ASEAN Engineering Journal

Full Paper

SUSPICIOUS TRANSACTION IDENTIFICATION AND RECOVERY IN CASSANDRA DATABASE

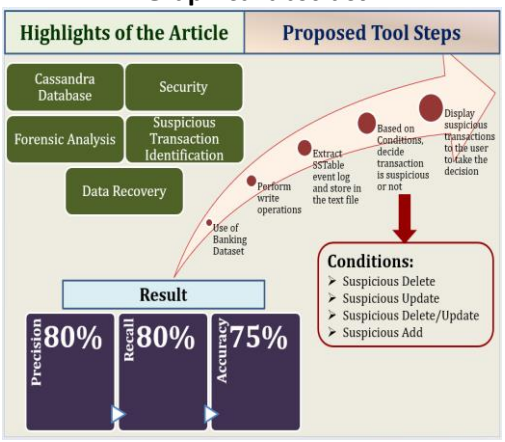
Rupali Chopade[a]*, Vinod Pachghare[b], Damini Sheth[b], Nikhil Dhavase[c], Yogita Sinkar[d]

[a]Department of Computer Science & Engineering, DES Pune University, Pune, India
[b]Department of Computer Engineering, COEP Technological University, Pune, India
[c]Department of Information Technology, Marathwada Mitramandal's College of Engineering, Pune, India
[d]Department of Computer Engineering, SVPM College Of Engineering, Malegaon (Bk), India

Article history
Received
17 April 2024
Received in revised form
16 July 2024
Accepted
31 July 2024
Published online
28 February 2025

*Corresponding author
rupali.chopade@despu.edu.in

Graphical abstract



Abstract

Forensic analysis of databases is a challenging and important research field in digital forensics. Most of the applications use databases to store the data. Cassandra is a NoSQL database that offers data replication for high availability, fault-tolerance and ensures no single point of failure. Given its growing popularity, financial institutions have begun to consider Cassandra as a potentially useful database for their organization. Considering the abundant amount of fraud and its implications that can occur at financial institutions, it is needed to ensure that no suspicious transaction on the Cassandra database goes unnoticed by the organization. In addition, being able to recover lost data due to malicious activities is equally necessary. This article presents a tool which helps in identifying suspicious transactions in a financial institution and an option to recover that data.

Keywords: Cassandra, NoSQL, Data Recovery, Suspicious Transaction, Database Forensics.

1.0 INTRODUCTION

Database security and database forensics are important and challenging research fields. Security of data is an important concern to every organization. Data security is essential in every sector of industry like medical [1], banking, education, e-commerce etc. Relative to relational databases, NoSQL databases are more recent and hence, less researched and worked [2]. Even among the different NoSQL databases such as Cassandra, MongoDB and Redis, comparatively less research has gone into Cassandra. Table 1 shows the comparative analysis of MongoDB, Redis and Cassandra, in terms of data model, overall ranking of database, existing research work related to data recovery and security features available.

Table 1 NoSQL Database Comparison

	MongoDB	Redis	Cassandra
Data Model	Document	Key-Value	Column
Overall Rank[3]	5	6	12
Data Recovery Research	Internal Structure[4]	Internal Data Dictionary[5]	Not any Specific
Security	Queryable Encryption, Client Side Field Level Encryption, TLS/SSL, x509, Server Side storage engine encryption, LDAP and Kerberos connectors, and built-in SCRAM or certificates for authentication and authorization	a layer that implements ACLs, verifies user input, and determines what actions to take against the Redis instance	Support for client connections via TLS/SSL User authorizations with roles and user authentication

This existent research gap and the need to make it convenient and easier for the user to recover lost data has influenced our choice of development work. The research challenges associated with Cassandra database forensics are depicted in Figure 1.

Forensic challenges of Cassandra Database	Data Distribution and Replication
	Event Reconstruction
	Consistancy Model
	Lack of Centralized Metadata
	Access Controls and Audit Logging
	Volume and Scale
	Complex Data Model

Figure 1 Cassandra Database Forensic Challenges

Cassandra's data structure is the column family store which contains key/value pair. Individual columns are combined in a row and identified by a partition key [6]. Row consists of one or more columns and the primary key. The data partitioning concept in Cassandra is based on its partition key and it is passed through a hash function. Within the cluster, the same partition key data will be stored on the same node. The read and write operations in the Cassandra database are shown in Figure 2 [7].

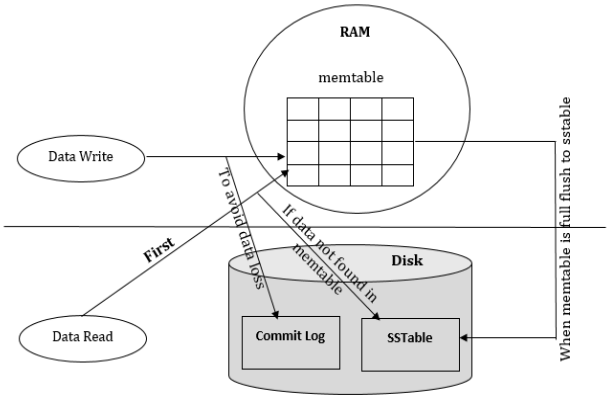


Figure 2 Read and write in Cassandra

Whenever a write operation is executed, it will get stored in memtable- the memory resident table and it will also get stored in the commitLog which is available on disk. When memtable entries are full data will be flushed from memtable to SSTable (Sorted Strings Table). SSTable is available on disk. During read operation data will be searched in memtable first and if it is not available then it will be checked in SSTable. Considering that Cassandra is a very recent technology, and because of its distributed nature; forensic analysis algorithms and tools are still going through the research and development phase [8, 9]. In this scenario, we have performed forensic analysis for Cassandra's distributed system architecture and implemented a tool. It uses the inherent features of Cassandra's delete and update operations. The developed tool introduces the notion of identifying all possible tampering of delete and update transactions and provides a complete transaction candidate set for review.

1.1 Organization of the Paper

The rest of the article is organized as follows. In section 2 we describe research related to forensic analysis of Cassandra and other databases. In section 3 we move on to present the preliminary. The critical assessment of the problem has been explained in section 4. It includes the delete process, review of missing data in Cassandra and its causes and the SSTable storage format. In section 5, the proposed tool and its implementation details are presented. Section 6 analyzes the outcomes of the testing process carried out on the application developed. Several criteria for measurement of accuracy of the developed tool and the evaluation of its performance are used. Section 7 comprises the conclusions drawn from the performed analysis and future work.

2.0 RELATED RESEARCH

In recent times, Cassandra is gaining popularity due to its scalable and availability nature. Given that it's a NoSQL database, it is evident that Cassandra lacks the amount of research as compared to relational and other NoSQL databases [10, 11]. Most of the work and papers reviewed within this domain dated as late as 2017 and 2018, which further substantiates the evident

recentness of the selected research work area. Works related to research on Cassandra and its forensic analysis can be broadly categorized under the themes of challenges in forensic attribution in NoSQL databases, security vulnerabilities of NoSQL databases, implications of forensic attribution, methods and processes used to determine forensic readiness and automation of forensics processes.

2.1. Forensic Examination Challenges of NoSQL Databases

Hauger et al. cited 'attribution' to be an important motivating factor behind performing database forensics [12]. They have presented database triggers and their impact on digital forensics analysis. If a trigger is tested positive by the algorithm mentioned in the article, its actions are forensically examined. It mentions two phases in the forensic examination process where the 1st phase deals with forensic acquisition and analysis of databases and the 2nd phase interprets data for reconstruction and attribution. An implementation challenge observed is the presence of false positive errors which required one to manually inspect each trigger. It further studied the logging features and access control in NoSQL databases. An important finding is that NoSQL databases either inherently lack logging capabilities and access control or did not enable them by default. The logging feature in Cassandra is default, which supports but not the access control feature. Out of the four NoSQL databases studied, only Cassandra provided native functionality for triggering and that too, only DML triggers. Thus, it was concluded that forensic analysis of any database, including Cassandra heavily depends on the configuration performed on the database by the database administrator.

2.2 Security vulnerabilities in SQL and NoSQL databases

The article by Shahriar and Haddad [13] primarily deals with the threats that NoSQL technologies pose to users (particularly learners of Massive Open Online Courses). This article has initially provided a comparison between SQL and NoSQL databases and an overview of MongoDB and Cassandra. Additionally, it goes on to give details of considerations regarding NoSQL security, prevalent security issues in Cassandra and MongoDB and some examples of attacks. The major possible attacks in Cassandra are: CQL injection, DoS and XSS. While Auditing of Cassandra is better, MongoDB provides better Authorization.

2.3 Forensic-attribution-based works and its implications on NoSQL databases

Hauger and Olivier [14] highlighted the need of performing accurate forensic attribution on NoSQL databases as they have become a new target for hackers. In recent times owing to the fact that NoSQL databases such as Cassandra are gaining popularity among organizations to store sensitive information. The article surveyed the top five NoSQL databases namely MongoDB, Cassandra, Redis, HBase and Neo4j to analyse the extent to which they can provide authentication and authorisation features to their user organizations. The traces left by security features of these databases were used for this purpose. In Cassandra, authentication is pluggable and needs to be configured by changing the settings in the configuration file. By default, Cassandra performs no authentication and no credentials are entered because the configuration uses AllowAllAuthenticator by default. Cassandra, however, does allow an alternative PasswordAuthenticator which uses a system table to store credentials of users in an encrypted format. As far as access control is concerned, Cassandra doesn't provide pre-defined fixed roles to its users. Just like authentication, an authorization in Cassandra is also pluggable and needs to be configured in the configuration file. By default, Cassandra creates two log files namely the system log file and the debug log file. A separate commit log file also exists. The fact that these log files are generated in Cassandra indicates that Cassandra has an in-built logging facility by default along with the features of authorization and authentication. However, access control in Cassandra is not enabled by default. It also mentions the potential threat that could occur if the log files are tampered [15-17].

2.4 Forensic-readiness-based works

Rowlingson et al. has proposed a ten-step process [18] to enable organizations to collect and use credible digital evidence to its fullest potential. The purpose is to minimize the costs associated with carrying out an investigation in response to the occurrence of a suspicious incident. It underlines the importance of being forensically ready and prepared at all times. Log files can be one of the potential pieces of evidence that can be collected beforehand and greatly benefit the organization. Authors also sheds light on the possibility of a great majority of crime or tampering being an insider's job. The goal to be achieved by being forensically ready is to reduce the time and cost of forensic examination. The steps mentioned covers the areas of defining business scenarios, identifying available resources, determining requirements, establishing capability to gather evidence, establishing policy to securely store evidence, monitoring to detect major incidents, specifying circumstances for launching full-fledged investigation, training staff to increase awareness, documenting a case to describe incident and its impact, ensuring legal review.

2.5 Forensic-automation-based works

Today's computerized and networked environment necessitates the collection of a large volume of evidence for forensic analysis and investigation. Hence, the advancement of automation in the analysis phase of forensic digital investigations is inevitable [19]. An advantage of the increased automation would be reduction in time and efforts. In the initial stages of work, the focus is on automating the acquisition and identification of evidence. Later on the focus is shifted to full automation of all phases of the digital investigation process. A main motivating factor behind this automation is to alleviate the burden on human analysts. This article also analyzes the feasibility of remote evidence acquisition. The main challenges affecting the speedy identification, collection and analysis of evidence were identified to be predominantly related to volume, velocity, heterogeneity, manual or quasi-automated procedures, challenges related to work-force and shortcomings of the current available legal framework. The solutions proposed so far include continuous revision of legal

framework, training of personnel, and prioritization of evidence sources. These solutions are broadly related to the three domains namely legal solutions, human resource solutions and technological solutions. The contributions made by the authors include development of methods to automate integration and searching of evidence from many heterogeneous sources. The article cites three important characteristics of gathered digital evidence which are latency, fidelity and volatility. A goal of the study is to ascertain that high-level queries of analysts are fed into the system and evidence knowledge is extracted via automated reasoning.

3.0 PRELIMINARY

3.1 User Classes And Characteristics

Admin User: This user will have access to data regarding suspicious delete or update transactions.
Non-admin User: This user will decide which of the transactions are genuinely performed by an opponent and then data recovery of that data will take place. A non-admin user belongs to the user organization or is an individual user who wished to avail the facility of data recovery offered by the tool.

3.2 Operating Environment

The proposed tool can be accessed using secured internet connection using a web browser by an organization. This application for forensic analysis is to be operated in an environment having Cassandra database on its systems.

3.3 Design and Implementation Constraints

The tool can only recover data which has been deleted or gone missing due to transactions which have been marked as suspicious.

3.4 Assumptions and Dependencies

- The application assumes that the data received regarding suspicious transactions is accurate.
- It depends on the accuracy of fetched data regarding suspicious delete or update transactions.
- The system assumes that the user of the application has sufficient knowledge to know which transaction is by an opponent and which transaction is valid.

3.5 Domain selection for forensic analysis tool development

For this research work, focus is on Financial Institutions. The financial Institutions like Banks, Insurance Companies, etc. are suffering from an increase in fraud incidents in India and abroad, thereby directly affecting their bottom-line. Need for robust forensic analysis by Financial Institutions is therefore not an option but the need of the hour. Financial Institutions generate tremendous amounts of data. However, a large amount of money is lost every year due to fraud or other malpractices such as information deletion, or unauthorized updates, etc. The evidence is mostly found in different digital media, in the form of active, deleted, hidden, lost or logs etc. The key to analyzing it is turning the data into meaningful information.

- Understand applicable requirements and policies
- Collect required data
- Analyze the data with tool developed for Forensics Analysis
- Present the evidence in an understandable manner

3.6 Fields used in the domain of Financial Institutions

The fields used in the dataset are shown in Table 2. The dataset has been prepared for financial institutions application.

Table 2. Fields Used in Application

Field Name	Description
Transaction ID	Every transaction is identified by its unique ID
Customer ID	Unique identification number for customer
Operation Type	Type of financial transaction performed
Transaction Amount	The amount associated in transaction
Monthly Income	Monthly income of customer
Loan Defaulter Flag	Repayment of loan amount (True / False)
Timestamp	Timestamp associated with transaction performed

4.0 CRITICAL ASSESSMENT OF PROBLEM

To critically assess the problem of recovering data and using it for forensic analysis, the focus is on following two aspects:
i. Data recovery of deleted data [20, 21]
ii. Data recovery of missing data during updates

4.1 Delete Process in Cassandra

Cassandra supports the 'DROP KEYSPACE' and 'DROP TABLE' command for data deletion, which is immediate. Along with these, there are other two methods of deletion in which deletion takes place with some delay.
These methods are:
i. User issues a delete command
ii. User marks record (row/column) with Time To Live (TTL)

In the first method, when the delete command is issued, a tombstone, a deletion marker (that marks the record to be deleted), gets added to that particular record. Then this tombstone is written to SSTable [22]. Tombstone is associated with a grace period which is a period expressed in gc_grace_seconds that gets over then tombstone gets deleted by compaction. The default value for grace period is 864,000 seconds (ten days). However, each table can decide its own value for the grace period. In the second method, the user marks the row/column with TTL value. Setting TTL value is applicable for insert and update operations. When TTL value expires, that particular record is marked with a tombstone and then this tombstone is written to SSTable. Now when the grace period of tombstone denoted in gc_grace_seconds expires then a

tombstone gets deleted by compaction. This process is explained in Figure 3.

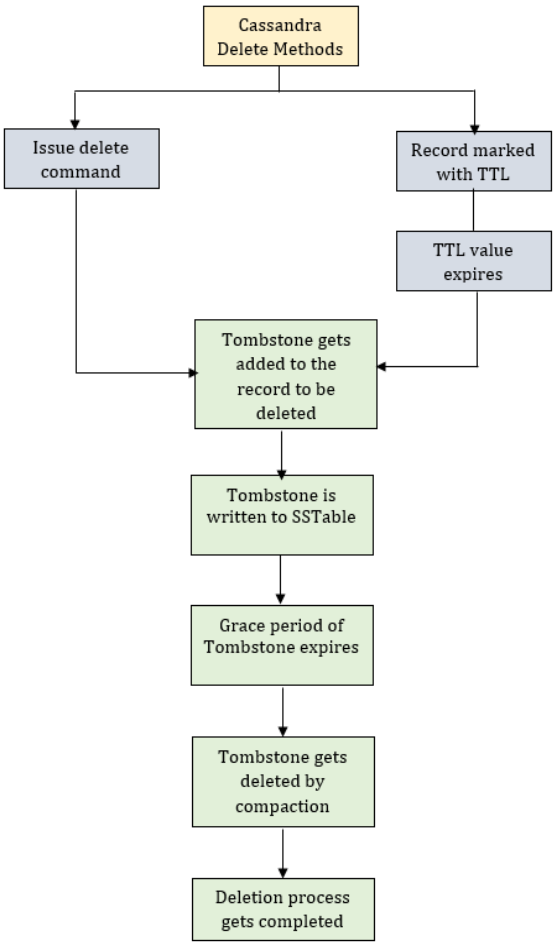


Figure 3 Delete process in Cassandra

4.2 Missing Data in Cassandra and it's Causes

The new updated data cannot be observed by the user in the following two cases [22]. The data could either have failed to get updated i.e. the user would see old data. The data could appear to have gone missing i.e. the user sees no data. In the 1st scenario where the user sees old data instead of new updated data, the main cause can be identified to be as value in 'writetime' during update operation being ahead of the actual time. This is a system error. In this first scenario, the application takes care of data recovery by checking the SSTable for the 'writetime' of the update operation. In the second scenario, the user sees no data after an update operation is performed i.e. user sees neither the old value nor the new updated value. The main cause for the 2nd scenario is that the record that user was trying to update had a tombstone for it and hence, got deleted by compaction after the grace period associated with tombstone expired. The 2nd scenario can be solved by checking SSTable for tombstones before any update operation and not updating any record which has a tombstone for it. The missing data scenarios have been explained in Figure 4.

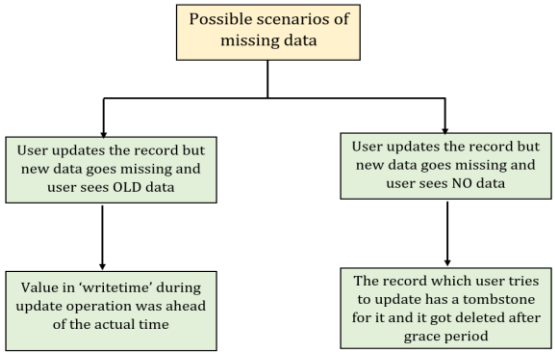


Figure 4 Missing Data Scenarios in Cassandra and Its Causes

4.3 Cassandra SSTable Storage Format

SSTable stands for Sorted Strings Table. SSTables are the immutable data files that Cassandra uses for persisting data on disk. As shown in Figure 2 whenever memtable data has been flushed to disk then Cassandra creates a new SSTable. SSTable files of a column family are stored in its respective column family directory. The compaction allows you to combine multiple SSTables into one. Once the new SSTable has been written, the old SSTables can be removed. Each SSTable is composed of multiple components [23] that are stored in separate files [24]. The SSTable event long dump of deleted records is as shown in Figure 5.

SSTable Event Log Dump of Deleted Record


```
sstable (eventlog_dump_2019oct26)
[
 {
  "partition" : {
   "key" : [ "d8ad1f80-f7b0-11e9-a063-c755130e66ac" ],
   "position" : 60
  },
  "rows" : [
   {
    "type" : "row",
    "position" : 125,
    "liveness_info" : { "tstamp" : "2019-10-26T05:24:13.302391Z" },
    "cells" : [
     { "name" : "description", "value" : "IT" },
     { "name" : "name", "value" : "Information Technology" }
    ]
   }
  ]
 },
 {
  "partition" : {
   "key" : [ "c9093780-f7b0-11e9-a063-c755130e66ac" ],
   "position" : 126,
   "deletion_info" : { "marked_deleted" : "2019-10-26T05:24:56.966557Z", "local_delete_time" : "2019-10-26T05:24:56Z" }
  },
  "rows" : [ ]
 }
]
```


Figure 5 SSTable Event Log Dump

5.0 THE PROPOSED TOOL

The workflow of the process to be followed while implementing the application tool for forensic analysis is shown in Figure 6.

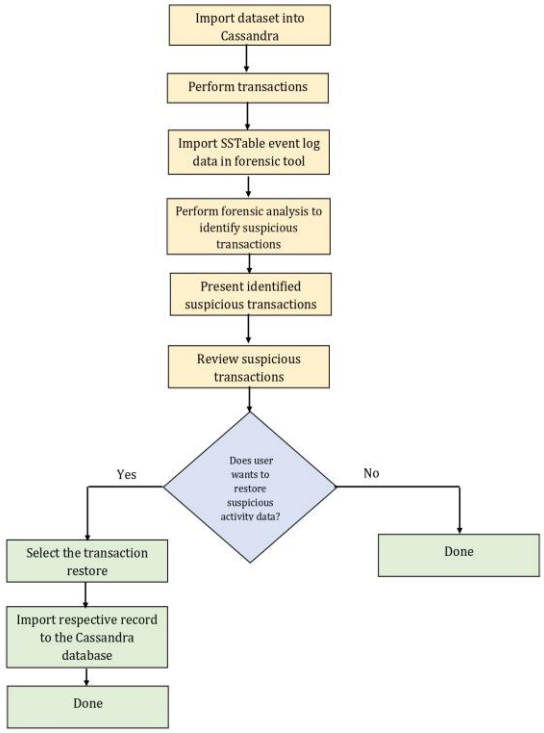


Figure 6 Process Workflow

The development of the forensic analysis tool occurred incrementally and step-wise. These steps can broadly be classified and included into several phases as below:

Step 1: The sample dataset prepared manually related to baking transactions and is imported in the database. The dataset snapshot is shown in Figure 7.

cust_trans_id	customer_id	type_of_operation	amount	salary_or_income	is_loan_defaulter	tans_date_time
149df030-2d50-11ea-8202-4ffd11dfcd72	customer030	Loan	984440	120000	FALSE	4/11/2019 15:30
15274e00-2d5c-11ea-8202-4ffd11dfcd72	customer013	Loan	2000000	209600	FALSE	16-11-2019 08:35
181bf260-2d5b-11ea-8202-4ffd11dfcd72	customer003	FD	70000	14000	FALSE	7/11/2019 20:45
23806680-2d5c-11ea-8202-4ffd11dfcd72	customer028	FD	400000	48800	FALSE	16-11-2019 13:35
25ba9f10-2d5c-11ea-8202-4ffd11dfcd72	customer006	FD	250000	45700	FALSE	16-11-2019 13:35
30435a30-2d52-11ea-8202-4ffd11dfcd72	customer024	FD	450000	80000	FALSE	7/11/2019 15:45
325b7cd0-2df7-11ea-992a-87dc29801792	customer029	FD	200000	200000	FALSE	19-11-2019 10:35
37ef3980-2d5b-11ea-8202-4ffd11dfcd72	customer019	FD	92000	400000	FALSE	8/11/2019 15:45
3fc88b10-2d5c-11ea-8202-4ffd11dfcd72	customer016	Loan	700000	74800	FALSE	17-11-2019 11:47
49fa6810-2d5c-11ea-8202-4ffd11dfcd72	customer027	FD	25000	75700	FALSE	17-11-2019 14:47
4a847510-2d5b-11ea-8202-4ffd11dfcd72	customer002	Loan	902100	25000	FALSE	10/11/2019 15:45
56095ef0-2d5b-11ea-8202-4ffd11dfcd72	customer020	Loan	902100	20000	FALSE	10/11/2019 15:45
57709aa0-2d5c-11ea-8202-4ffd11dfcd72	customer008	FD	200000	74800	FALSE	18-11-2019 15:47
5bc3c920-2d51-11ea-8202-4ffd11dfcd72	customer031	FD	620010	80000	FALSE	6/11/2019 15:05
64341ff00-2d5c-11ea-8202-4ffd11dfcd72	customer010	FD	500000	135000	FALSE	18-11-2019 15:17
674a6b40-2d52-11ea-8202-4ffd11dfcd72	customer005	Loan	821000	76000	FALSE	8/11/2019 16:45
6dba4820-2d5b-11ea-8202-4ffd11dfcd72	customer026	FD	68000	56000	FALSE	11/11/2019 15:41
7c53a980-2d51-11ea-8202-4ffd11dfcd72	customer022	FD	210100	40000	FALSE	6/11/2019 8:20
818a5930-2d51-11ea-8202-4ffd11dfcd72	customer015	Loan	300000	90000	TRUE	1/11/2019 15:15
8467cb10-2d5b-11ea-8202-4ffd11dfcd72	customer001	FD	5000	5200	FALSE	12/11/2019 15:42
972ffd80-2d5b-11ea-8202-4ffd11dfcd72	customer033	Loan	8000	15200	TRUE	12/11/2019 15:35
9b20c0b0-2d50-11ea-8202-4ffd11dfcd72	customer025	Loan	120000	83000	FALSE	4/11/2019 12:35
a417acb0-2d50-11ea-8202-4ffd11dfcd72	customer007	Loan	70000	36000	TRUE	5/11/2019 15:05
a7d32d10-2d5b-11ea-8202-4ffd11dfcd72	customer009	Loan	81000	22200	TRUE	14-11-2019 10:35
b550ac40-2d53-11ea-8202-4ffd11dfcd72	customer023	FD	20100	40000	FALSE	7/11/2019 8:45
b9e6dc90-2d5b-11ea-8202-4ffd11dfcd72	customer012	FD	210100	200000	FALSE	14-11-2019 10:35
c586ef90-2d5b-11ea-8202-4ffd11dfcd72	customer017	FD	22000	36500	FALSE	15-11-2019 11:35
d426d510-2d5b-11ea-8202-4ffd11dfcd72	customer018	FD	100000	100000	FALSE	16-11-2019 15:35
dfceb040-2d5b-11ea-8202-4ffd11dfcd72	customer010	FD	300000	135000	FALSE	16-11-2019 07:35
e3413830-2d4f-11ea-8202-4ffd11dfcd72	customer032	Loan	690000	65000	FALSE	2/11/2019 15:25
e99517b0-2d4f-11ea-8202-4ffd11dfcd72	customer004	FD	76000	45000	FALSE	3/11/2019 15:25
f3c5aa40-2d51-11ea-8202-4ffd11dfcd72	customer021	Loan	550200	87600	FALSE	5/11/2019 20:05
f4f19b90-2d5b-11ea-8202-4ffd11dfcd72	customer014	FD	20000	39600	FALSE	16-11-2019 08:35

Figure 7 Sample Dataset

Step 2: The write transactions have performed on the dataset.

Step 3: Extract the SSTable event log information. The node tool is useful to extract the event log information from SSTable and will store it in a text file. In this phase, it is important to decide the audit period (can be weekly/biweekly/monthly. The upload/import of a text file having a log of operations should be generated at the end of the decided time period.

Step 4: Identify the suspicious transaction. Whether the performed transaction is suspicious or not, is decided by using specific conditions. These conditions are explained in Table 3.

Step 5: Present identified suspicious transactions to the user and ask for a decision whether to restore it or not. This result of the application tool has shown in Figure 8. In this phase, the user or user organization after being presented with transactions that the tool believes are likely to be suspicious, selects the ones that it actually wants to retrieve.

The data that the application deems suspicious, but the user doesn't wish to recover, represent the FP (false positives) because they were wrongly marked as positively suspicious by the tool whereas in reality the final call taken by the user or user organization proceed that they were a false alert. Finally, the user or user organization can fetch a .csv file of the data and records that have been selected for recovery. Later to complete the data recovery process, these records can be imported into the Cassandra database.

6.0 EXPERIMENTATION AND PERFORMANCE ANALYSIS

A system with specifications: windows 10 64-bit OS, 16 GB RAM with Intel Core i5 processor has been used for implementation. Application tool has been designed with HTML, PHP, JavaScript and Cassandra version 3.11.6. We analyze the effectiveness, performance and accuracy of the tool based on its ability to correctly perform the important functions like fetching suspicious delete/update transactions correctly, recovering deleted data, recovering data that goes missing due to incorrect update transactions. To evaluate the performance of the tool, a confusion matrix is created which can be depicted as below. The columns represent the actual values and the rows represent the predicted values. Positive stands for a transaction being marked suspicious by the application. On the other hand, negative indicates a transaction not being categorized as suspicious by the tool.

Table 3 Suspicious Transactions Conditions

ID	Suspicious Transaction		
	Type	**Identification Logic**	**Description**
1	Suspicious Delete	If (type_of_operation==Loan AND amount !=Zero)	If a person has taken a loan and the current loan outstanding amount is NOT ZERO, in this case if this loan transaction is deleted, then it should be a suspicious transaction.
2	Suspicious Update	If is_defaulter_loan flag is changed from YES to NO when the loan outstanding amount is NOT ZERO	If the current loan outstanding amount is not zero and loan amount is identified as a loan defaulter and somebody has updated it as a not a loan defaulter account, then transaction can be suspicious
		If Salary amount is updated and update month is NOT (March, April, May)	If salary amount is updated and update month is (March, April, May) because annual review is done at the end of financial year and salary data is updated in month of (March, April, May)
3	Suspicious Delete / Update	If transaction delete/update time is NOT office working hrs. i.e. 7AM to 9PM	Normally loan related transactions should happen during normal business hrs. If the transaction has been performed between non-working hrs, then transactions must be reviewed.
4	Suspicious Add	If typeOfOp= FD and FD amount is created to a value greater than 12 times the salary (monthly income)	Typically a fixed deposit is created by people to save an amount out of the salary they receive. If a fixed deposit transaction is created by adding a record where the fixed deposit amount is 12 times the monthly salary income of the person, then this transaction must be reviewed.

Analyze And Restore Data

Found following 5 suspicious transactions/s

	Original Data								Log Table Data	
	Sr No	**cust_trans_id**	**customer_id**	**type_of_operation**	**amount**	**salary_or_income**	**is_loan_defaulter**	**tans_date_time**	**type_of_suspect**	**updated_value**
☐	1	818a5930-2d51-11ea-8202-4ffd11dfcd72	customer015	Loan	300000	90000	TRUE	1/11/2019 15:15	Loan Defaulter Flag is Changed From YES to NO When Loan Outstanding Amount is NOT Zero	FALSE
☐	2	8467cb10-2d5b-11ea-8202-4ffd11dfcd72	customer001	FD	5000	5200	FALSE	12/11/2019 15:42	FD Amount > 12 Times Monthly Income	80000
☐	3	56095ef0-2d5b-11ea-8202-4ffd11dfcd72	customer020	Loan	902100	20000	FALSE	10/11/2019 15:45	Salary Amount is Updated and Update Month is January	22000
☐	4	f3c5aa40-2d51-11ea-8202-4ffd11dfcd72	customer021	Loan	550200	87600	FALSE	5/11/2019 20:05	Loan Outstanding Amount is NOT Zero, But the Loan Transaction is Deleted	550200
☐	5	181bf260-2d5b-11ea-8202-4ffd11dfcd72	customer003	FD	70000	14000	FALSE	7/11/2019 20:45	Transacation Modified NOT in Office Working Hours	70000

Figure 8 Allowing User to Select from Potential Suspicious Transactions

True positive (TP): The transactions belonging to this category are marked as suspicious by the tool developed and rightly so i.e. the user accepts and agrees with the tool categorizing it as suspicious

False Positive (FP): Here, in this set of transactions, the application does find certain transactions to be suspicious based on the logic that the tool follows. However, when the user organization scrutinizes these transactions, it finds that the tool wrongly marked them as suspicious and that those transactions were in fact completely valid. Here, the tool's inadequacy can be viewed. FPs are precisely the reason why human review is required.

False Negative (FN): In this section, the transactions included are those suspicious operations that were not identified by the application. False negatives are harmful to the user or user organization.

True Negative (TN): In this category are the valid operations of delete and update that the application rightly doesn't classify as suspicious.

The TP and TN part of the matrix indicate accurate functioning of the analysis tool wherein the suspicious nature of an operation is captured accurately and a valid transaction is marked valid. The cases wherein the tool demonstrates inadequacy are reflected in FP and FN. While the presence of FP is not that dangerous from the security perspective and merely adds to the burden of the user's job; the FN can have serious implications as potentially hazardous operations go undetected. To evaluate the accuracy of the model by using the confusion matrix, two measures are used namely 'Precision' and 'Recall'. Precision is calculated using equation (1) and recall is calculated using equation (2). The accuracy of the proposed tool is calculated using equation (3).

$$\text{Precision} = (TP) / (TP+FP) \quad (1)$$

$$\text{Recall} = (TP) / (TP+FN) \quad (2)$$

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN) \quad (3)$$

Now, for the giving set of 8 operations that occurred on table with over 190,800 entries, five operations of update and delete were classified as suspicious by the application tool. Out of these 5 transactions, 4 were decided by the user organization to be restored. The 1 update transaction which is categorized as being suspicious by the system but is actually a valid operation as per the user can be called a FP (False Positive). This FP update operation caught the attention of the tool owing to the fact that the FD amount for the customer was set to more than twelve times the monthly salary/income of the user. However, upon further scrutiny by the user, it was made evidently clear that a false alert had been raised by the application. On the other hand, a delete operation which should have been caught as being suspicious, does not get enlisted by the tool in the potentially suspicious transactions as it takes place at around half past eight in the evening and the rules for detection of suspicious delete operation only detects those operations that take place before 7 a.m. or after 9 p.m.. So, in this case where the delete operation is an insider's job, the potential threat has gone unnoticed. This particular transaction accounts for presence of FN (False negative) wherein an operation performed with malicious intent never gets enlisted as a potentially suspicious transaction. Thus, the entries in the confusion matrix can be noted to be as follows:

TP = 4, FP = 1, TN = 2, FN = 1

Now, taking the above observations into account, the performance of proposed tool is calculated for the dataset pertaining to financial institutions:

Precision = (TP) / (TP+FP) = 80%

Recall = (TP) / (TP+FN) = 80%

Accuracy = (TP+TN) / (TP+TN+FP+FN) = 75%

Security Analysis

The important aspects related to security can be considered as access to all suspicious transactions should be with admin user only and the application should ensure that no data breach occurs and information regarding recovered data is inaccessible to anyone not having required authority. As mentioned above, as far as security is concerned, it is worthwhile to reiterate that the presence of FN (False Negative) is a major security concern given that the tool's inability to detect potentially harmful operations by malicious users could cost a lot to the user organization.

7.0 CONCLUSION

It can be difficult to comprehend the specific database technology because of their different internal structures and query languages, yet it is essential to understand relational databases and NoSQL databases. Large datasets must be handled by forensic investigators carefully, sometimes necessitating the use of specialized instruments and methods for data extraction, processing, and analysis. In this article, we have proposed an application tool that has been designed to identify suspicious transactions and recover data specifically for the domain of financial institutions. The logic used for identification of potentially malicious-intent transactions are presented. The tool reduces human efforts needed as the user is now presented with potentially suspicious transactions and doesn't have to go through the entire history of operations performed on the database of the financial organization. This is useful to hold potentially suspicious users accountable for performing or instigating performance of malicious operations. This tool allows the user to download a CSV file to help in recovery of row entries that the tool finds suspicious and the user approves as being a result of malicious activity. Finally, performance analysis has revealed that the accuracy of the proposed scheme is 75% and the precision and recall, which are two measures of performance, calculated using the confusion matrix are both 80%. In the future, we intend to make the forensic analysis tool to perform its tasks without the need of external triggering. In the front-end, in future, we aim to add a functionality that will enable the tool to automatically find the number of columns and hence, the schema and attributes of the dataset which the tool takes as input.

Acknowledgment

We would like to thank Deccan Education Society Pune University, COEP Technological University, Pune and

Marathwada Mitramandal's College of Engineering, Pune for providing valuable support for the conduction of this study.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper

References

[1] N. F. Mohammed, S. A. Ali, and M. J. Jawad, 2020. "Biometric-based medical watermarking system for verifying privacy and source authentication," *Kuwait Journal of Science,* 47(3)

[2] R. Chopade and V. K. Pachghare, 2019. "Ten years of critical review on database forensics research," *Digital Investigation,* 29.

[3] "DB-Engines Ranking - popularity ranking of database management systems (n.d.)", Retrieved March 15, 2023, from https://db-engines.com/en/ranking

[4] Yoon, J. and Lee, S., 2018 "A method and tool to recover data deleted from a MongoDB", Digital Investigation, 24: 106-120.

[5] Chopade, R. and Pachghare, 2021.V., "A data recovery technique for Redis using internal dictionary structure", *Forensic Science International: Digital Investigation*, 38: 301218.

[6] Documentation. (n.d.). Retrieved January 26, 2023, from https://cassandra.apache.org/doc/latest/

[7] "How is data written? | Apache Cassandra 3.x. (n.d.)", Retrieved January 26, 2023, from https://docs.datastax.com/en/cassandra-oss/3.x/cassandra/dml/dmlHowDataWritten.html

[8] E. C. Cankaya and B. Kupka, 2016. "A survey of digital forensics tools for database extraction," in Future Technologies Conference (FTC), 1014–1019.

[9] R. Chopade and V. Pachghare, "Evaluation of Digital Forensic Tools in MongoDB Database Forensics," in *Progress in Advanced Computing and Intelligent Engineering,* 427–439. Springer,

[10] A. Prasad and B. N. Gohil, 2014. "A Comparative Study of NoSQL Databases.," *International Journal of Advanced Research in Computer Science*, 5(5)

[11] N. Mangle and P. B. Sambhare, 2013. "A Review on Big Data Management and NoSQL Databases in Digital Forensics," *International Journal of Science and Research,* 4.

[12] W. K. Hauger et al. 2018. "*Forensic attribution challenges during forensic examinations of databases*," University of Pretoria,

[13] H. Shahriar and H. M. Haddad, 2017. "Security vulnerabilities of nosql and sql databases for mooc applications," *International Journal of Digital Society*, 8(1): 1244–1250,

[14] W. K. Hauger and M. S. Olivier, 2018. "NoSQL databases: forensic attribution implications," *SAIEE Africa Research Journal*, 109(2): 119–132,

[15] R. Chopade and V. Pachghare, 2020 "Performance Analysis of Proposed Database Tamper Detection Technique for MongoDB," in International Congress on Information and Communication Technology.393–400.

[16] A. Golhar, S. Janvir, R. Chopade, and V. K. Pachghare, "Tamper Detection in Cassandra and Redis Database—A Comparative Study," pp. 99–107, 2020, DOI: 10.1007/978-981-15-0790-8_11.

[17] Kumbhare, R., Nimbalkar, S., Chopade, R., Pachghare, V.K. 2020. Tamper Detection in MongoDB and CouchDB Database. In: Bhalla, S., Kwan, P., Bedekar, M., Phalnikar, R., Sirsikar, S. (eds) *Proceeding of International Conference on Computational Science and Applications. Algorithms for Intelligent Systems*. Springer, Singapore. DOI: https://doi.org/10.1007/978-981-15-0790-8_12

[18] R. Rowlingson et al., 2004. "A ten step process for forensic readiness," *International Journal of Digital Evidence,* 2(3): 1–28,

[19] I. Homem, 2018. "*Advancing Automation in Digital Forensic Investigations," Department of Computer and Systems Sciences*, Stockholm University,

[20] A. Abadi, A. Haib, R. Melamed, A. Nassar, A. Shribman, and H. Yasin, 2016. "Holistic disaster recovery approach for big data NoSQL workloads," in *2016 IEEE International Conference on Big Data (Big Data).* 2075–2080.

[21] A. Kathpal and P. Sehgal, 2017. "${$BARNS$}$: Towards Building Backup and Recovery for NoSQL Databases," *Workshop on Hot Topics in Storage and File Systems.*

[22] Unable to delete or insert/update data (or "missing data") – DataStax Support. (n.d.). Retrieved June 28, 2023, from https://support.datastax.com/hc/en-us/articles/360006487657-Unable-to-delete-or-insert-update-data-or-missing-data.

[23] V. Pachghare and R. Chopade, 2021. "A technique to analyze a cassandra tombstone for identification of operations", *In Advances in Intelligent Systems and Computing,* 1184.

[24] Key Concepts: Cassandra SSTable Storage Format. (n.d.). Retrieved December 12, 2023, from http://distributeddatastore.blogspot.com/2013/08/cassandra-sstable-storage-format.html