for evaluation.

Under Plan 3, the transformation Indirect Load is used, which can fetch multiple flat files into a master file, which can be further queried using transformations like Filter, Expression and Aggregator.

#### 4. Results and Discussion

In the experimental setup, 15 data sources of NASDAQ from [www.eoddata.com](http://www.eoddata.com) is collected from 1/7/2014 to 22/7/2014 for different identifiers. The data sources are available as flat files each having similar attributes as <identifier>, <date>, <open>, <high>, <low>, <close> and <volume>. All required relations are preprocessed by cleaning and equalizing sampling cycle. In the dataset selected for this approach less than 2% of data only required preprocessing. In the stage of pattern extraction, attribute selection is applied for the purpose of aggregate and filter transformations. In plan 1, by applying Union transformation, all the data sources are merged together yielding 33280 different tuples. All individual source flat files are mapped into a target file using Source Qualifier. Aggregation is done for maximum sale of shares grouping both identifier and date in ascending order. If 20% overall aggregate value is less than 20%, proposal can be given to sell the shares otherwise to buy shares. Fig. 1 shows the behavior patterns of 8 series of data discussed under applying through transformations like Aggregator on and Filter. The condition in filter transformation is given in such a way that none of the tuples fall behind 20%. In plan 3, by indirect load, all source flat files are loaded into a target file. Taking target file as the source file, expression transformation is applied to check if variation in sale is more than 20%, through aggregation transformation, relations are grouped based on identifier and date. Through router transformation, variation in sale more than 20% is derived in one target file and remaining data are collected in the next target file. The figure 1a shows the distribution of data populated after plan I and figure 1b shows the distribution of data populated after plan 3. The Volume of shares is found to be the maximum during weekends for all the identifiers and Figure 2 shows that the execution time is 10 seconds for plan 3, which is the optimized plan which is discussed in Table 2.

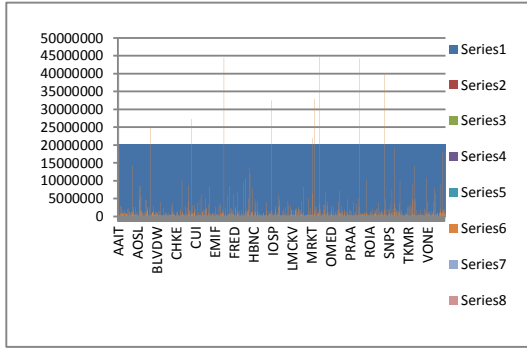


Fig. 1a Trend analysis after execution through Plan 1

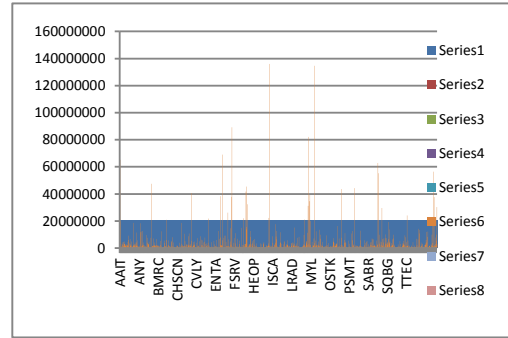


Fig. 1b Trend analysis after execution through Plan 3

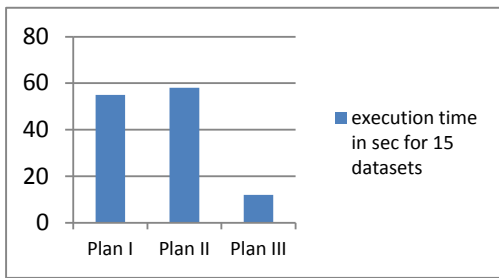


Fig. 2 Comparison of execution time

Before Forecasting, the correlation of the attributes are studied in descriptive manner as shown in Fig. 3(a), 3(b), 3(c),3(d), 3(e) and 3(f). The diagram which depicts the relation between date and High as in 3(a), reveals the fact sale is high during weekends. Taking High and Volume as attributes as in Fig. 3(b) and residual plot in 3(c) the same fact is evident and also all others values are distributed around the mean. The dataset doesnot have outliers.

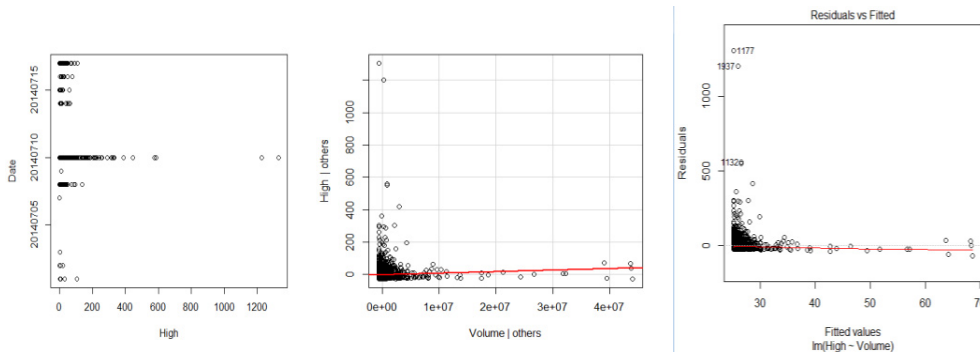


Fig. 3 a) Date Vs High

Fig. 3 b) High Vs Volume

Fig. 3 c) Residuals

The curve is fitted through Regression and the results obtained through QQ plots and residuals and Cook's distance measures how much an observation influences the overall model or predicted values shows the points lying within the boundary for the selected attributes as shown in Fig. 3(d), 3(e) and 3(f)



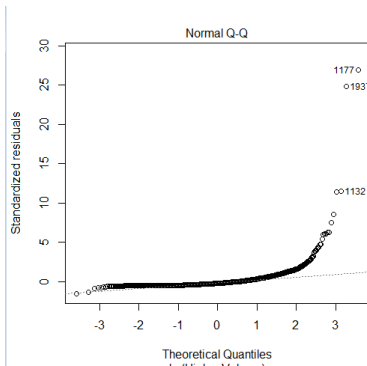


Fig. 3(d) Normal Q-Q Plot

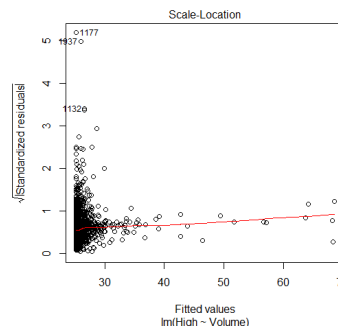


Fig. 3(e) Fitted Values

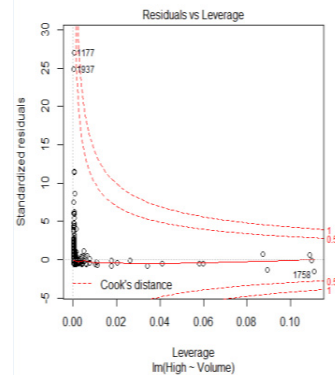


Fig. 3(f) Residuals Vs Leverage

Taking the attributes Date, Volume, High and Low as the most suitable attributes for forecasting, prediction is done and selecting Date as the attribute for forecasting is considered as the best predictor which shows the least Mean Error, Root Mean Squared Error and Mean Absolute Error as in Table 3.

Table 3. Prediction error values

Attribute	Smoothing Parameters	ME	RMSE	MAE
Date	alpha = 0.2657 Initial state l = 20140709.994 sigma: 0	7.071575e-06	1.301127	0.4320202
Volume	alpha = 1e-04 beta = 1e-04 phi = 0.9374 Initial states: l = 10085413.0998 b = -635213.4602 sigma: 2478216	10894.07	2478216	784702.1
High	alpha = 1e-04 Initial states: l = 25.8115 sigma: 48.3842	0.02289647	48.38425	21.06311
Low	alpha = 1e-04 Initial states: l = 25.0835 sigma: 47.6379	0.08876921	47.63792	20.60272

Forecasting is using the attributes Date, Volume, High and Low for the years 2050,2100, 2150,2200 and 2250, the major variation is seen in NASDAQ, ENDOFDAY Data collected from (www.eoddata.com ) only after the year 2100 as shown in figures 4 a, 4 b, 4 c and 4 d.

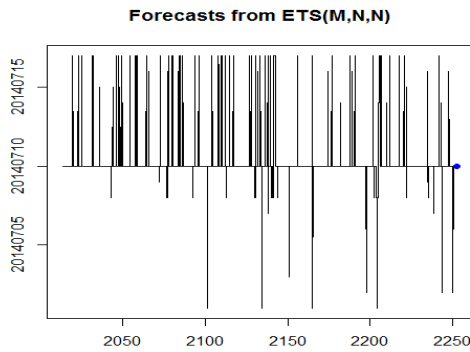


Fig. 4(a) Forecast for Date as attribute

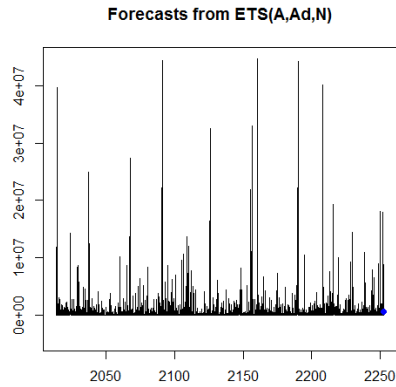


Fig. 4(a) Forecast for Volume as attribute

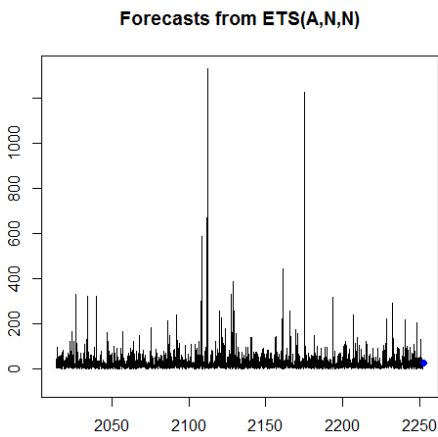


Fig. 4(c) Forecast for High as attribute

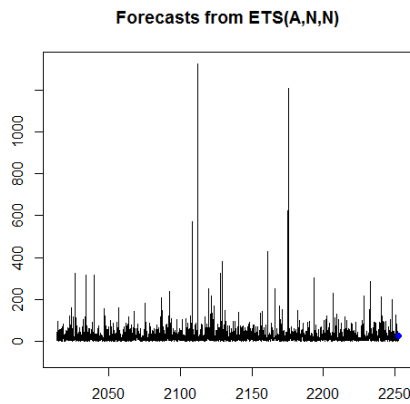


Fig. 4(d) Forecast for Low as attribute

### 5. Conclusion

In this paper, taking stock exchange as domain we have modelled our domain with bitemporal conceptual data model with time and fact as attributes. We have presented an algorithm called 2P-TO algorithm for preprocessing and sequencing of transformations which is done using informatica. The temporal databases can be preprocessed by sequencing of transformations in order to optimize the evaluation plan. The transformations can be applied through various evaluation plans. Choosing the plan which removes bottleneck and identifying the optimized plan evaluated and presented. The Indirect load transformation is found superior compared to rest of the transformation. Further, the correlation between the attributes is studied descriptively and forecasting with best attributes are done. Among all attributes, forecasting the dataset with date as predictor has yielded the best result with least error. Further, the dataset is not having major distortions, all values lie around the normal density function. Presently, the evaluation plans are interpreted manually by various possibilities. Further, the research can be extended to include unsupervised optimized plan evaluation.

## References

- [1] Faraz Rasheed, Mohammed Alshalalfa, Reda Alhadjj. Efficient Periodicity Mining in Time Series Databases using Suffix Trees. *IEEE Transactions on Knowledge and Data Engineering*. 2011; 23(1): 79 – 94.
- [2] Efthymia Tsamoura, Anastasios Gounaris, Yannis Manolopoulos. Optimization of decentralized multi-way join queries over pipelined filtering services. Springer-Verlag. 2012; 94(12): 939–972.
- [3] Magda Osman, Fei Yan, John Iilingworth, David Windridge, Josef Kittler, Teofilo decompos, William Christmas. Domain Anomaly Detection in Machine Perceptron : A System Architecture and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2014; 36(5).
- [4] Remi Emonet, Jagannadan, Varadarajan, Jean-Marc obodez. Temporal Analysis of Motif mixtures using Dirichlet processes. *IEEE transactions on pattern analysis and Machine Intelligence*. 2014; 36(1) :140-156.
- [5] Yongnian Zhang, Yifan Zhang, E. Swears. N. Larios, Ziheng wang, Qiang Ji. Modelling Temporal interactions with Interval Temporal Bayesian Networks for Complex. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2013; 35(10): 2468-2483.
- [6] Bin Liu,Elke A. Rundensteiner. Optimizing Cyclic Join View Maintenance over Distributed Data Sources. *IEEE Transactions on Knowledge and Data Engineering*. 2006; 18(3): 363-376
- [7] Yanchang Zhao, Shichao. Generalized Dimension-Reduction Framework for Recent-Biased Time Series Analysis. *IEEE Transactions on Knowledge and Data Engineering*. 2006; 18(2): 231-244
- [8] Jun-ichi Takeuchi, Kenji Yamanishi . A unifying Framework for Detecting Outliers and change Points from Time Series.*IEEE Transactions on Knowledge and Data Engineering*. 2006; 18(4): 482-492.
- [9] Carlos Ordonez. Integrating K-Means Clustering with a Relational DBMS using SQL. *IEEE Transactions on Knowledge and Data Engineering*.2006; 18(2) : 188-201.
- [10] Christian S. Jensen, Richard T. Snodgrass, Michael D. Soo. Extending existing Dependency Theory to Temporal Databases. *IEEE Transactions On Knowledge And Data Engineering*. 1996; 8(4): 563-582.
- [11] Byunk Suk Lee, Gio Wiederhold. Outer Joins and Filters for Instantiating objects from Relational Databases through views.*IEEE Transactions on Knowledge and Data Engineering*. 1994;6(1): 108-119
- [12] Morgan D. Monger, Ramon A. Mata-Toledo, Pranshu Gupta. “TEMPORAL DATA MANAGEMENT IN NOSQL DATABASES”.
- [13] Alessandro Artale and Enrico Franconi (2009), “Foundations of Temporal conceptual Data Models”, Springer Verlag, LNCS 5600, Pp. 10-35
- [14] Christian S. Jensen and Richard T. Snodgrass, Semantics of Time-Varying Information”