IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 2, 2020, accepted February 20, 2020, date of publication February 27, 2020, date of current version March 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976885

Database Forensic Investigation Process Models: A Review

ARAFAT AL-DHAQM[1], SHUKOR ABD RAZAK[1], SITI HAJAR OTHMAN[1], ABDULALEM ALI[1], FUAD A. GHALEB[1], ARIEFF SALLEH ROSMAN[2], AND NURAZMALLAIL MARNI[3]

[1]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM), Johor 81310, Malaysia
[2]Center for Research in Fiqh Science & Technology, Universiti Teknologi Malaysia (UTM), Johor 81310, Malaysia
[3]Academy of Islamic Civilization, Faculty of Social Sciences and Humanities, Universiti Teknologi Malaysia (UTM), Johor 81310, Malaysia

Corresponding author: Arafat Al-Dhaqm (mrarafat@utm.my)

ABSTRACT Database Forensic Investigation (DBFI) involves the identification, collection, preservation, reconstruction, analysis, and reporting of database incidents. However, it is a heterogeneous, complex, and ambiguous field due to the variety and multidimensional nature of database systems. A small number of DBFI process models have been proposed to solve specific database scenarios using different investigation processes, concepts, activities, and tasks as surveyed in this paper. Specifically, we reviewed 40 proposed DBFI process models for RDBMS in the literature to offer up-to-date and comprehensive background knowledge on existing DBFI process model research, their associated challenges, issues for newcomers, and potential solutions for addressing such issues. This paper highlights three common limitations of the DBFI domain, which are: 1) redundant and irrelevant investigation processes; 2) redundant and irrelevant investigation concepts and terminologies; and 3) a lack of unified models to manage, share, and reuse DBFI knowledge. Also, this paper suggests three solutions for the discovered limitations, which are: 1) propose generic DBFI process/model for the DBFI field; 2) develop a semantic metamodeling language to structure, manage, organize, share, and reuse DBFI knowledge; and 3) develop a repository to store and retrieve DBFI field knowledge.

INDEX TERMS Database forensic, digital forensic, investigation process model.

I. INTRODUCTION

Database Forensic Investigation (DBFI) is a branch of Digital Forensics (DF) that examines database contents [1] to identify, detect, acquire, analyse, and reconstruct database incidents as well as construct a chronological timeline of intruder activities. The current DBFI literature has generally focused on case-by-case or ad hoc scenarios, and there remain several challenges that have yet to be addressed such as:

1. Some works come from before the advent of the Internet and thus only exist in ''paper-form'', are mainly unclear, and have absent documents.
2. There are only a very small number of review papers on this topic such as those published in 2009 [1].
3. Each database system has a different infrastructure, which results in specific DBFI models and processes [18].
4. A Database System (DBMS) has three dimensions: internal, conceptual, and external [18]. Thus, the multidimensionality of DBMS potentially complicates investigations if investigators are not familiar with one or more database dimensions.

There exists an absence of standardized models that unify concepts and terminologies to reduce confusion and assist in organizing and structuring field knowledge.

This paper has three main objectives: 1) present a broad literature review of the DBFI domain that will assist field researchers in comprehending DBFI from different perspectives; 2) discuss the issues and drawbacks of the DBFI domain; and 3) suggest some solutions for the discovered limitations.

The rest of this paper is structured as follows: Section 2 provides the study background and related works. Section 3 gives a brief overview of the digital forensic field. Section 4 reviews the identified DBFI models. Section 5 presents the research methodology. Section 6 gives

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

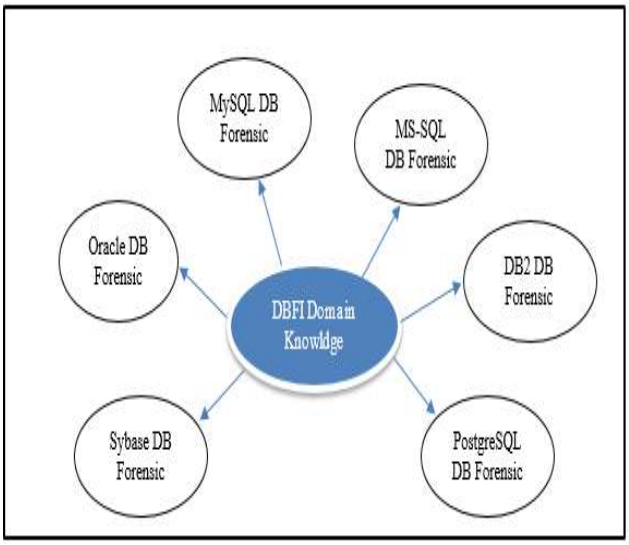

FIGURE 1. Practical DBFI studies on various DBMSs.

the discussion and analysis results. Section 7 concludes this paper.

II. BACKGROUND AND RELATED WORKS

DBFI deals with database contents, specifically their metadata (data dictionary). This helps achieve several tasks such as identification, collection, preservation, reconstruction, analysis, and the documentation of evidence against database cases [1]. However, the complexity and multidimensionality of Database Management Systems (DBMSs) have been the sole focus of DBFI research [1]–[6]. Most importantly, there have been limited practical studies on solving specific DBFI issues. Specific DBFI studies have covered various DBMSs as shown in Fig 1. For example, specific and limited investigation models for overall Oracle database concepts and techniques have been widely proposed [7]–[16]. Similarly, the forensic investigation model studied by [15] used specific steps to discover the information operations performed on the database discussed by [1]. The Log Miner tool was investigated in [17], which permitted a Database Administrator (DBA) or forensic analyst to rebuild the actions that took place in a database [18]. In addition, Litchfield suggested seven (7) practical investigation forensic models. Litchfield addressed information accessibility from different sources such as redo logs, dropped objects, authentications, flashbacks, and the recycling bin. A forensic textbook published on the Oracle database by [17] focused on practical issues for DBA [1]. Furthermore, the investigation model for fact collection for compromised Oracle databases presented by [14] was established on a sequence of practical methods that were originally suggested by Litchfield.

Only a few studies on Microsoft SQL (MSSQL) databases that have used specific forensic practices have been noticed in the popular literature [2], [19]–[22]. The SQL Server Forensic Analysis Methodology is one of these available practical databases [2]. Reference [2] consists of 4 investigated phases: preparation, incident verification, artifact collection, and artifact analysis for MSSQL server databases [5], [23]. Another applied scenario for real world cases was conducted by [20] to collect and examine signs from a conceded database using MSSQL. It entails practical ideas on how a database can changed. Another forensic tamper detection model for sensitive data was created by [21] while a detection and investigation model was developed by [19]. These models detect database servers and investigate collected data. A methodology for detecting suspicious transactions within a database was proposed by [22] that monitors database transactions on a continuous basis and helps make decisions on whether a databases transaction is legitimate or suspicious by combining multiple pieces of evidence.

None of the previously mentioned models focused on creating a common model for the DBFI field. The process models solved specific database incidents, scenarios, or case studies. Accordingly, they contain irrelevant processes that rendered certain activities and tasks redundant.

Moreover, MySQL RDBMS consists of a few practical forensic studies investigated by [3], [24]–[28]. To mention only a few, a framework on MySQL database forensic analysis was developed by [3] that concentrated on the discovery of malicious tampering. In [24], a MySQL database model for the detection of inconsistencies was studied to identify and sense conflicts in database records. In order to reconstruct basic SQL statements through InnoDB redo logs, a rebuild of basic SQL statements was proposed by [25]. The study by [25] focused on Data Manipulation Language (DML) statements and overlooked Data Definition Language (DDL) declarations. Previous reconstruction models were enhanced by [26], including DDL reconstruct statements. Additionally, the technical investigation model suggested by [27] showcased admittance for a user's MySQL database without the need for user assistance. This is advantageous in emergencies where the user is absent or under examination. The forensic investigation approach by [28] was proposed to test the forensic richness of storage engines in MySQL DBMS. [28] features three investigation processes: preliminary analysis, execution, and analysis. The study of [28] investigated the impact of storage engines on the generation of persistent forensic data in MySQL DBMS systems.

From the above statements, it can be understood that previous research on DBFI approaches mainly discussed the DBFI field from three perspectives: technology, investigation processes, and dimensions as highlighted by [18]. In view of this, the DBFI field lacks a structured and unified model for facilitating, managing, sharing, and reusing DBFI field knowledge amongst field practitioners [4], [23], [29], [30].

III. DIGITAL FORENSICS

Digital Forensics (DFs) are applied to ensure the consistency and truthfulness of evidence gathered at computer crime scenes. DFs includes the identification, extraction, preservation, analysis, documentation, and explanation of computer

data [31]. However, technological growth has emphasized other elements of DF investigations. DF investigations have grown from computers and networks to include portable electronic devices, graphics, software, and DBFI [32], [33]. Computer Forensics explains a wide range of log file information. It examines internet histories by utilizing the actual electronic files inside a drive [34]. Mobile Forensics focuses on simple data such as call logs, Short Message Services (SMS), or emails [35]. Network Forensics is connected to the monitoring and analysis of computer network traffic [36]. However, this study focuses on the DBFI domain. It was observed by [18], [30], [4], [37], [38] that DFs are not suitable for database systems because of their diversity and multi-dimensionality. Also, DFs pay attention to one dimension (file systems) [1] and mainly focus on identification, collection, handling, storage, incident response, and training [37]. However, database incidents may be difficult to trace unless multiple digital investigation aids are combined with database analysis [37]. In addition, DF practices do not cover transactional database concepts [37], [39]. In conclusion, the DF domain has difficulties in working with the DBFI domain. The multidimensionality and diversity of RDBMS is a barrier to researchers hoping to develop a standard approach for the DBFI domain. For this reason, existing DF models do not cover database system concepts [39]. The next section provides an overview of DBFI models.

IV. DATABASE FORENSIC MODELS OVERVIEW

Section III discusses the DF domain. The DF domain has difficulty working along the DBFI domain due to multidimensional nature and the diversity of RDBMS. This section provides an overview of DBFI models.

Generally, the DBFI field deals with database contents and metadata that connotes (data dictionary) the comparison of documented evidence against database incidents [1], [3]. However, a surfeit of different kinds of investigations using different approaches has been proposed in the literature [2], [5], [7]–[16], [19]–[22], [3], [24]–[28], [30], [40]–[45]. For example, an investigation process model was developed by [15] that performed tasks to discover the operations carried out on a database [1]. Their solution featured four research processes: shelving database operations, gathering data, rebuilding the database, and fixing database integrity. However, the emphasis was on Oracle database concepts. Additionally, a Log Miner tool was developed by [17] for Oracle databases that reconstructed actions that occurred when auditing features were turned off. Nevertheless, it is inadequate for forensic analysis due to the anomalies present in forensic analysis [18].

[7] Proposed model to demonstrated how an examiner can use an Oracle log file to reveal attacker events. The binary format of redo logs shows forensic examiners evidence that can be found and investigated. It covers how evidence can be integrated into a timeline of events. The study also discovered how attackers' cover their tracks after a failed attempt and how to spot them. Redo logs were emphasized as an amusing source of evidence for a forensic examiner when investigating a compromised Oracle database server.

[8] developed model to recover evidence from dropped or purged Oracle objects. It allows investigators to recover evidence directly from the data files of a compromised server, although an attacker may drop objects. Several Oracle views and tables assist the investigator in locating dropped objects such as OBJ$, SOURCE$, IDL_UB1$, IDL_CHAR$, and RECYCLEBIN$.

The investigation model to captures evidence of attacks against authentication mechanisms using Listener log files and audit trails was proposed by [9]. Listener log files contain details about connections to database servers such as IP addresses, Service Identifier (SID) names, and instance names. Audit trails may contain details about successful and unsuccessful logins and logoffs. Thus, an examiner can gather evidence on authentication mechanisms from Listener log files and audit trails.

The disconnection of database servers from a network to capture volatile data was proposed by [10]. Two investigation processes were proposed to retrieve fragile data from database servers, Identification and Evidence Collection. The identification process deals with the disconnection of database servers from a network, the preparation of the forensic environment, and the forensic techniques used move captured data. The evidence collection process was used to collect volatile data from compromised database servers. Recovering and carefully storing volatile data for later analysis requires the use of forensic studies. It gives forensic inspectors the opportunity to collect non-volatile data in a ''human readable'' form that is easier to observe than stored binary.

The detection investigation forensic model, was proposed by [11] to discuss how examiners find evidence of data theft in the absence of auditing. The model shows how an Incident Responder/DBA may determine a breach of an Oracle database server has occurred when there is no audit trail and it is suspected that an attacker has gained unauthorized and select access to data. Furthermore, a textbook was published on Oracle databases by [16]. However, it was written at the practical level and was intended for DBA, and as such did not focus on an underlying/cognitive model [1]. Also, an investigation collection model was proposed by [14] to collect evidence from Oracle databases that was based on the series of practical models proposed by David Litchfield for analysing database tampering in Oracle databases.

In 2008, [2] proposed a SQL server forensic analysis methodology to collect and analyse evidence from MSSQL server databases. It consists of four phases: investigation preparation, incident verification, artefact collection, and artefact analysis. It deals specifically with SQL server databases [2]. Also, another database server detection and investigation process model was proposed by [19] to detect database servers and collect data. It consists of three phases: server detection, data collection, and collected data investigation. However, it cannot deal with volatile artefacts. The

detection of inconsistencies database model was proposed by [3] to identify and name bytes and interpret them for MySQL database systems. With this knowledge, it is possible to detect inconsistencies in a database. However, nothing has been discovered on using multiple log files and caches for further analysis [3]. The model used MySQL database server log artefacts. Additionally, the reconstruction model was proposed by [25] to reconstruct basic SQL statements from redo logs to restore deleted or updated values. However, it concentrates on DML statements and ignores the basic DDL statement [26].

A practical forensic approach for reconstructing basic SQL DDL statements was proposed by [26] to enhance the previous approaches. A framework was proposed by [4] to identify, collect, analyse, validate, and document digital evidence to discover malicious tampering. It was based on two stages. Stage 1 collected and analysed non-volatile data and Stage 2 collected, analysed, and reconstructed volatile data to compare results. Apart from the various DBFI domain knowledge approaches proposed for DBMS, there are also several forensic tamper detection models and analysis algorithms for database systems that have been proposed in the literature. For example, a discovering methodology and scenarios for detecting covert database systems was developed by [46] to assist investigators in discovering and detecting covert database systems. A model for efficient digital evidence collection was proposed by [47] to collect evidence from business databases on authorized and unauthorized events using database features such as triggers, log file backups, and replications.

A forensic tamper detection model was proposed by [48] to detect compromised database audit logs using strong one way hash functions. However, this algorithm cannot analyse intruder activities or decide when tampering has occurred, what data was altered, or to identity an adversary [48]. A model to investigate a compromised database management system was proposed by [4] that consisted of two examination processes: identification and collection. The identification process prepares database forensic layers, methods, and environment while the collection process gathers suspicious database management system data and moves it to a protected area for further forensic examination.

A list of digital forensics tools for extracting, recovering, analyzing, and documenting data from databases was provided by [49]. A model proposed by [50] was used to assess the integrity of live databases by recognizing and reporting log tampering based on the forensic analysis of database storage and the detection of inconsistencies between database logs and physical storage states. A new driven model was presented by [52] to derive solution models for DBFI to facilitate the storage, management, sharing, and reuse of DBFI domain knowledge. A rebuilding tool was presented by [53] to rebuild original database schema when databases have been compromised or destroyed. A model to collect, preserve, and analyse database metadata and database attacks was proposed by [42] that consists of four investigation processes: collection and preservation, analysis of anti-forensic attacks, analysis of database attacks, and preserving evidence reports. Additionally, [41] proposed a model to reconstruct database events and detect intruder activities that consisted of two investigation processes: collection and reconstructing evidence. The collection process gathers evidence by replicating sources and the reconstructing evidence process rebuilds user activities and detects malicious activities. Recently, a review paper was introduced by [54] that focused on the last ten years relational databases forensic analysis research and artefacts. The next section discusses the research methodology used in this study.

V. RESEARCH METHODOLOGY AND RESEARCH QUESTIONS

The researchers used a process adapted from [39]. The process consisted of 3 phases:

i) Selecting a field topic

ii) Selecting online databases and finding relevant literature

iii) Reviewing existing literature.

Thus, a detailed study of existing DBFI process models was conducted to understand common issues and challenges in the DBFI field.

A. PHASE I: SELECTING A FIELD TOPIC

In this stage the selection topic was determined. The chosen topic was determined using questions relating to what the topic addresses or how the topic background. There were three fundamental questions that became research references:

1. What are the DBFI process models that already exist in the literature?
2. Are there any generic models/frameworks for the DBFI field?
3. What are the limitations of existing DBFI process models and what are possible solutions to address those limitations?

B. PHASE II: SELECTING ONLINE DATABASES AND FINDING RELEVANT LITERATURE

In this stage, the scope of the review was determined. This study used the phrase ''Database Forensic'' to find a collection of models related to DBFI. This step gathered knowledge sources. The Web of Science, Scopus, IEEE Explore, ACM, Springer Link, and Google Scholar are famous digital libraries that were used to identify relevant papers in the DBFI field. For this purpose, this study used the search keyword ''Database Forensic''. Searches were limited to 2004-2019. This produced a total of 40 out of 919 articles from all database search engines. Thus, forty (40) out of 924 articles were found to focus purely on DBFI processes, activities, database crimes, concepts, and tasks after the removal of duplicates and public health and medicine articles as well as screening for topic and abstracts. In this study, research articles, conference papers, books, book chapters, and dissertations were considered while other types of documents

TABLE 1. Systematic review protocols.

Database Search Engines	"Database Forensic"
Web of Science	12
Scopus	46
IEEE Explore	6
Springer Links	61
Google Scholar	794
ACM	5
Total	924

were excluded from the analysis. Also, articles that discussed Deoxyribonucleic Acid (DNA) were also removed. The search protocols are summarized in Table 1. The next section reviews the exiting literature on the DBFI domain.

C. PHASE III: EXISTING LITERATURE REVIEW

This study discovered that researchers and developers dealt with the DBFI field from three perspectives:

i) DBFI Dimensions (destroyed, compromised, and modified)

ii) DBFI Technology (tools, algorithms, and methods)

iii) DBFI Processes (preparation, collection, analysis, and presentation).

However, this research varies in perspective, coverage, and findings. Notably, certain models covered all three DBFI perspectives, whereas others only focused on two or one.

DBFI Dimension Perspective: DBFI was classified into three dimensions by [18] as: compromised, damaged, and modified. A compromised database (conceptual dimension) is defined as a database where some metadata or software in the Database Management System (DBMS) has been modified by an attacker, even though the database is still operational [18]. A damaged database (internal dimension) refers to database where data or data files may have been modified, deleted, or copied from their original location into other places. These databases may or may no longer be operational depending on the extent of the damage [18]. A modified database (external dimension) refers to a database that has not been compromised or damaged but has undergone changes due to normal business processes since the event of interest [18].

DBFI Technology Perspective: The technology perspective covers forensic tools, algorithms, and methods in the DBFI domain [7]–[15], [43]. For example, the Log Miner tool proposed by [55] allows a DBA or forensic analyst to reconstruct the actions that took place in a database. Moreover, seven (7) forensic tools, algorithms and methods have been proposed by Litchfield [9], [7], [8], [11], [56] that address information from redo logs, dropped objects, authentication, flashbacks, and the recycling bin.

DBFI Process Perspective: This perspective contains investigation process models that discuss the DBFI domain [2], [15], [20], [27], [38], [46], [3], [57]–[59]. For example, the SQL Server Forensic Analysis Methodology proposed by Fowler [2] consists of four investigation phases: investigation preparation, incident verification, artifact collection, and artifact analysis. The SQL Server Forensic Analysis Methodology deals with MSSQL server databases. Also, a detection and investigation model was developed by [19] to detect database servers, collect data, and investigate the collected data.

Therefore, this paper classified models into two categories based on their coverage [33], [60]. The first category contained models that covered at least two DBFI dimensions, contained DBFI technology, and had at least two investigation processes. These were called ''full-coverage'' models due to the fact they covered a wide range of DBFI perspectives. The second category included models that covered two DBFI dimensions, contained DBFI technology, and only had one investigation process. These are called ''partial-coverage'' models due to the fact they cover a partial range of DBFI perspectives. Based on these categorizations, this study found that twenty-three (23) out of forty (40) models were full-coverage models and fifteen (15) out of forty (40) models were partial-coverage models. The rest of the models that covered a specific DBFI perspective, which were called ''specific-coverage'' models, were ignored by this article. Table 2 shows the categorization of these DBFI models. The limitations of the DBFI process models are discussed from the following three dimensions, which are discussed in detail in Section 5:

1) Redundant and irrelevant investigation processes

2) Redundant and irrelevant investigation concepts and terminologies.

3) A lack of unified models that manage, share, and reuse DBFI knowledge.

1) REDUNDANT AND IRRELEVANT INVESTIGATION PROCESSES

Due to the diversity of RDBMS infrastructure, several DBFI process models have been proposed to deal with DBFI from an investigation process perspective. However, none of these models are a common model. The process models that have been proposed solve specific database incidents, scenarios, or case studies. Consequently, they produce redundant and irrelevant processes that have rendered certain activities and tasks redundant. The proposed models discuss the DBFI field from four investigation process perspectives: preparation, collection, analysis, and presentation.

Preparation Process Perspective: The models that discuss the DBFI field from a preparation process perspective have various redundant investigation processes, activities, and tasks as shown in Table 3. For example, the ''Suspension of Database Operation'' process proposed by [15] is used to isolate database servers from users in order to capture database activities, while the ''Verification'' process proposed by [20] is used to verify and check incidents as well as isolate database servers. The ''Identification'' process proposed by [10] is used to disconnect database servers from the network in order to capture volatile data. Two investigation processes were proposed by [2]: ''Investigation preparation''

TABLE 2. Categorization of the DBFI process models.

ID	Year	Database Forensic Investigation Models	First Category	Second Category
1.	2004	System and method for investigating a data operation performed on a database [15]	√	
2.	2004	Oracle database forensics using LogMiner [43]		√
3.	2005	Forensic Analysis of a SQL Server 2005 Database Server [20]	√	
4.	2006	Forensic tamper detection in SQL server [21]		√
5.	2007	Oracle Forensics Live Response [10]	√	
6.	2007	Finding Evidence of Data Theft in the Absence of Auditing [11]		√
7.	2007	Discovering Methodology and Scenario to Detect Covert Database System [46]		√
8.	2007	Oracle Forensics Part 2: Locating dropped objects [8]		√
9.	2008	SQL Server Forensic Analysis Methodology [2]	√	
10.	2009	Database forensic investigation based on table relationship analysis techniques [57]	√	
11.	2009	Evidence Investigation Methodologies for Detecting Financial Fraud based on Forensic Accounting [68]	√	
12.	2009	On metadata context in Database Forensics [1]	√	
13.	2009	Efficient model for detection data and data scheme tempering with purpose of valid forensic analysis [47]		√
14.	2010	Methods for Efficient Digital Evidence of Collection of Business Processes and Users' Activity in eLearning Environments [69]		√
15.	2011	The Method of Database Server Detection and Investigation in the Enterprise Environment [19]	√	
16.	2011	Assembling Metadata for Database Forensics [70]		√
17.	2012	Digital Evidence for Database Tamper Detection [14]	√	
18.	2012	Framework for Database Forensic Analysis [3]	√	
19.	2012	A Workflow to Support Forensic Database Analysis [38]	√	
20.	2012	On Dimensions of Reconstruction in database forensic [18]	√	
21.	2012	Reconstruction in Database Forensics [23]		√
22.	2012	Arguments and Methods for Database Data Model Forensics [71]		√
23.	2013	Forensic Analysis of Databases by Combining Multiple Evidence [22]	√	
24.	2014	Database Forensic :Investigating Compromised Database Management Systems [4]	√	
25.	2014	Role of metadata in forensic analysis of database attacks [42]	√	
26.	2014	Towards a forensic-aware database solution: Using a secured database replication protocol and transaction management for digital investigations [41]	√	
27.	2014	Schema reconstruction in database forensics [53]		√
28.	2014	Forensic Investigation of MySQL Database Management System [27]		√
29.	2014	Towards adapting metamodelling technique for database forensics investigation field [45]		√
30.	2015	Ideal log setting for database forensics reconstruction [6]	√	
31.	2015	Database forensic analysis through internal structure carving [5]	√	
32.	2016	A Methodology to Test the Richness of Forensic Evidence of Database Storage Engine: Analysis of MySQL Update Operation in InnoDB and MyISAM Storage Engines [28]	√	
33.	2016	Conceptual investigation process model for managing database forensic investigation knowledge	√	
34.	2016	A generic database forensic investigation process model	√	
35.	2017	Database Forensic Analysis with DB Carver [40]		√
36.	2017	Development and validation of a Database Forensic Metamodel (DBFM) [72]	√	
37.	2017	CDBFIP: Common Database Forensic Investigation Processes for Internet of Things [73]		√
38.	2017	Implementing Chain of Custody Requirements in Database Audit Records for Forensic Purposes [74]		√
39.	2017	Carving database storage to detect and trace security breaches [50]		√
40.	2018	A method and tool to recover data deleted from a MongoDB [51]		√
41.	2018	Five Stages of Database Forensic Analysis: A Systematic Literature Review [75]	√	
		Total	**23**	**15**

and ''Incident verification'' to identify and verify database incidents through a preliminary investigation, prepare forensic workstations and forensic toolkits to respond to incidents, and then disconnect database servers.

Collection Process Perspective: The models that discuss the DBFI field from the collection process perspective have various redundant investigation processes, activities, and tasks as shown in Table 4. For example, the ''Collecting data'' process proposed by [15] is used to gather data, metadata, and intruder activities from database servers, while the ''evidence collection'' process proposed by [20] is used to collect evidence from victim database servers such as SQL Server connections, session data, transaction logs, database files, default SQL Server trace files, and SQL Server error logs. The ''Artefact collection'' process proposed by [20] is used to collect volatile and non-volatile MSSQL Server database artefacts such as log files, data files, data caches, transaction logs, log files, and windows log events. The ''Data Extraction'' and ''Table Relationship Search and Join'' processes proposed by [57] are used to extract data from database tables and collect various file types, such as email attachments, multimedia files, and images from file servers and database systems. The ''Artefact Collection'' process proposed by [3] is used to collect data from files identified in the previous processes. A summarization of collection processes models is illustrated in Table 4. The ''Artefact Collection'' process proposed by [22] is used to collect volatile and nonvolatile MSSQL Server database artefacts such as log files, data files, data caches, and transaction logs. The ''Collect suspect database system'' process proposed by [4]

allows investigators to collect and extract suspect database management system data and move it to a secure area for further forensic investigation. The ''Collection and Preservation Process'' proposed by [42] is used to collect log files (database files, transaction logs, cache files, text files, binary log files, error log files, server error logs, and memory dumps,) and protect metadata collected from log files. The ''Collection'' process proposed by [41] is used to gather evidence by replicating investigation sources. Finally, the ''Execution'' process proposed by [28] allows investigators to use forensic tools and procedures to create forensic images and collect metadata values from identified target files.

Analysis Process Perspective: The process models that discuss the DBFI field from an analysis process perspective have various redundant investigation processes, activities, and tasks. For example, two processes were proposed by [15] to reconstruct and restore database systems: ''reconstructing a database'' and ''restoring database integrity''. ''Reconstructing a database'' is used to rebuild intruder activities and reveal malicious actions, while ''restoring database integrity'' is used to restore database consistency. Four investigation processes were proposed by [20] to analyze database crimes: ''Timeline Creation'', ''Media Analysis'', ''Data Recovery'', and ''String Search''. The ''Timeline Creation'' process is used to construct an initial timeline that maps out notable digital events for use during the ''Media Analysis'' process. The ''Media Analysis'' process uses the timeline constructed in the ''Timeline Creation'' process to reveal malicious intruder activities. After discovering malicious activities, the database system recovers data for user access through the ''Data Recovery'' process. The ''Search String'' process is used to further investigate transactions outside the scope of the investigation to identify rows for reconstruction. A summary of the analysis process models is illustrated in Table 5.

2) REDUNDANT AND IRRELEVANT INVESTIGATION CONCEPTS AND TERMINOLOGIES

This issue is somewhat related to the first issue. The frequency/redundancy of investigation processes in DBFI models produces many redundant/frequent concepts, activities, and tasks that share meanings, functions, or names. For example, [2], [20] defined the ''Event'' concept as ''The event which added to timeline'', [14] defined the ''Event'' concept as ''The events which are copied to the collection server for analysis'', and [3] defined the ''Event'' concept as ''Events with failed database login attempts, successful login for user, and irregular activity of database that can be identified as having been added to the investigation timeline''. In addition, [1], [4], [6], [10] mentioned a similar concept called ''Incident'' and defined it as ''any action implemented to compromise the confidentiality, availability and integrity of an information system'', whereas [1] defined it as an ''action/event that corrupts the data accidentally or deliberately caused and compromises the confidentiality, availability and integrity of an information system''.

TABLE 3. Preparation process models.

No	Model source	Similar Processes	Activity and Meaning
1.	[15]	Suspend Database Operations	Database operations are suspended, at least long enough to capture evidence of the intruder's actions. This may entail disabling new logins, terminating any or all existing sessions and disconnecting the database from users.
2.	[20]	Verification	Verifies and checks incident, isolates database server and confirms the incident.
3.	[20]	System Description	Document the system information that identified in verification process such as system name, serial number, operating system, system function, and physical description.
4.	[10]	Identification process	Deals with disconnecting database server from network to capture volatile data as well as prepare forensic environment and forensic techniques used to move captured data.
5.	[57]	Identification process	Identification process is used to disconnect database server from network to capture volatile data as well as prepare forensic environment and forensic techniques used to move captured data.
6.	[2]	Investigation Preparation	Identifies and prepares forensic workstations and forensic toolkits to respond to an incident and then disconnect from the database server.
7.	[2]	Incident Verification	Verifies the database incident through preliminary investigation.
8.	[57]	Table Relationship Search and Join Process	Used to extract all table-spaces in the database, select the target, select the tables which store investigation data, and repeatedly check the other table field.
9.	[68]	Data Acquirement with Seizure and Search Warrant	Detect and secure the database system resources and gather evidence that relate to the accounting fraud. Also, protect the data resources of the corporation. Moreover, conduct an interview with DBA to validate the existence of a server managed by the corporation.
10.	[19]	Server Detection	Server detection is used to identify and detect the victim database server.
11.	[14]	Setup Evidence Collection Server	Preparing the investigation environment to reveal an incident.
12.	[38]	Incident reporting	capture database incident through user report, system audit, or triggered events
13.	[18]	Determining Acquisition Method	Identifies the proper acquisition methods for that dimension.
14.	[4]	Identification	prepare the database forensic layers, methods and environment
15.	[41]	Preliminary analysis	create an architectural visualization of the, identify files and folders in layers below the storage engines' layer, prepare and use forensic tools and procedures to create an initial image, collect metadata values of the identified target files, and record the metadata of the target files.

3) A LACK OF UNIFIED MODELS THAT MANAGE, SHARE, AND REUSE DBFI KNOWLEDGE

Several investigation models have been proposed for the DBFI field by previous researchers. However, the proposed

TABLE 4. Collection processes models.

No	Model source	Similar Processes	Activity and Meaning
1.	[15]	Collecting Data	Collecting data is going to assemble data, metadata and intruder activities from the database server.
2.	[20]	Evidence Collection	Collects evidence from victim database server.
3.	[10]	Collection process	Collection process uses collected volatile data from compromised database server.
4.	[2]	Artefact Collection	Artefact collection is used to collect volatile and nonvolatile MSSQL Server database artefacts such as log files, data files, a data cache, transaction logs, and log files.
5.	[57]	Data Extraction Process	Used to extract data from all database tables that identified and located in Table Relationship Search and Join Process. Additionally, collect the various file types, such as attached files of email, multimedia files and image files that stored in the file server, and in the database systems.
6.	[68]	Begging of Investigation	Extract fraud data from the database server.
7.	[1]	Metadata Extraction	Metadata Extraction is used to extract the metadata of database dimensions that are used to determine who was authorized to perform a certain action.
8.	[19]	Data Collection	Data Collection is divided into a stage of selectively collecting files and a stage of collecting the entire files.
9.	[14]	Collecting Files	Collecting Files is used to gather data from the specified sites such as redo logs, data blocks, audit trails, live response, views, oracle recycle bin, and system change number.
10.	[3]	Artefact Collection	Artefact collection is used to collect and extract database files and metadata from compromised databases.
11.	[38]	Collection process	Collect physical and digital data.
12.	[18]	Collection of Non-volatile Artefacts	Collection of nonvolatile artefacts such as database files, log files, and log transactions.
13.	[18]	Collection of Volatile Artefacts	Collect volatile artefacts such as data caches, redo log and undo log
14.	[22]	Artefact Collection	Artefact collection is used to collect volatile and nonvolatile MSSQL Server database artefacts such as log files, data files, data cache, transaction logs, log files and so on
15.	[4]	Collection suspect database system	The collection process is used to collect and extract suspected database management system data and move them for further examination.

TABLE 5. Analysis processes models.

NO	Model	Similar Processes	Activity or Meaning
1.	[15]	Restoring Database Integrity	Restoring database integrity is used to restore database consistency.
2.	[20]	Data Recovery	Recover data to be ready for user access
3.	[20]	Search String	Used to further investigation into transactions which occurred outside of the scope of this investigation to identify rows for reconstruction.
4.	[2]	Artefact Analysis	Artefact analysis focuses on analyzing authentication and authorization artefacts as well as configuring and versioning artefacts. Furthermore, it analyzes activity reconstruction and data recovery artefacts, which makes up the largest grouping of artefacts.
5.	[68]	Financial and Business Data Analysis	Deep analysis of the account data and other related business data should be executed. It is used to reveal fraudulent transactions.
6.	[1]	Restoration and Searchability	Recreation of data that have been (partially) destroyed or only partially recovered.
7.	[3]	Artefact Analysis	All data acquired through incident verification and collection phases are consolidated and analyzed.
8.	[22]	Forensic Analysis	Forensic analysis involves temporal detection, the determination of the time. It also involves spatial detection, the determination of where the location of the data in the database was altered.
9.	[42]	Analysis anti-forensic attacks, Analysis database attack	Reconstruct and analyze database attacks to reveal who the attacker is., when the attack happened, where it happened and how it happened.
10.	[6]	Reconstruction process	Identifying changes made to a database, identifying who may be responsible for the changes, confirming what we expect to see in the database, and determining the timeline of events in the database.
11.	[5]	Recovering Database Schema	Discovering the original schema, structure identifiers, identifies pages from the same structure, and discover other components of the schema.
12.	[28]	Analysis stage	Compare the values of metadata of each file before the database operation and after the operation, identify changes in metadata values after the operation, and identify the files that have been affected.

models are specific in nature and do not cover the entire DBFI field. For example, an investigation process model was proposed by [15] to discover information on the operations performed on an Oracle database. The SQL Server Forensic Analysis Methodology was proposed by [2] to collect and analyze evidence from MSSQL server databases.

This methodology is comprised of four phases: investigation preparation, incident verification, artefact collection, and artefact analysis. The Database Server Detection Process Model was proposed by [19] to detect database servers and collect data. It consists of three phases: server detection, data collection, and the investigation of collected data. However, it cannot deal with volatile artefacts.

A framework was proposed by [3] to deal with the forensic analysis of MySQL server databases. This framework consists of four main investigation processes: identification, artefact collection, artefact analysis, and final forensic report. The identification process is used to identify databases, binary logs, log files, and text files on database servers using MySQL Utility programs. The identification process is used to disconnect database servers from the network to capture volatile data, prepare an forensic environment, and allow for the use forensic techniques to move captured data. The artefact collection process collects data from files identified in the previous processes. The artifact analysis process analyzes data acquired through the identification and collection processes. This allows notable events to be identified and added to an investigation timeline to assist examiners in classifying action patterns and related database actions that may not be sequentially logged within collected log files. Finally, the final forensic report process includes all the investigation steps and tools used during the entire investigation process.

A framework was proposed by [47] to provide a greater understanding of volatile and delicate nature of database forensic artifacts to legal groups and non-technical users dealing with database violations. It consists of six investigation processes: incident reporting, examination preparation, physical & digital examination, documentation & presentation, post examination, and post examination analysis. The incident reporting process is used to capture a database incident through user reports, system audits, or triggered events. An initial report is prepared and the examination proceeds to the next process. The examination preparation process prepares the database forensic tools used to identify database systems, isolate networks, and freeze crime scenes. The physical & digital examination process is used to collect and preserve physical and digital evidence. It consists of two processes: physical examination and digital examination. The physical examination process is used to capture and preserve evidence from a physical crime scene. It consists of several activities: preservation, survey, documentation, search, collection, reconstruction, and report. The digital examination process, on the other hand, begins by preserving the digital crime scene and is based on the reports obtained from the physical examination process as well as dead (offline) or live (online) analysis. After the survey is completed and documented, volatile evidence is collected, followed by non-volatile evidence. The digital examination process includes numerous activities such as: preservation, survey, documentation, search, collection, volatile collection, non-volatile collection, evidence validation, digital crime scene reconstruction, and report. After evidence collection, data is analyzed, evidence is validated, and the entire crime scene is reconstructed using temporal (when), relational (who, what, and where) and functional (how) analysis and a report is generated. The fourth process of this framework is the documentation & presentation process, which combines all investigation reports before separating them into technical and legal reports. Finally, the post examination process archives and secures data and evidence while the post examination analysis process returns equipment and collected evidence to their rightful owners. However, this framework is fully based on existing DF investigation models.

Additionally, a database forensic analysis model was proposed by [5] to reconstruct database activities through internal structure carving, via reconstructing volatile artefacts, and recovering database schema. However, this model is specific for reconstructing volatile artifacts only. Forensic methodology to test the tracks of any storage engine on the internal files of a DBMS has been proposed by [28]. This will help in flagging and listing files that have been affected by a particular database operation. These files can then be analyzed to interpret the actual content to see the nature of change to determine the worth of the evidence. This model provided three investigation stages, namely preliminary analysis, execution, and analysis; however, it is specific for MySQL database system only. Reference [40] proposed a reconstruction process model for rebuilding database content from a database image without using any log or system metadata. A special forensic tool called ''DBCarver'' has been proposed for this task that permits reconstruction of database storage. A forensic tamper detection of sensitive data model in MSSQL Server was presented by [21]. Despite the SQL Server provides authentication and authorization mechanisms, it fails to protect the system from malicious attacks from insiders. Accordingly, this model not only provides tamper-prevention measures, but shows how inside tampering can be identified and detected and how to localize the affected data. This model provides detection process. A discovering model to detect the covert database systems in an organization was proposed by [46]. It consists of several digital forensic techniques. The covert database systems in the organization are built in order to hide evidence about any illegal activities in the organization including company. Therefore, this model presented ''Detection process'' that is used to detect the hidden evidence about any illegal activities. Reference [23] proposed a reconstruction model to enables forensic investigators to determine whether data of interest was present in a database at an earlier time even though several database modifications may have been performed since that time. Therefore, this model proposed the ''Reconstruction'' process along with a database reconstruction algorithm to determine whether data of interest was present in a database at an earlier time. A collection process model has been proposed by [70] to locate key evidence and maintain the integrity and reliability of the evidence. This model proposed a ''Collection'' process along with database forensic methods. The method segments a DBMS into four abstract layers (data model layer, data

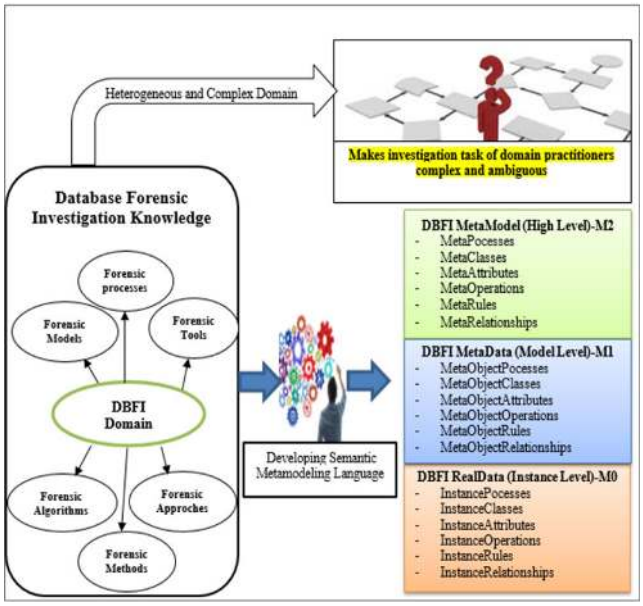

FIGURE 2. DBFI field knowledge gap and proposed research solution.

dictionary layer, application schema layer, and application data layer) that separate the various levels of DBMS metadata and data.

Through this survey, it was obvious that the DBFI domain lacks a structured and unified model/framework that facilitates the management, sharing, and reuse of DBFI domain knowledge. Fig. 2 displays the DBFI field knowledge gap and the proposed research solution. The DBFI field has suffered from several issues that have resulted in it becoming a heterogeneous, confusing, and unstructured domain. Examples of these issues as shown in Fig. 2 and include a variety of database system infrastructures, the multidimensional nature of database systems, and field knowledge effectively being scattered in all directions. The variety of database system infrastructures and their multidimensional natures has only allowed the DBFI field to address specific incidents as Database Management System (DBMS) has a specific forensic investigation model/approach. Consequently, different concepts and terminologies in terms of forensic investigation processes and the scattering of field knowledge has produced challenges for DBFI investigators and practitioners. This knowledge (models, processes, techniques, tools, frameworks, methods, activities, approaches, and algorithms) is neither organized nor structured. Furthermore, it is widely dispersed across the Internet, books, journals, conferences, online databases, dissertations, reports, and organizations. The proposed solution for this issue is to develop a comprehensive/high abstract level framework that facilitates the management, sharing, and reuse of DBFI field tasks and activities among domain practitioners.

VI. RESULTS AND DISCUSSION

This section discusses and proposes solutions for the discovered research gaps. Specifically, three directional solutions are recommended to resolve the three limitations discussed in Section 5 Items 1-3.

A. TO PROPOSE COMMON INVESTIGATION PROCESSES AND CONCEPTS FOR THE DBFI FIELD

As discussed in Section 5 Item 3: there are four investigation DBFI process perspectives: preparation, collection, analysis, and presentation. The main goal of the proposed investigation processes is to address the redundant and irrelevant investigation processes present in the DBFI field that cause confusion among domain practitioners. Each proposed investigation process will contain all the benefits of previous models.

For example, the proposed investigation process for the preparation perspective process illustrated in Table 3 contains all of the investigation tasks, activities, and methods of existing investigation processes. It includes the preparation of investigation environments, the identification of forensic tools, the seizing of investigation sources (volatile and nonvolatile artefacts), the capturing of investigation sources, the isolation of suspect database servers, the isolation of suspect users, the conduction of interviews, the preparation of incident responses, the identification of highly qualified experts, the verification of database incidents (compromised, destroyed, or modified), the documentation of investigation tasks, and the preservation of investigation sources. For example, the Suspension of Database Operation process in [61] isolates database servers from users to capture database activities, while the Verification and System Description processes in [20] verifies and checks database incidents, isolates database servers, confirms incidents, and documents system information such as name, serial number, operating system, functions, and physical description. The Identification process in [56] deals with disconnecting database servers from a network in order to capture volatile data. The purpose of the Investigation preparation and Incident verification processes in [2] is to identify and verify database incidents through a preliminary investigation, prepare forensic workstations, prepare forensic toolkits to respond to incidents, and disconnect the database server. The Database Connection Environment proposed [57] prepares the investigation environment and obtains necessary permission to access and execute required commands. The Table Relationship Search and Join process extracts tablespaces in a database, selects targets, selects tables to store investigation data, and repeatedly checks other table fields. The Data Acquirement with Seizure and Search Warrant secures the location of evidence and extracts evidence that relates to a crime or incident [62]. Another process of interest is Server Detection, which detects the servers running a database system. This process includes grasping the overall network circumstances and topologies inside a company to identify and detect victimized database servers [19].

The proposed collection perspective process is the second investigation process that will combine all investigation tasks, activities, and methods from existing investigation processes as shown in Table 4. It covers the collection of volatile

artefacts, nonvolatile artefacts, volatile data, and nonvolatile data; the protection of authentic gathered data (hash data, backed up data, copied data, and imaged data), and the transfer of collected data. For example, the Data Collection process proposed by [61] assembles data, metadata, and intruder activities. Similar processes were proposed by [20] such as the Evidence Collection process that collects evidence from victimized database servers and the Evidence Collection process proposed by [56] that collects volatile data from compromised database servers. The Artefact Collection process proposed by [2] collects volatile and non-volatile MSSQL Server database artefacts such as log files, data files, data caches, and transaction logs. The Data Extraction process proposed by [57] extracts relationship data that connects columns in database tables. The Begging of Investigation process proposed by [62] extracts fraud data from a database server. The Metadata Extraction process proposed by [1] extracts metadata on database dimensions and determines who was authorized to perform a certain action. The Data Collection process presented by [19] is subdivided into two stages that collect partial field and entire files. The Artefact Collection process proposed by [3] collects and extracts database files and metadata from compromised MySQL Server databases.

Additionally, the third proposed investigation process, the analysis perspective process, contains most of the activities and tasks of existing investigation processes as shown in Table 5. It consists of the examination of collected data (checking of authentic data), the reconstruction of timeline events, event filtering, event analysis, and evidence production.

Finally, the last proposed investigation process, the presentation perspective process, covers similar existing investigation processes such as the presentation of evidence, decision making, and inviting the offender and victim to the court.

B. TO DEVELOP A SEMANTIC METAMODELING LANGUAGE THAT MANAGES AND SHARES DBFI FIELD KNOWLEDGE

A sematic metamodeling language must has the ability to quickly design and integrate semantically rich languages in a unified way [63]. Metamodeling was used to accomplish this. A metamodel is a language model that captures important properties and features. These include supported language concepts, textual and/or graphical syntax, and semantics (what the models and programs written in the language mean and how they behave) [63]. Metamodels unify languages because the same metamodeling language is used for each case. Therefore, a metamodeling language is a Unified Modelling Language (UML) as proposed by the Object Management (OMG) group [64]. The UML is a visual language that is rich with graphical notations and a comprehensive set of diagrams and elements. It includes several languages that describe different aspects of a system such as class diagrams for structural modelling or activity diagrams for behavioral modelling. In this paper, a semantic metamodeling language is suggested to solve the interoperability, heterogeneity, and complexity of the DBFI field. The interoperability of the DBFI domain can be solved by developing a semantic metamodeling language (metamodel). A metamodel is a model that explains another model. Metamodels can specify concepts, attributes, operations, and associations to a specific domain [65], [66]. A Metamodel is the precise definition of modeling elements (concepts, attributes, operations, associations, and rules) needed to create semantic models [67] and domain models. A metamodel is thus a prescriptive/description model of a semantic modeling language. It is used to solve the ambiguity and heterogeneity of complex domains through the generation of solution models. Metamodels have three levels (M0, M1, and M2) as illustrated in Fig. 2. Concepts below M2 belong to M1 or M0. Any concept above M0 can be instantiated at M1 or M2. The M2-level is reserved for metamodel components, including the explanation of metadata construction and semantics as illustrated by UML concepts (classes, attributes, operations, relations, and notations). The M1-level represents the model level, including the metadata that defines data in the information level. Finally, the lowest level (M0) is dedicated to user models and is also named the information level (user data).

C. TO DEVELOP A REPOSITORY FOR THE DBFI FIELD

The main purpose of a repository is to store and retrieve DBFI field knowledge in an easy way. Various DBFI file experiences can be combined into a single repository that can then be reused to facilitate and support DBFI field decisions. The created repository will be a collection of organizational, operational, planning, logistics, and administrative procedures and policies that have been executed by different DBMS through investigation processes. The benefits of the proposed/developed repository to domain practitioners are: i) the simplification of common communications between different DBFI field practitioners through a common representation layer that includes all processes, concepts, tasks, and activities that must exist in the DBFI field; ii) the provision of guidelines and model development processes that assist domain practitioners in managing, sharing, and reusing DBFI field knowledge; iii) enabling domain practitioners to easily create a new solution model through electing and combining sets of concept elements (attribute and operations) based on their own model requirements; and iv) enabling domain practitioners to quickly gain access and reuse relevant DBFI field knowledge.

VII. CONCLUSION

A total of 40 DBFI process models were reviewed in this article. Process model researchers have used different approaches with different stages/phases and terminology. Most DBFI process models are specific and focus on specific RDBMS events, so they only provide low-level details. Furthermore, none of the studied DBFI process models can be called 'standardised' as each model has a different perspective. This paper contributes to the DBFI field by presenting

a broad literature review that will assist field researchers in comprehending DBFI. This study studies all existing DBFI works, discuss the issues and drawbacks of the DBFI field, and suggest some solutions for the discovered limitations. The following are a few ideas for future works in the DBFI field: 1) the proposal of a generic DBFI process/model for the DBFI field; 2) the development of a semantic metamodeling language that structures, manages, organizes, shares, and reuses DBFI knowledge; and 3) the development a DBFI repository for the storage and retrieval of DBFI field knowledge.

REFERENCES

[1] M. S. Olivier, "On metadata context in database forensics," *Digit. Invest.*, vol. 5, nos. 3–4, pp. 115–123, Mar. 2009.
[2] K. Fowler, *SQL Server Forenisc Analysis*. London, U.K.: Pearson, 2008.
[3] H. K. Khanuja, "A framework for database forensic analysis," *Comput. Sci. Eng., Int. J.*, vol. 2, no. 3, pp. 27–41, Jun. 2012.
[4] H. Q. Beyers, "Database forensics: Investigating compromised database management systems," Univ. Pretoria, Pretoria, South Africa, Tech. Rep., 2014.
[5] J. Wagner, A. Rasin, and J. Grier, "Database forensic analysis through internal structure carving," *Digit. Invest.*, vol. 14, pp. S106–S115, Aug. 2015.
[6] O. M. Adedayo and M. S. Olivier, "Ideal log setting for database forensics reconstruction," *Digit. Invest.*, vol. 12, pp. 27–40, Mar. 2015.
[7] D. Litchfield, "Oracle forensics part 1: Dissecting the redo logs," NGSSoftware Insight Secur. Res., Next Gener. Secur. Softw., Sutton, U.K., Tech. Rep., 2007.
[8] D. Litchfield, "Oracle forensics part 2: Locating dropped objects," NGSSoftware Insight Secur. Res., Kigali, Rwanda, Tech. Rep., 2007.
[9] D. Litchfield, "Oracle forensics: Part 3 isolating evidence of attacks against the authentication mechanism," NGSSoftware Insight Secur. Res., Kigali, Rwanda, Tech. Rep., 2007.
[10] D. Litchfield, "Oracle forensics part 4: Live response," NGSSoftware Insight Secur. Res., USA, Tech. Rep., 2007.
[11] D. Litchfield, "Oracle forensics part 5: Finding evidence of data theft in the absence of auditing," NGSSoftware Insight Secur. Res., Next Gener. Secur. Softw., Sutton, U.K., Tech. Rep., 2007.
[12] D. Litchfield, "Oracle forensics part 6: Examining undo segments, flashback and the oracle recycle bin," NGSSoftware Insight Secur. Res., Next Gener. Secur. Softw., Sutton, U.K., Tech. Rep., 2007.
[13] D. Litchfield, "Oracle forensics part 7: Using the Oracle system change number in forensic investigations," NGSSoftware Insight Secur. Res., Kigali, Rwanda, Tech. Rep., 2007.
[14] S. Tripathi and B. Baburao Meshram, "Digital evidence for database tamper detection," *J. Inf. Secur.*, vol. 3, no. 2, pp. 113–121, 2012.
[15] D. Wong and K. Edwards, "System and method for investigating a data operation performed on a database," Google Patents 10/879 466, Dec. 29, 2004.
[16] P. M. Wright and A. J. Ullrich, "Using oracle forensics to determine vulnerability to zero day exploits," InfoSec Reading Room, SANS Inst., Bethesda, MD, USA, Tech. Rep., 2007.
[17] *Oracle Database Forensics Using LogMiner*, P. M. Wright-GSEC and G. GCFW, New York, NY, USA, 2005.
[18] O. M. Fasan and M. S. Olivier, "On dimensions of reconstruction in database forensics," in *Proc. WDFIA*, 2012, pp. 97–106.
[19] N. Son, "The method of database server detection and investigation in the enterprise environment," in *Proc. FTRA Int. Conf. Secure Trust Comput., Data Manage., Appl.* Cham, Switzerland: Springer, 2011, pp. 164–171.
[20] *A Real World Scenario of a SQL Server 2005 Database Forensics Investigation*, Kevvie Fowler, GCFA Gold, CISSP, MCTS, MCSD, MCDBA, MCSE, Inf. Secur. Reading Room Paper, SANS Inst., Rockville, MD, USA, Tech. Rep., 2007.
[21] A. Basu. (2006). *Forensic Tamper Detection in SQL Server*. [Online]. Available: http://www.sqlsecurity.com/chipsblog/archivedposts
[22] H. K. Khanuja and D. D. S. Adane, "Forensic analysis of databases by combining multiple evidences," *Int. J. Comput. Technol.*, vol. 7, no. 3, pp. 654–663, Jun. 2013.
[23] O. M. Fasan and M. Olivier, "Reconstruction in database forensics," in *Advances in Digital Forensics VIII*. Cham, Switzerland: Springer, 2012, pp. 273–287.
[24] P. Frühwirt, M. Huber, M. Mulazzani, and E. R. Weippl, "InnoDB database forensics," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 1028–1036.
[25] P. Fruhwirt, P. Kieseberg, S. Schrittwieser, M. Huber, and E. Weippl, "InnoDB database forensics: Reconstructing data manipulation queries from redo logs," in *Proc. 7th Int. Conf. Availability, Rel. Secur.*, Aug. 2012, pp. 625–633.
[26] P. Frühwirt, P. Kieseberg, S. Schrittwieser, M. Huber, and E. Weippl, "InnoDB database forensics: Enhanced reconstruction of data manipulation queries from redo logs," *Inf. Secur. Tech. Rep.*, vol. 17, no. 4, pp. 227–238, May 2013.
[27] A. C. Lawrence. (2014). *Forensic Investigation of MySQL Database Management System*. [Online]. Available: https://scholarworks.uark.edu/cgi/viewcontent.cgi?referer=http://scholarworks.uark.edu/cgi/viewcontent.cgi?article=1007&context=csceuht&httpsredir=1&article=1007&context=csceuht
[28] J. O. Ogutu, "A methodology to test the richness of forensic evidence of database storage engine: Analysis of MySQL update operation in InnoDB and MyISAM storage engines," Univ. Nairobi, Nairobi, Kenya, Tech. Rep., 2017.
[29] W. K. Hauger and M. S. Olivier, "The state of database forensic research," in *Proc. Inf. Secur. South Afr. (ISSA)*, Aug. 2015, pp. 1–8.
[30] W. K. Hauger and M. S. Olivier, "The state of database forensic research," Inf. Secur. South Africa (ISSA), Tech. Rep., Aug. 2015, pp. 1–8.
[31] W. G. Kruse, II and J. G. Heiser, *Computer Forensics: Incident Response Essentials*. London, U.K.: Pearson, 2001.
[32] G. Palmer, "A road map for digital forensic research," in *Proc. 1st Digit. Forensic Res. Workshop*, Utica, NY, USA, 2001, pp. 27–30.
[33] A. C. Bogen, "Selecting keyword search terms in computer forensics examinations using domain analysis and modeling," Mississippi State Univ., Mississippi, MS, USA, Tech. Rep. DTR-T001-01, 2006.
[34] M. F. M. Buang and S. M. Daud, "A Web-based KM system for digital forensics—Knowledge sharing capability," in *Proc. Int. Conf. Multimedia Comput. Syst. (ICMCS)*, May 2012, pp. 528–533.
[35] W. A. Jansen and A. Delaitre, *Mobile Forensic Reference Materials: A Methodology and Reification*. Gaithersburg, MD, USA: U.S. Department of Commerce, National Institute of Standards and Technology, 2009.
[36] E. Casey, *Handbook of Digital Forensics and Investigation*. New York, NY, USA: Academic, 2009.
[37] C. P. Grobler, C. P. Louwrens, and S. H. von Solms, "A framework to guide the implementation of proactive digital forensics in organisations," in *Proc. Int. Conf. Availability, Rel., Secur. (ARES)*, Feb. 2010, pp. 677–682.
[38] R. Susaimanickam, "A workflow to support forensic database analysis," Murdoch Univ., Perth, WA, Australia, Tech. Rep., 2010.
[39] J. Yoon, D. Jeong, C.-H. Kang, and S. Lee, "Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study," *Digit. Invest.*, vol. 17, pp. 53–65, Jun. 2016.
[40] J. Wagner, "Database forensic analysis with DBCarver," in *Proc. CIDR*, 2017, pp. 1–10.
[41] P. Frühwirt, P. Kieseberg, K. Krombholz, and E. Weippl, "Towards a forensic-aware database solution: Using a secured database replication protocol and transaction management for digital investigations," *Digit. Invest.*, vol. 11, no. 4, pp. 336–348, Dec. 2014.
[42] H. Khanuja and S. S. Suratkar, "Role of metadata in forensic analysis of database attacks," in *Proc. IEEE Int. Advance Comput. Conf. (IACC)*, Feb. 2014, pp. 457–462.
[43] P. M. Wright, "Oracle database forensics using LogMiner," in *Proc. SANS Inst.*, Jun. 2004, pp. 336–348.
[44] L. Pasquale, D. Alrajeh, C. Peersman, T. Tun, B. Nuseibeh, and A. Rashid, "Towards forensic-ready software systems," in *Proc. 40th Int. Conf. Softw. Eng. New Ideas Emerg. Results (ICSE-NIER)*, 2018, pp. 9–12.
[45] A. M. R. Al-Dhaqm, S. H. Othman, S. Abd Razak, and A. Ngadi, "Towards adapting metamodelling technique for database forensics investigation domain," in *Proc. Int. Symp. Biometrics Secur. Technol. (ISBAST)*, Aug. 2014, pp. 322–327.
[46] G. T. Lee, S. Lee, E. Tsomko, and S. Lee, "Discovering methodology and scenario to detect covert database system," in *Proc. Future Gener. Commun. Netw. (FGCN)*, 2007, pp. 130–135.
[47] J. Azemović and D. Mušić, "Efficient model for detection data and data scheme tempering with purpose of valid forensic analysis," in *Proc. Int. Conf. Comput. Eng. Appl. (ICCEA)*, 2009, pp. 322–327.

[48] R. T. Snodgrass, S. S. Yao, and C. Collberg, "Tamper detection in audit logs," in *Proc. 13th Int. Conf. Very Large Data Bases*, vol. 30, 2004, pp. 1–34.
[49] E. C. Cankaya and B. Kupka, "A survey of digital forensics tools for database extraction," in *Proc. Future Technol. Conf. (FTC)*, Dec. 2016, pp. 1014–1019.
[50] J. Wagner, A. Rasin, B. Glavic, K. Heart, J. Furst, L. Bressan, and J. Grier, "Carving database storage to detect and trace security breaches," *Digit. Invest.*, vol. 22, pp. S127–S136, Aug. 2017.
[51] J. Yoon and S. Lee, "A method and tool to recover data deleted from a MongoDB," *Digit. Invest.*, vol. 24, pp. 106–120, Mar. 2018.
[52] A. Al-Dhaqm, S. Razak, and S. H. Othman, "Model derivation system to manage database forensic investigation domain knowledge," in *Proc. IEEE Conf. Appl., Inf. Netw. Secur. (AINS)*, Nov. 2018, pp. 75–80.
[53] O. M. Adedayo and M. Olivier, "Schema reconstruction in database forensics," in *Proc. IFIP Int. Conf. Digit. Forensics*. Cham, Switzerland: Springer, 2014, pp. 101–116.
[54] R. Chopade and V. K. Pachghare, "Ten years of critical review on database forensics research," *Digit. Invest.*, vol. 29, pp. 180–197, Jun. 2019.
[55] P. M. Wright, "Oracle database forensics using LogMiner," Global Inf. Assurance Certification Paper, SANS Inst., Bethesda, MD, USA, Tech. Rep., 2014.
[56] D. Litchfield, "Oracle forensics part 4: Live response," NGSSoftware Insight Secur. Res., Next Gener. Secur. Softw., Sutton, U.K., Tech. Rep., Apr. 2019.
[57] D. Lee, J. Choi, and S. Lee, "Database forensic investigation based on table relationship analysis techniques," in *Proc. 2nd Int. Conf. Comput. Sci. Appl. (CSA)*, Dec. 2009, Art. no. 5404235.
[58] N. Son, "The method of database server detection and investigation in the enterprise environment," in *Secure and Trust Computing, Data Management and Applications*. Cham, Switzerland: Springer, 2011, pp. 164–171.
[59] A. Al-Dhaqm, S. Razak, S. H. Othman, K.-K.-R. Choo, W. B. Glisson, A. Ali, and M. Abrar, "CDBFIP: Common database forensic investigation processes for Internet of Things," *IEEE Access*, vol. 5, pp. 24401–24416, 2017.
[60] A. Ahmed, M. Kleiner, L. Roucoules, R. Gaudy, and B. Larat, "Model-based interoperability solutions for the supervision of smart gas distribution networks," in *Proc. 11th Syst. Syst. Eng. Conf. (SoSE)*, Jun. 2016, pp. 1–5.
[61] D. Wong and K. Edwards, "System and method for investigating a data operation performed on a database," Google Patents 10/879 466, Dec. 29, 2005.
[62] J. Choi, K. Choi, and S. Lee, "Evidence investigation methodologies for detecting financial fraud based on forensic accounting," in *Proc. 2nd Int. Conf. Comput. Sci. Appl.*, Dec. 2009, Art. no. 5404202.
[63] T. Clark, P. Sammut, and J. Willans, "Applied metamodelling: A foundation for language driven development (third edition)," 2015, *arXiv:1505.00149*. [Online]. Available: http://arxiv.org/abs/1505.00149
[64] Z. Cheng, "Formal verification of relational model transformations using an intermediate verification language," Nat. Univ. Ireland Maynooth, Maynooth, Ireland, Tech. Rep., 2012.
[65] I. Poernomo, "A type theoretic framework for formal metamodelling," in *Architecting Systems with Trustworthy Components*. Cham, Switzerland: Springer, 2006, pp. 262–298.
[66] J. Bézivin and O. Gerbé, "Towards a precise definition of the OMG/MDA framework," in *Proc. 16th Annu. Int. Conf. Automated Softw. Eng. (ASE)*, 2001, pp. 273–280.
[67] N. Koch and A. Kraus, "Towards a common metamodel for the development of Web applications," U.S. Patent, 2003.
[68] J. Choi, K. Choi, and S. Lee, "Evidence investigation methodologies for detecting financial fraud based on forensic accounting," in *Proc. 2nd Int. Conf. Comput. Sci. Appl.*, Dec. 2009, Art. no. 5404202.
[69] J. Azemović and D. Mušić, "Methods for efficient digital evidences collecting of business proceses and users activity in eLearning enviroments," in *Proc. Int. Conf. e-Educ., e-Bus., e-Manage. e-Learn.*, Jan. 2010, pp. 126–130.
[70] H. Beyers, M. Olivier, and G. Hancke, "Assembling metadata for database forensics," in *Advances in Digital Forensics VII*. Cham, Switzerland: Springer, 2011, pp. 89–99.
[71] H. Beyers, M. Olivier, and G. Hancke, "Arguments and methods for database data model forensics," in *Proc. WDFIA*. 2012, pp. 139–149.
[72] A. Al-dhaqm, S. Razak, S. H. Othman, A. Ngadi, M. N. Ahmed, and A. A. Mohammed, "Development and validation of a database forensic metamodel (DBFM)," *PLoS ONE*, vol. 12, no. 2, Feb. 2017, Art. no. e0170793.
[73] A. Al-Dhaqm, S. Razak, S. H. Othman, K.-K.-R. Choo, W. B. Glisson, A. Ali, and M. Abrar, "CDBFIP: Common database forensic investigation processes for Internet of Things," *IEEE Access*, vol. 5, pp. 24401–24416, 2017.
[74] D. A. Flores and A. Jhumka, "Implementing chain of custody requirements in database audit records for forensic purposes," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 675–682.
[75] R. Bria, A. Retnowardhani, and D. N. Utama, "Five stages of database forensic analysis: A systematic literature review," in *Proc. Int. Conf. Inf. Manage. Technol. (ICIMTech)*, Sep. 2018, pp. 246–250.

ARAFAT AL-DHAQM was born in Mukhayliya, Lahj, Yemen, in 1976. He received the B.S. degrees in information system from the University Technology of Iraq, in 2002, and the master's and Ph.D. degrees in computer science (digital forensic, and information security) from University Technology Malaysia, in 2013 and 2018, respectively, where he is currently pursuing the Ph.D. degree, under the supervision of Associate Professor Dr. S. Razak.

SHUKOR ABD RAZAK is currently an Associate Professor with Universiti Teknologi Malaysia. His research interests are on the security issues for mobile ad hoc networks, mobile IPv6, vehicular ad hoc networks, and network security. He also actively conducts several types of research in digital forensic investigation, wireless sensor networks, and cloud computing. He is the author and coauthor for many journals and conference proceedings at national and international levels.

SITI HAJAR OTHMAN received the Ph.D. degree from the University of Wollongong, Australia. She is currently a Senior Lecturer with the Department of Computer Science, University Technology Malaysia, Malaysia. Her current research interests include conceptual modelling, metamodeling, disaster management, information security, computer forensic, knowledge retrieval, disaster recovery, and business continuity planning.

ABDULALEM ALI received the B.S. degree in computer science from Thamar University, Yemen, in 2004, and the M.S. degree in information security from Universiti Teknologi Malaysia, Malaysia, in 2013, where he is currently pursuing the Ph.D. degree in computer science. His current research interest includes propose high level model (metamodel) for mobile forensics.

FUAD A. GHALEB received the B.Sc. degree in computer engineering from Sana'a University, Yemen, in 2003, and the M.Sc. and Ph.D. degrees in computer science - information security from the Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia, in 2014 and 2018, respectively. He currently works as a Postdoctoral Fellow with the School of computing, Faculty of Engineering. His research interests include vehicular network security, cyber threat intelligence, intrusion detection, data science, data mining, and artificial intelligence.

ARIEFF SALLEH ROSMAN received the Ph.D. degree in Islamic jurisprudence from the International Islamic University Malaysia. He is currently an Associate Professor with the Center for Research in Fiqh Science and Technology, Universiti Teknologi Malaysia. His research interest is in the Islamic jurisprudence related to science and technology issues.

NURAZMALLAIL MARNI received the Ph.D. degree in philosophy and civilizational studies from Universiti Putra Malaysia. He is currently a Senior Lecturer with the Academy of Islamic Civilization, Faculty of Social Sciences and Humanities, Universiti Teknologi Malaysia. His research interests include epistemology, Islamic medicine, and Islamic jurisprudence related to science and technology issues. He is the author and coauthor of many journals and conference proceedings at national and international levels.