NoSQL- A Solution for Handling Big Data

Nidhi Srivastava

Abstract*: *The data produced nowadays is the Big data which is unstructured, semi-structured or structured in nature. It is difficult for SQL to handle such large amount of data with varied forms so NoSQL was introduced which gives many advantages over SQL. It is a schema less database which allows horizontal scaling, scalability and distributed framework. SQL is based on ACID properties (Atomicity, Consistency, Isolation, Durability) whereas NoSQL is based on BASE (Basic Availability, Soft state, Eventual consistency). This paper introduces the concepts of NoSQL, advantages of NoSQL over SQL, different types of NoSQL databases with special reference to the document store database, MongoDB. It explains in detail about the MongoDB and then experimentally evaluates the performance of the queries executed in SQL and NoSQL (MongoDB). The experiment conducted shows that NoSQL queries are executed faster as compared to SQL queries and can also handle huge amount of unstructured data very effortlessly and easily.

Keywords* : *Big Data, MongoDB, NoSQL, Relational database, SQL.

I. INTRODUCTION

SQL was used to access the structured data stored in the relational databases. With increase in the usage of the social media, data relating to this increased abundantly. This led to the storage of data in the form of documents, images, videos, pictures, audio, email, etc. This was a huge unstructured data which had requirement of large storage space. This huge data later on was termed as Big data and was characterized by three V's namely volume, velocity and variety [1, 2]. It was difficult for relational databases to handle the increase in demand of applications that require large storage for big data. In 1998, NoSQL term was first used to show databases not using SQL. This term again gained popularity in 2009 because of the demand of applications used in Web 2.0 domain [3]. The social network sites and the seller websites generate trillions of data which is huge enough for relational databases to process. The databases which handles a large amount of complex data suffers from problems like deadlocks. NoSQL also known as Not Only SQL are non-relational model [4]. In NoSQL there is no schema of the data being stored, which acts as an advantage over SQL [5]. Other reasons for preference of NoSQL over SQL are that it gives the advantages like horizontal scaling, design simplicity and availability, concurrency, fault tolerance and scalability. NOSQL can handle large amount of data known as big data which is unstructured, semi structured or structured in nature [6,7]. In this paper Section I gives the introduction, Section II describes in detail about the different types of NoSQL databases available. Section III explains and gives features of Mongo DB, a type of NoSQL document store database. Section IV shows the implementation and result of the experiment and finally Section V gives the conclusion.

II. DIFFERENT TYPES OF NOSQL DATABASES

A. Key Value Stores

In this a key which is unique in nature is assigned to single value or set of values. The data stored in the database can be accessed using this key, which is atomic and indestructible. There is no fixed set schema for the data stored in this database. In short the data is unstructured. It makes use of hash table. Unrelated data are stored in the different key value pair so foreign key and various SQL operators are not required. An advantage of this type is that any kind of new values can be added at runtime as it will not conflict with the earlier stored data. This database is particularly useful in distributed systems where the data can be managed very efficiently. Key value store is used in Facebook, Twitter, Project Voldemort and Membase. [1,2,3,8,9].

Key	Value/Attributes
1001	order_no: orderdate: qty: dely_add: salesman_name:
1002	order_no: orderdate: dely_add: salesman_name:
1003	order_no: orderdate: qty: dely_add:

Fig. 1. Key Value Database for Order table

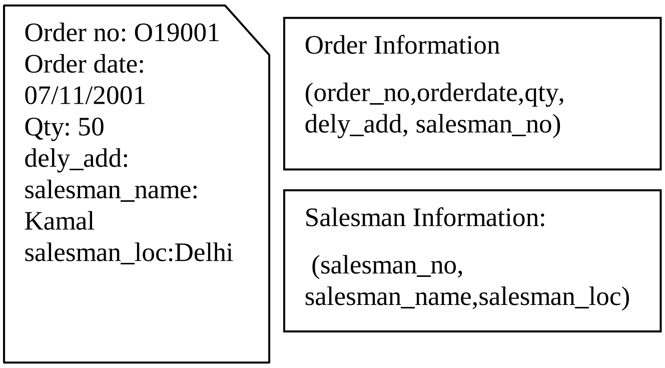
B. Document Stores

The document database stores data which is semi-structured and is in form of Java Script Object Notation (JSON), Extensible Markup Language (XML) or in some other related format. Any document with entirely new and different attributes can be stored easily and also new attributes can be added to the already prevailing documents. Each record and the data associated with it are stored in the document and this can be compared to a tuple of the relational database.

Revised Manuscript Received on October 05, 2019

Dr. Nidhi Srivastava, Asst. Professor, Amity Institute of Information Technology, Amity University, Uttar Pradesh, Lucknow Campus, Lucknow, India. Email: nsrivastava2@lko.amity.edu

This has the advantage that accessing of data becomes easy and it also decreases the usage of join operations and eases the complex queries. Varied documents are laid together into a collection which does not force constraint on any fixed document structure. The documents in document database can be indexed on the basis of primary identifier and also on the basis of its properties. Some examples of these are Mongo DB and Couch DB. [1,2,3,8,9].



Order no: O19001
Order date:
07/11/2001
Qty: 50
dely_add:
salesman_name:
Kamal
salesman_loc:Delhi

Order Information

(order_no,orderdate,qty, dely_add, salesman_no)

Salesman Information:

(salesman_no, salesman_name,salesman_loc)

Fig. 2. Document Data Store

C. Document Stores

In this the related data is stored in the form of column known as column families. There are rows which have unique row key with numerous columns in it. The number of columns in each row of the record can vary. Columns within columns can be nested in this. The outer column is known as super column. A part of the table in this can be searched as each and every entry in the column is indexed. Because of column indexing searching and look-ups of the data is fast. It is easy to maintain database stores. Some of the examples of these types of data stores are Cassandra, Hbase. [1,2,3,8,9].

Row Key	Column Families

1001	order_no: O19001	orderdate: 07/11/2001	qty: 50	dely_add: 24A Omaxe Palace Delhi	salesman_name: Kamal

1002	order_no: O19002	orderdate: 10/01/2002	dely_add: 10 babbar Road Palace Delhi	salesman_no: S00001

1003	order_no : O19003	orderdate: 25/04/2002	qty:70	dely_add: 40 Vineet Khand, Lucknow

Fig. 3. Column Data Store

D. Graph Databases

It focusses on relationships between data rather than data itself. It makes use of nodes, relationship between nodes and the properties of nodes. Graph databases make use of graph algorithms. This type of databases is used in applications where emphasis is on searching and traversing. This is a better way of searching data than relational database as it is quicker in searching data according to the shortest path algorithm. These databases can be scaled horizontally. Examples of these types of data are Neo4G and InfoGrid. [1,2,3,8,9].

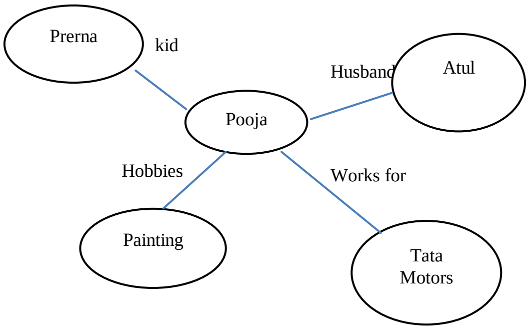


Fig. 4. Graph Database

III. MONGODB

It is a schema less document store database written in C++. It uses JSON like documents. It is flexible as different fields can be used in different documents and also the different data structure can be used. It provides advantages like better performance, adaptability in business, flexibility and agility. It is a cluster of nodes. Since it is a type of document store database, the documents are stored in form of collections. A table in relational databases parallels to a collection in Mongo DB, a single document is same as row and a field is mapped to a column [10,11,12]

Features of MongoDB:

Some of the features of MongoDB are as follows [11,13,14]

a. For read and write operations it has a rich query language. It provides features like sorting, secondary indexes, etc.

b. Performance is very high as each and every attribute in the document is indexed and this leads to faster queries.

c. It is open access software easily available and can very conveniently be installed, configured and maintained.

d. It is easily scalable.

e. Sharding is one of the most important features of MongoDB. It provides horizontal scaling and thus can handle the unexpected load easily by increasing the number of machines.

f. It supports multiple storage engines. In fact Mongo DB uses Wired Tiger storage engine which also has multiple storage engine.

Retrieval Number: A9896109119/2019©BEIESP
DOI: 10.35940/ijeat.A9896.109119

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

IV. EXPERIMENTAL RESULT & EVALUATION

The terms used and the way query are handled in SQL and NoSQL are very different. In NoSQl instead of tables, collections are created and rows or columns are replaced by key-value pairs and documents [15,16,17,18]. We have conducted the experiment to compare various queries as used in SQL and MongoDB. Same data is being inserted and retrieved in both the cases to show the difference. The queries for SQL are run on Oracle 10G. MongoDB queries are run on Mongo 4.2. The various SQL queries and its corresponding queries in Mongo DB are shown below:

1. **SQL QUERIES:**

a) Create Query:

Create table sales_order (order_no varchar(8),orderdate date, qty int, dely_add char(50), salesman_no varchar(8))

b) Insertion Query:

Insert into sales_order (order_no, orderdate, qty, dely_add,salesman_no) values (‘O19002’,’12-Jun-08”, 2, ‘24A Omaxe Palace Delhi’,’S00001’)

c) Select Query:

Select * from sales_order where order_no=’O19001’

d) Delete Query:

Delete from sales_order where order_no=’O19001’

e) Drop Query:

Drop table sales_order;

2. **MONGO DB Queries:**

a) Create Query:

No special command. Insert command is used for creation and insertion of records.

b) Insertion Query:

db.sales_order.insert({order_no:“O19002”, orderdate:“12-June-08”, qty:2, dely_add:“ 24A Omaxe Palace Delhi”, salesman_no:“ S00001”})

c) Select Query:

db.sales_order.find({ order_no:”O19001”})

d) Delete Query:

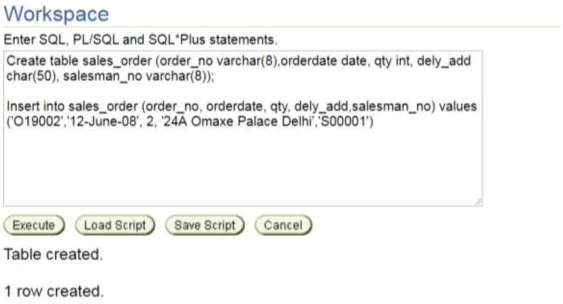
db.sales_order.remove({order_no:”O19001”})

e) Drop Query:

db.sales_order.drop();

The figure 5 to figure 7 show the queries when run on Oracle. Fig. 5 shows the command for creating table and then inserting data into the created table. Fig. 6 shows query to retrieve a particular record from the table. Finally Fig. 7 shows query to delete a particular record and also drop the entire table.

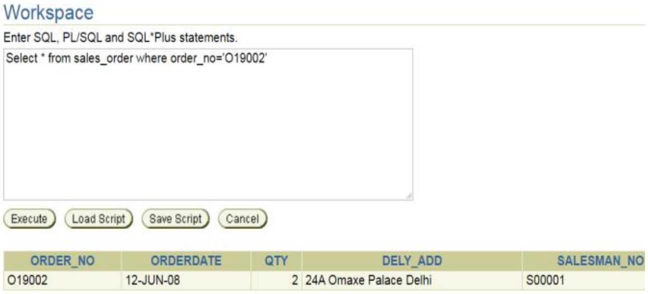
Figures 8 to figure 10 show the queries when run on MongoDB. In MongoDB separate command need not be utilized for creating the collection. When using insert command, collection is automatically created. Fig. 8 gives the command to create and insert the data when id is not given by the user. In this case an id is given by the system as shown in fig. 8 where find command is used. Fig. 9 shows the insert command when the id is supplied by the user, and the id is visible in fig. 9 where find command is used. Fig. 10 finally shows how to delete a particular record and drop a table.



```
Create table sales_order (order_no varchar(8),orderdate date, qty int, dely_add
char(50), salesman_no varchar(8));

Insert into sales_order (order_no, orderdate, qty, dely_add,salesman_no) values
('O19002','12-June-08', 2, '24A Omaxe Palace Delhi','S00001')
```

```
Table created.

1 row created.
```

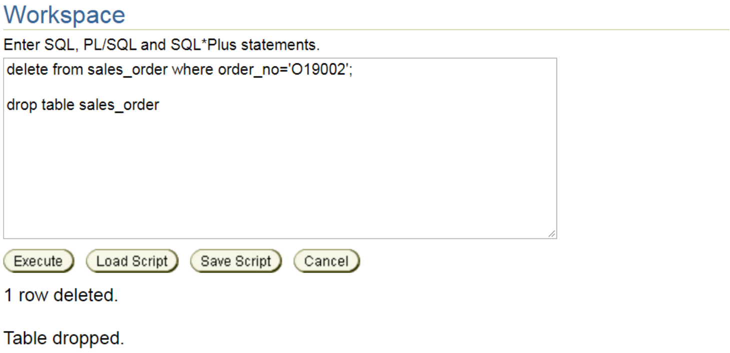
Fig. 5. Create table command and query for insertion of data



```
Select * from sales_order where order_no='O19002'
```

ORDER_NO	ORDERDATE	QTY	DELY_ADD	SALESMAN_NO
O19002	12-JUN-08	2	24A Omaxe Palace Delhi	S00001

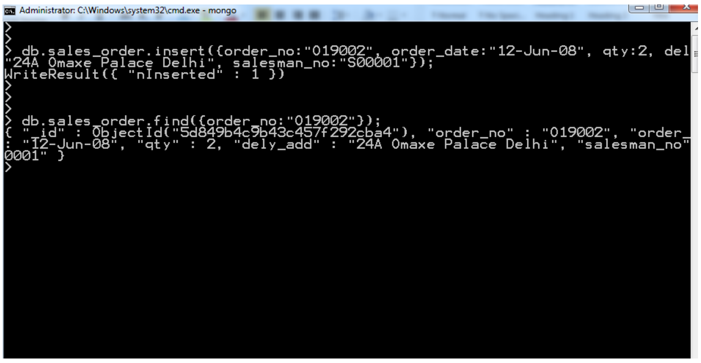
Fig. 6. Retrieving a particular record



```
delete from sales_order where order_no='O19002';

drop table sales_order
```

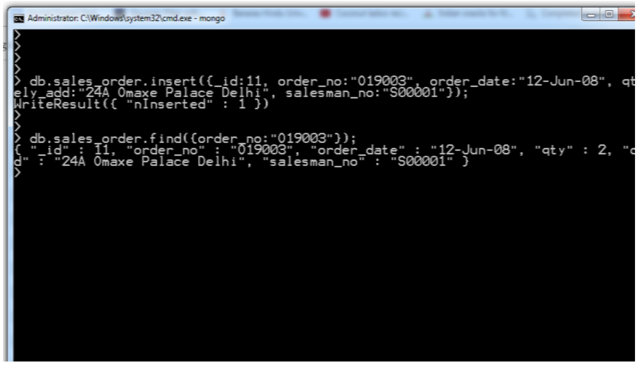
```
1 row deleted.

Table dropped.
```

Fig.. 7. Deleting a particular record and dropping the table



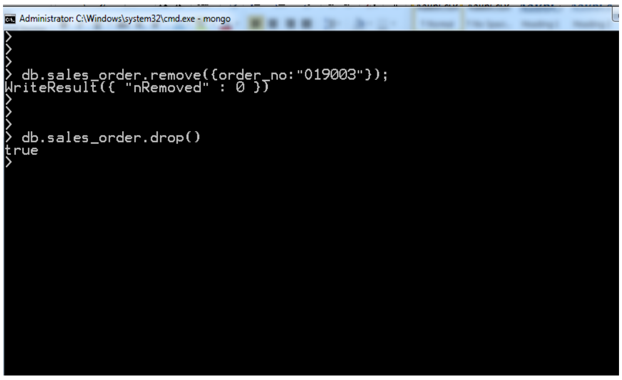
```
>
>
> db.sales_order.insert({order_no:"O19002", order_date:"12-Jun-08", qty:2, del
"24A Omaxe Palace Delhi", salesman_no:"S00001"});
WriteResult({ "nInserted" : 1 })
>
>
>
> db.sales_order.find({order_no:"O19002"});
{ "_id" : ObjectId("5d849b4c9b43c457f292cba4"), "order_no" : "O19002", "order_
: "12-Jun-08", "qty" : 2, "dely_add" : "24A Omaxe Palace Delhi", "salesman_no"
0001" }
>
```

Fig. 8. Insert query when id not supplied by user & query for retrieving particular data



```
Administrator: C:\Windows\system32\cmd.exe - mongo
>
>
>
>
> db.sales_order.insert({_id:11, order_no:"019003", order_date:"12-Jun-08", qt
ely_add:"24A Omaxe Palace Delhi", salesman_no:"S00001"});
WriteResult({ "nInserted" : 1 })
>
>
> db.sales_order.find({order_no:"019003"});
{ "_id" : 11, "order_no" : "019003", "order_date" : "12-Jun-08", "qty" : 2, "c
d" : "24A Omaxe Palace Delhi", "salesman_no" : "S00001" }
>
```

Fig. 9. Insert query when id is supplied by user



```
Administrator: C:\Windows\system32\cmd.exe - mongo
>
>
>
> db.sales_order.remove({order_no:"019003"});
WriteResult({ "nRemoved" : 0 })
>
>
>
> db.sales_order.drop()
true
>
```

Fig.10. Deleting a particular data and dropping the table.

Further in this experiment many data in bulk was inserted into and deleted from the table in SQL and collections in MongoDB and the performance is shown below in the figure 11.

TABLE 1 –Time taken in inserting data (sec)

No. of Data Inserted	SQL (secs)	NoSQL (MongoDB) (secs)
50	0.015	0.17
100	0.026	0.23
500	0.96	0.57
1000	1.48	1.0
5000	2.95	1.58
10000	8.54	3.68
50000	13.11	7.14

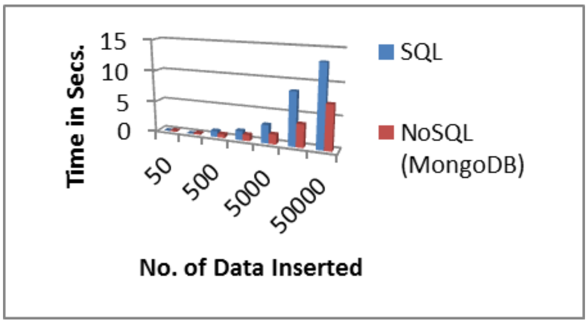


Fig. 11: Time taken to insert data.

Table 1 show the time taken to insert data in SQL and MongoDB and Fig. 11 shows the same in graph. It is clear from the result that the time taken to insert data in MongoDB is much less as compared to that of SQL. Similarly, while deleting data the time taken to delete data in MongoDB is almost same irrespective of the number of records deleted whereas, in SQL it is more as shown in table 2 and depicted in Fig. 12. The result shows that the insert and delete queries performed on large number of records in MongoDB takes much less time than the ones in SQL. This clearly shows that NoSQL is capable of handling huge data easily and also takes much less time in executing the queries relating to it.

TABLE 2 –Time taken in deleting data (sec)

No. of Data Deleted	SQL (secs)	NoSQL (MongoDB) (secs)
50	0.045	0.02
100	0.062	0.02
500	0.051	0.03
1000	0.085	0.03
5000	1.65	0.03
10000	1.78	0.03
50000	3.56	0.03

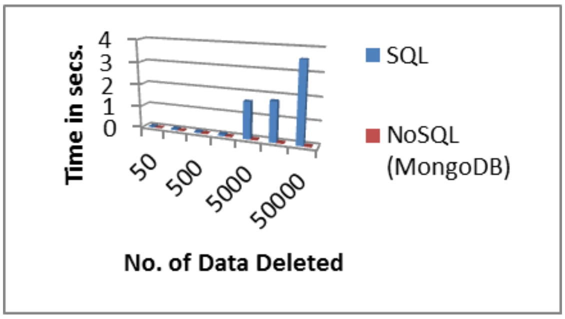


Fig. 12: Time taken to delete data

V. CONCLUSION

NoSQL databases have become quite popular nowadays as no strict schema is followed which helps in arranging the data in any form and reduces the needs for joins and foreign keys. Since NoSQL are open source software it is easily available. MongoDB is a document store database which stores data in collections in which the size can be easily changed. In this paper MongoDB as NoSQL is used to show a comparison between the SQL and NoSQL databases. The performance as seen from the experimental evaluation clearly shows that MongoDB is faster as compared to the SQL when large set of data, big data, is taken into consideration. Although the security of the data is a concern in NoSQL but with time new techniques are being developed through which this disadvantage will be overcome.

REFERENCES

1. "Handling Big Data using NoSQL" , Jagdev Bhogal; Imran Choksi, 29th International Conference on Advanced Information Networking and Applications Workshops, pp. 393-398, IEEE, 2015

2. “Analysis of Various NoSql Database”, Pragati Prakash Srivastava; Saumya Goyal; Anil Kumar, International Conference on Green Computing and Internet of Things (ICGCIoT), pp. 539- 544, 2015
3. “NoSQL Evaluation A Use Case Oriented Survey”, Robin Hecht, Stefan Jablonski, International Conference on Cloud and Service Computing, pp. 336-341, IEEE, 2011
4. “Comparison of SQL, NoSQL and NewSQL Databases for Internet of Things”, Haleemunnisa Fatima, Kumud Wasnik, IEEE Bombay Section Symposium (IBSS), IEEE 2016
5. “Capacity Measurement and Planning for NoSQL Databases”, Ramya Pasumarti, Rushin Barot, Susie Xia, Ang Xu, Haricharan Ramachandra, 11th International Conference on Semantic Computing, pp 390- 394, IEEE 2017
6. “Exploring the Merits of NoSQL: A Study Based on MongoDB”, Benymol Jose, Sajimon Abraham, International Conference on Networks & Advances in Computational Technologies (NetACT), 2017
7. “Choosing the right NoSQL database for the job: a quality attribute evaluation”, João Ricardo Lourenço , Bruno Cabral, Paulo Carreiro, Marco Vieir, Jorge Bernardino, Lourenço et al. Journal of Big Data (2015) 2:18
8. “NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison”, A B M Moniruzzaman, Syed Akhter Hossain, International Journal of Database Theory and Application Vol. 6, No. 4. Pp. 2013
9. “Persisting Big Data: The NoSQL Landscape”, Alejandro Corbellinia, Cristian Mateosa, Alejandro Zuninoa, Daniela Godoya, Silvia Schiaffinoa, Information Systems, Volume 63, pp. 1-23, January 2017.
10. “Application of NoSQL Database MongoDB”, Yunhua Gu, Shu Shen, Jin Wang, Jeong-Uk Kim, International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp. 158-159, IEEE 2015
11. “MongoDB and NoSQL Databases”,Vidushi Jain, Aviral Upadhyay International Journal of Computer Applications, Volume 167 – No.10, pp. 16-20, June 2017
12. “MongoDB NoSQL Injection Analysis and Detection”, Boyu Hou, Kai Qian, Lei Li, Yong Shi, Lixin Tao, Jigang Liu, IEEE 3rd International Conference on Cyber Security and Cloud Computing, 2016.
13. “Security Analysis Of Unstructured Data In Nosql Mongodb Database”, Jitender Kumar, Varsha Garg, International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), 2017
14. “Review of NoSQL Databases and Performance Testing on HBase”,WUMUTI• NAHEMAN, JIANXINWEI International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Dec 20-22, 2013, Shenyang, China, 2013
15. The Selection of NoSQL Database in E-government System of China Meteorological Administration Shuai Wang, Ying Zhu, Fudi Wang, Zifeng Qian, IEEE 2nd International Conference on Big Data Analysis, 2017
16. “Knowledge-Based Fast Web Query Engine using NoSQL”, Prof. Dr. Peter A. Cooper, Blimat Ahmed Omar, 6th International Symposium on Digital Forensic and Security, IEEE2018
17. “A Comparative Study of Relational and Non-Relational Database Models in a Web- Based Application”, Cornelia Győrödi, Robert Gyorodi, Roxana Sotoc International Journal of Advanced Computer Science and Applications • Vol. 6, No. 11, November 2015
18. “MongoDB vs Oracle - database comparison”, Alexandru Boicea, Florin Radulescu, Laura Ioana Agapin, Third International Conference on Emerging Intelligent Data and Web Technologies, IEEE 2012

AUTHORS PROFILE

Dr. Nidhi Srivastava is currently working as Assistant professor in Amity University. Her research area is Human Computer Interaction, e-governance and Cloud Computing. She has about 14 years of teaching experience. She has published many research papers in various reputed national and international journals and conferences. She is in the editorial and reviewer board of many prestigious journals and conferences.