Computers, Materials & Continua
DOI:10.32604/cmc.2020.013739
Article

Tech Science Press

A New Database Intrusion Detection Approach Based on Hybrid Meta-Heuristics

Youseef Alotaibi*

Department of Computer Science, College of Computer and Information Systems, Umm Al Qura University, Makkah, 21421, Saudi Arabia
*Corresponding Author: Youseef Alotaibi. Email: yaotaibi@uqu.edu.sa
Received: 19 August 2020; Accepted: 29 September 2020

Abstract: A new secured database management system architecture using intrusion detection systems (IDS) is proposed in this paper for organizations with no previous role mapping for users. A simple representation of Structured Query Language queries is proposed to easily permit the use of the worked clustering algorithm. A new clustering algorithm that uses a tube search with adaptive memory is applied to database log files to create users' profiles. Then, queries issued for each user are checked against the related user profile using a classifier to determine whether or not each query is malicious. The IDS will stop query execution or report the threat to the responsible person if the query is malicious. A simple classifier based on the Euclidean distance is used and the issued query is transformed to the proposed simple representation using a classifier, where the Euclidean distance between the centers and the profile's issued query is calculated. A synthetic data set is used for our experimental evaluations. Normal user access behavior in relation to the database is modelled using the data set. The false negative (FN) and false positive (FP) rates are used to compare our proposed algorithm with other methods. The experimental results indicate that our proposed method results in very small FN and FP rates.

Keywords: Adaptive search memory; clustering; database management system (DBMS); intrusion detection system (IDS); quiplets; structured query language (SQL); tube search

1 Introduction

Data can be considered a significant asset for all modern organizations, which may store a huge amount in their databases and cloud. Database management systems (DBMSs) are used by organizations to control access to these data by both internal and external users. Access must be protected, especially from competitors, and thus security and privacy are considered key issues for organizations to protect their data [1].

The literature describes many threats to databases from both internal and external users. External users can access databases using web applications or computer networks [2]. In addition, the literature reveals various techniques that can be used to protect computer networks from external threats, such as fire-wall

systems and detection models at the network layer. However, external users may still have access to these database resources. They can also use web applications to execute queries on back-end databases, and can hack network systems. Therefore, network security systems have recently used intrusion detection (ID) techniques to protect network resources against irregular behavior [3]. Network intrusion detection systems (IDS) can be considered an open research area because detecting misuse of databases by legitimate users is not possible using IDS. Hence, databases are still vulnerable to internal threats.

As internal users are allowed to bypass fire-wall systems and network security, as such internal threats can be considered more difficult to detect than external threats. Internal users can login to the system and gain access to non-authorized resources [4]. These threats, which cannot be handled using network intrusion systems, mostly aim to affect Structured Query Language (SQL) commands for database systems, or trigger transactions to access non-privileged resources.

Traditional DBMS assign resources to users to control access to database resources. However, they cannot handle user queries or SQL injections that can be defined as an attack in which malicious code is added to strings passed to the SQL server for execution [5]. Inserting code directly into user input variables, which can be concatenated with SQL executions and commands, can be considered a key form of SQL injection.

SQL injections can be considered a key dangerous issue for database systems as legitimate users are able to apply them. Further, DBMS cannot guarantee data privacy and security against these threats; thus a traditional DBMS architecture must be developed to adapt to new threats and ensure high data privacy and security for organizations [6].

IDS can be considered one of the most important parts of any well-secured system as they can detect malicious actions. They can be used to protect network resources and distinguish between malicious and legitimate transactions [7]. They can be considered the most satiable solution for SQL injections and detecting internal threats as they can analyze queries before executing them. They can inform responsible persons in the database system, such as the site security officer (SSO) or database administrator (DBA), who then handle such threats. Moreover, they can trigger actions to handle intrusions [8].

The literature suggests that building profiles for representing users' normal access behavior is the main way to detect malicious transactions [9]. User profiles can be used to describe normal behavior and users who deviate from this behavior are known as intruders. Therefore, clustering algorithms are needed to build profiles and group together users who demonstrate similar interaction behavior. Further, machine learning algorithms can be used to implement the ID task to classify issued user transactions against related user profiles. For higher security and better protection against internal threats, IDS must be added to the DBMS [10].

Only one profile for each role needs to be built, to minimize the number of profiles; the role represents all users who have similar database access behavior. This can help the IDS to easily control and maintain profiles for organizations with large numbers of users.

We can classify organizations into two types when creating an IDS for them: (1) Organizations with role-based access control where one or more roles are assigned for each user; and (2) Organizations without detailed role architecture assigned for each user [11].

In the first type, the role can be considered a package of authorizations for database resources where there is a clear role hierarchy assigned to each user. In this case, a machine learning algorithm is used to permit or deny non-authorized resource use via issued queries, by comparing queries against user-related roles and identifying them as malicious transactions or otherwise.

In the second type, the application's authorizations can be used by organizations as the role for each user. In this case, clustering algorithms in the database log file can be used to cluster users who have similar

interaction behavior in such a profile. Intrusion-free log files, where users' normal behavior guarantees zero intrusions, can be used and then profiles can be used for each user as the role to detect anomalous behavior. Every user has to be mapped into one profile. Finally, similar to the first type, machine learning algorithms can be used to detect malicious transactions.

The literature shows that most existing algorithms have several drawbacks. Therefore, a new secured DBMS architecture using IDS where there is no previous role mapping for users in an organization is proposed in this paper. A simple representation for SQL queries is proposed to enable the clustering algorithm to be easily used. A new clustering algorithm known as tube search with adaptive search memory (TSASM) is applied to database log files to create user profiles that demonstrate the roles for the relevant users. Then, each user query issued is checked against the related user profile using this classifier to determine whether or not the query is malicious. The IDS will stop query execution or report the threat to the responsible person if the query is malicious. A simple classifier based on the Euclidean distance is used. The issued query is transformed to the proposed simple representation using a classifier where the Euclidean distance between the centers and issued query for all profiles is calculated. The IDS considers the issued query to be an authorized query if the representative user profile has the minimum distance with it. Otherwise, the issued query is considered malicious.

A synthetic data set is used for our experimental evaluations. Normal user access behavior for the database is modelled using this data set. The false negative (FN) and false positive (FP) rates are used to compare the proposed method with other methods. The experimental results indicate that the proposed method attains very small FN and FP rates.

The reminder of the paper is organized as follows. Section 2 represents related work on the ID. Section 3 presents the proposed database ID based on a hybrid meta-heuristics algorithm including the basics of the proposed IDS architecture, the proposed simple representation for SQL queries, and the proposed method to build IDS. Section 4 explains the numerical experiments and algorithm evaluations. Section 5 presents conclusions.

2 Related Work

The literature shows that most IDS research has been focused on the networks area. For example, a survey on network-based and host-based IDS was proposed in [12]. It defined corresponding systems characteristics. In addition, a prototype UNIX anomaly detection system was proposed in [13] as a host-based system to monitor host users of computer networks. This system included an automatic anomaly detection element that used a test depending on self-organized maps for testing when user behavior was anomalous. IDS based on an extension of the use of previous attack signatures was proposed in [14].

A payload-based network IDS was proposed in [15], where the normal application payload of network traffic was modelled in fully automatic style. In this method, a profile of byte frequency distribution and its payload standard was first computed. Second, the Mahalanobis distance was used to compare new data similarity against a threshold and create an alert if the new input distance exceeded the threshold.

The literature indicates little research has been undertaken on IDS proposed for database ID. Therefore, a new DBMS architecture is required that involves adapting traditional DBMS architecture by adding an ID layer to handle internal threats.

A new secured DBMS architecture was proposed in [16]. It added IDS to DBMS to detect intruders, while work is maintained by the DBMS during periods of attack. Another IDS approach in real time was proposed in [17]. It focused on exploiting data real-time properties to detect intruders. Unauthorized updated requests can be detected as they do not occur at the right time. This approach has some

drawbacks, such as the use of objects of temporal data in real-time database systems, which have to be updated occasionally as mismatches in the updating period may cause alarm.

Moreover, the detection of misuse in database systems (DEMIDS) approach was proposed in [18]. It was tailored for relational database systems. Log files were used to derive profiles that described the access pattern behavior of database users. The approach used item sets, also known as attribute sets, that referenced together in the query to build the profiles. Users' access could be described by these item sets while measures of distance among them were used to build the user profile. However, building a profile for each user could be considered the main drawback of the approach as this is a complicated task to implement for databases with large numbers of users. Hence, in our proposed approach profiles are built based on queries in log files instead of being based on the database scheme.

Another IDS approach involving the back-end database was proposed in [19,20], and was based on transaction fingerprints. SQL statements were matched against a set of well-known legitimate database transaction fingerprints.

An anomaly-based system was proposed for web databases in [21,22] and another anomaly-detection system approach for relational databases was proposed in [23,24]. It focused on attribute relations in the database scheme. Two systems were used for detection via that approach: (1) One system that focused on attribute reference values in the relations database; and (2) One system that focused on Δ relations to record the history of data value changes for the monitored attributes among two runs of the anomaly detection system. This approach is similar to the DEMIDS approach as both assumed knowledge about the database relations and structure among database objects.

Another approach that focused on web database attacks from users who make a large number of requests was proposed in [25] to handle only database floods. These requests can make a database slower when it interacts with other users. Profiles were used to model access control. In this method, one profile was built for the entire database and another profile was built for each user.

Another IDS approach that depended on information from database log files was proposed in [26,27] to generate profiles that represent database users' typical access. New expression representations for queries in log files—known as the coarse quiplet (c-quiplet), the medium-grain quiplet (m-quiplet), and the fine quiplet (f-quiplet)—were proposed. The profiles were treated as the roles and every user was mapped into a special profile. Moreover, a machine learning algorithm was proposed to classify each user-issued query against the user's mapped profile.

Another database IDS using data-centric IDS was proposed in [28,29]. Profiles were created by data-centric IDS to describe users' normal access behavior. Profiles included statistical data that described the returned results from the execution of users' previous queries. User behavior deviating from normal access behavior in profiles was defined as intrusion behavior. Syntax IDS was used by the data-centric system to handle dynamic databases and command parts where no data are returned from execution, such as delete, update, and insert.

In summary, the literature shows that almost all of the existing approaches have several drawbacks. Thus, a new simple query representation in log files is proposed in this paper. This representation is based on using real values—instead of binary arrays—in database tables, as well as attribute size. If the tables and attributes are large, large memory resources are needed. Further, a new clustering algorithm named TSASM is used in this paper. Role granularity is supported in building profile representations where profiles represent roles instead of users.

3 Proposed Database Intrusion Detection Based on Hybrid Meta-Heuristics Algorithm

The system architecture, data representation, and methodology for the proposed database ID based on the hybrid meta-heuristics algorithm are explained in the following subsections.

3.1 System Architecture

In this paper, the system architecture for the traditional DBMS has been adapted by adding the ID mechanism before the execution process for queries. Thus, the DBMS uses the IDS to analyze each query before execution. Fig. 1 shows the proposed IDS architecture.

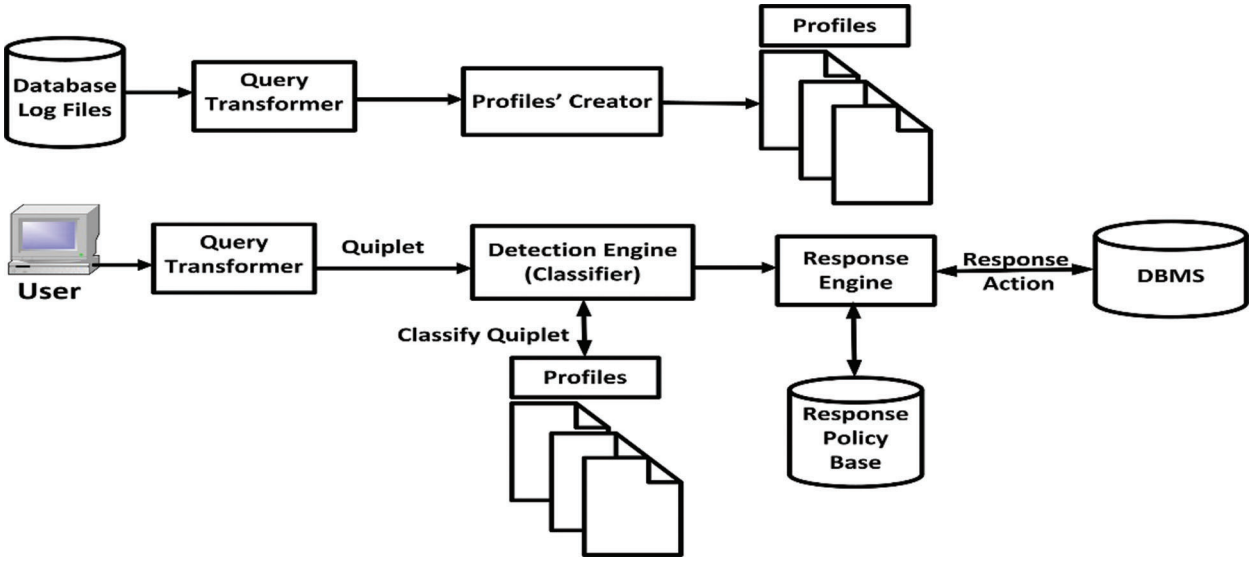

Figure 1: Proposed IDS architecture

In this proposed IDS architecture, the query transformer is first used to work on the log files of the database to transform SQL queries to quiplets. Quiplets can be considered a simple representation of SQL queries. Second, the profile creator, which can be considered a clustering technique, is used to process these quiplets. This cluster technique is used to cluster the data from log files into profiles. All users with similar access behavior are represented using these profiles. Third, the detection engine, which is a machine learning algorithm known as the classifier, is used to determine whether the users' issued queries are malicious or not. Fourth, the response policy base is used to identify the triggered action while a malicious query is detected. Authorized queries can mistakenly raise alarm using the proposed IDS. There are several occasional database user activities that can be classified as anomalous activity, such as fixing occasional problems. As the detection engine may not recognize these occasional activities, they can be considered intrusions. Further, each organization has policies for treating various situations in their response policy base. Sending an alarm to those responsible for the database, such as SSOs or DBAs, is an example of such polices. Last, the IDS uses authorized queries to update the profiles.

3.2 Data Representation

Extracting the information from log files is the most difficult challenge in the proposed approach. A query transformer is used to transform SQL queries into an applicable representation for classification and clustering algorithms. A simple representation for SQL queries, termed the quiplet, is used to build the profile blocks. Cluster algorithms in quiplets are used to build user profiles that detail normal interactions.

Quiplets are built as short structures for SQL queries. Each query is transformed into quiplet form. The user may issue several commands, such as insert, delete, update, and select. The select query can be represented as follows:

SELECT (Target-List),

FROM (Relation-List),

WHERE (Qualification).

where SELECT is the command; Target-List includes all selected fields or attributes; Relation-List represents the field relations in tables, which represent the join conditions between tables; and Qualification represents the conditions that must be fulfilled by returned results.

Each query is divided into five partitions and thus is known as a quiplet, which is the main unit of profiles. The quiplet is expressed as (Q(C,SR,SA,PR,PA)), where the five parts are command, selection relations, selection attributes, propagation relations, and propagation attributes [26]. The f-quiplet representation is used as it can provide detailed information for each query.

The quiplet representation form is applicable for all SQL commands including insert, delete, and update. Like the select command syntax, the delete and update commands have a qualification list and target list. However, the inserted data are placed in the relation and columns to be encoded as the projection relation and projection attributes respectively for the insert command syntax.

The tables and attributes are represented in real numbers in the proposed query representation. The SR and SA, which represent the tables and attributes respectively in the target list, can be considered vector numbers of real value, as can the PR and PA, which represent the tables and attributes respectively in the qualification. They also can store the tables and attributes in the relation list when there is any join condition. Tab. 1 provides an example of a database scheme. An example of transforming from query to quiplet representation is shown in Tab. 2.

Table 1: Database scheme example

Table number	Table name	Column number	Column name
1	R1	1	A
1	R1	2	B
1	R1	3	C
1	R1	4	D
2	R2	5	E
2	R2	6	F
2	R2	7	G
2	R2	8	H

Each SR, SA, PR, and PA real vector number is represented by two computed values known as (1) the number average value in vector; and (2) the number standard deviation in vector. Further, the command text is represented in numbers of real value. A new quiplet (n-quiplet) representation, written as Q(C, SR_{Avg}, SR_{Std}, SA_{Avg}, SA_{Std}, PR_{Avg}, PR_{Std}, PA_{Avg}, PA_{Std}), is used. Tab. 3 provides an example of n-quiplet form representation.

Table 2: Quiplet construction example

SQL command	Quiplet
SELECT R1.A, R1.C,R2.F, R2.DH FROM R1, R2 WHERE R1.B = R2.F	SELECT <1,2> <1,3,6,8> <1,2> <2,6>

Table 3: N-Quiplet construction example

Quiplet	New Quiplet (n-quiplet)
SELECT	1
< 1,2 >	< 1.5,0.707 >
< 1,3,6,8 >	< 4.5,3.109 >
< 1,2 >	< 1.5,0.707 >
< 2,6 >	< 4,2.828 >

3.3 Methodology

There is no predefined role information in the worked database. In addition, there is no role hierarchy assigned to each user; thus, generating profiles can be considered an unsupervised operation. A clustering algorithm is employed to generate profiles from a training data set. These generated profiles are used to represent the roles. Clusters are mapped to users, as every specified user has one cluster that includes the maximum number of records in the log files of that user. One cluster may be allocated to one or more users. Further, a machine learning algorithm is used to classify each issued user query against the user-mapped cluster for the ID process.

A distance function is specified by the clustering algorithm to define the similarities and differences among quiplets. The distance function between any two quiplets is computed using the Euclidean function. The quiplets are represented using the real number where each quiplet has nine values. Suppose that we have two quiplets named Q1(C1, SR_{Avg1}, SR_{Std1}, SA_{Avg1}, SA_{Std1}, PR_{Avg1}, PR_{Std1}, PA_{Avg1}, PA_{Std1}) and Q2(C2, SR_{Avg2}, SR_{Std2}, SA_{Avg2}, SA_{Std2}, PR_{Avg2}, PR_{Std2}, PA_{Avg2}, PA_{Std2}). The distance function (θ) between the Q1 and Q2 quiplets can be calculated using Eq. (1):

$$\theta(Q1, Q2) = [Z1 + Z2 + Z3 + Z4 + Z5 + Z6 + Z7 + Z8 + Z9]^{\frac{1}{2}} \tag{1}$$

where

$$Z1 = (C_1 - C_2)^2$$

$$Z2 = (SR_{Avg1} - SR_{Avg2})^2$$

$$Z3 = (SR_{Std1} - SR_{Std2})^2$$

$$Z4 = (SA_{Avg1} - SA_{Avg2})^2$$

$$Z5 = (SA_{Std1} - SA_{Std2})^2$$

$$Z6 = (PR_{Avg1} - PR_{Avg2})^2$$

$$Z7 = (PR_{Std1} - PR_{Std2})^2$$

$$Z8 = (PA_{Avg1} - PA_{Avg2})^2$$

$$Z9 = (PA_{Std1} - PA_{Std2})^2$$

The TSASM clustering algorithm is used in this paper. As the K-means (K-M) algorithm [30,31] can be easily implemented and showed better performance for real-world problems, it has been used in the TSASM. A tabu search (TS) is used to overcome K-M problems in this algorithm, such as receiving stuck in the local optima. Further, TS strategies such as diversification and intensification are used. The adaptive search memory (ASM) is used in this algorithm to cover the search space effectively and economically.

The Euclidean distance method is used by TSASM to define the similarities and differences among data. The center of each cluster is calculated as the mean value for points belonging to the cluster. For instance, let X denote the set of data objects; n, the number of data objects; and d, the dimensional space Rd—that is, $X = \{x^1, \ldots, x^n\}$, where $x^i \in R^d$, $i = 1, \ldots, n$—and we want to partition set X into K partitions. TSASM tries to obtain the best set of centers $C = \{C^1, \ldots, C^k\}$. Let C^j denote the center of cluster S_j with cardinality $|Sj|$. Then we can calculate C^j, $j = 1,\ldots,K$ according to the following equation:

$$C^j = \frac{1}{|S_j|} \sum_{x \in S_j} x \tag{2}$$

To measure the goodness of different sets of centers, the objective function $f(C)$ can be calculated for each set of centers using the minimum sum of squares [32,33], defined by:

$$f(C) = \frac{1}{n} \sum_{i=1}^{n} \min_{j = 1, \ldots, K} \|C^j - x^i\|^2 \tag{3}$$

where k corresponds to the Euclidean norm. The TSASM output is the K set of centers C^j, $j = 1,\ldots,K$, which partitions the data into K clusters. These clusters can be considered profiles for users.

Each user is mapped to their representative profile once it has been created using the clustering algorithm. A representative profile can be considered a profile that includes the maximum number of user-submitted queries. The proposed ID mechanism can be considered a simple classifier as it uses the θ between the issued query and the center of the mapped profile. The classifier is detected for both authorized and non-authorized queries. The query is used to update the profile and re-compute its center if it is an authorized query. Otherwise, an action is predefined in the response policy base using IDS triggers.

4 Numerical Experiments

A number of numerical experiments have been performed on a synthetic data set to verify the effectiveness of our proposed IDS algorithm. The data set is synthetically created to simulate users' access patterns in the real world. A zipf probability distribution function (PDF) is used to model the non-uniform user access and is defined for the random variable X using Eq. (4):

$$Zipf_{(X,N,s)} = 1/x^s \sum_{i=1}^{N} 1/i^s \tag{4}$$

where N is the number of elements and s is the parameter that characterizes the distribution. The cumulative density function for the zipf distribution when N is equal to 10 with different values of s is shown in Fig. 2.

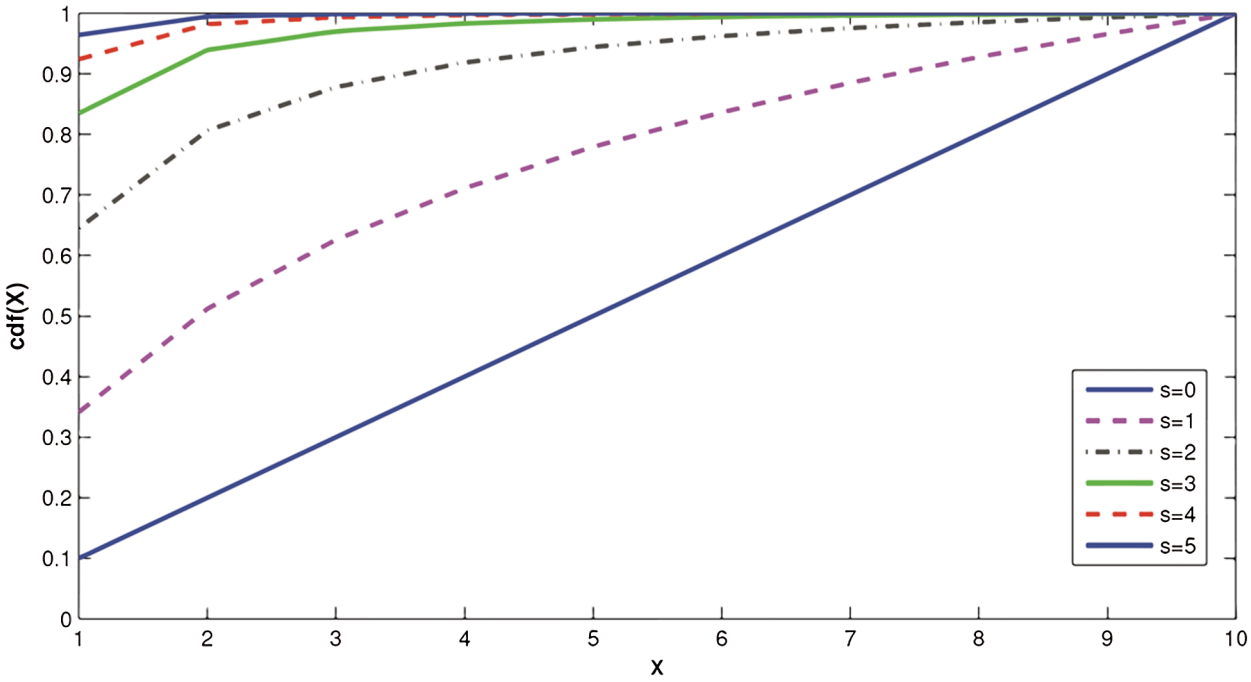

Figure 2: Zipf PDF when $N = 10$

In addition, uniform and multinomial PDFs are used in this paper. The multinomial distribution function can be considered a generalization of the binomial distribution, which is the probability distribution for the number of successes in n independent Bernoulli trials with the same probability of success in every trial. The analog of the Bernoulli distribution can be considered a categorical distribution in the multinomial distribution, where there are n independent trials and every trial can result in one of a fixed finite number K of possible outcomes through probabilities $P1,\ldots, Pk$, for $P_i \geq 0$ *for* $i = 1, \ldots, k$ and:

$$\sum_{i=1}^{k} P_i = 1 \tag{5}$$

Suppose Xi shows the number of times outcome number i was observed over the n trials. The vector $X = (X1, \ldots, Xk)$ will follow the multinomial distribution based on parameters n and p, where $p = (p1, \ldots, pk)$.

4.1 Data Set

A synthetic data set is generated to verify our proposed method by modelling real-world database log files. Each role r has probability $P(r)$ of appearing in the log file. Each role r has five components with their own probability of appearing in the command. The probability for all necessary query components to appear is specified by the data set generator. Each command c in role r has probability $P(c|r)$ of appearing. Each table t has probability $P(Pt|r,c)$ of appearing in role r in command c, which can be applied to both propagation and selection tables. Moreover, each attribute a belonging to table t $(a \in t)$ has probability $P(Pa|r,t,c)$ of appearing in role r in command c, which can be applied for both propagation and selection attributes.

Malicious queries are created to simulate both internal and external threats. They are created from the same probability distribution via a different role number. For instance, when the role information related to a normal query is equal to 1, the role is simply changed to any role other than 1 to make the query anomalous.

For our proposed method used to create the synthetic data set in this paper, the database scheme includes 20 tables and 10 columns in every table. It has nine roles, as shown in Tab. 4. Query submissions for the roles are governed based on the PDF zipf($N = 9$, $s = 1$). The first six roles can submit only selected queries as they are read-only. The last three roles can submit all commands including insert, update, delete, and select as they are read–write roles. They submit the commands insert, update, delete, and select with probabilities 0.1, 0.7, 0.1, and 0.1 respectively. The data set is generated with cardinality of 1000 queries.

Table 4: Description of data set roles

Role No.	Table access	Col access
1	Zipf(1-5,s)	Zipf(1,10)
2	Zipf(6-10,s)	Zipf(1,10)
3	Zipf(11-15,s)	Zipf(1,10)
4	Zipf(16-20,s)	Zipf(1,10)
5	Zipf(1-10,s)	Zipf(1,10)
6	Zipf(11-20,s)	R_Zipf(1,10)
7	Zipf(1-20,s)	Zipf(1,10)
8	Zipf(1-20,s)	R_Zipf(1,10)
9	Uniform(20)	Uniform(10)

4.2 Results

The experimental results for our proposed algorithm are compared with the IDS results, which use three query representations as follows: (1) m-quiplet; (2) c-quiplet; and (3) f-quiplet.

The single query of the database log file is represented by a c-quiplet that includes five fields: SQL-CMD, PROJ-RELCOUNTER, PROJ-ATTR-COUNTER, SEL-REL-COUNTER, and SEL-ATTR-COUNTER. The SQL-CMD field is symbolic and links to the issued SQL command. The PROJ-RELCOUNTER and PROJ-ATTR-COUNTER fields are numerical and link to the number of relations and attributes involved in the SQL query projection part. The SEL-REL-COUNTER and SEL-ATTR-COUNTER fields are numerical and link to the number of relations and attributes that involved in the SQL query selection part.

The single query of the database log file is represented by the m-quiplet, which includes five fields: SQL-CMD, PROJ-REL-BIN[], PROJ-ATTR-COUNTER[], SELRELBIN[], and SEL-ATTR-COUNTER[]. The SQL-CMD field is symbolic and links to the issued SQL command. The PROJ-REL-BIN[] field is a binary vector of size represented as a bit, which equals the number of relations in the database. The bit is set to 1 at position i when the i relation in the SQL query is projected. The PROJ-ATTR-COUNTER [] field is a vector of size equal to the number of relations in the database, where its i element links to the attribute number of the i relation projected in the SQL query. SELRELBIN[], and SEL-ATTR-COUNTER[] have the same vectors as PROJ-REL-BIN[] and PROJ-ATTR-COUNTER[]. However, the information is kept in the former—linked to the SQL query selections instead of the SQL query projections.

A detailed representation of the log query is represented by the f-quiplet, which includes five fields: SQL-CMD, PROJ-REL-BIN[], PROJ-ATTR-BIN[][], SELRELBIN[], and SELATTR-BIN[][]. The SQL-CMD field is symbolic and links to the issued SQL command. The PROJ-REL-BIN[] field is a binary vector of size presented in bits, and equal to the number of relations in the database. The bit is set to 1 at position *i* when the *i* relation in the SQL query is projected. PROJ-ATTR-BIN[][] is a vector of size *n* where *n* is the number of relations in the database. PROJ-ATTR-BIN[i][j] is equal to 1 when the projects of SQL query *j* attribute of *i* relation. Otherwise, it equals 0. The SELRELBIN[] field is a binary vector equal to 1 in its *i* position when the *i* relation is used in the preamble of a SQL query. SELATTR-BIN[][] is a vector of size *n* where *n* is the number of relations in the database. The SELATTR-BIN[i][j] is equal to 1 when the references of SQL query *j* attribute of *i* relation in the predicated query. Otherwise, it equals 0.

The SQL command corresponding to the select statement is shown in Tab. 5 with its representation based on all three quiplet types. A database scheme that includes two relations named R1 = A, B,C, D and R2 = E, F,G, H is shown as an example.

Table 5: Three examples of quiplet constructions

SQL command	c-quiplet	m-quiplet	f-quiplet
SELECT R1.A, R1.C,	select < 2 >< 4 >	select < 1,1 >< 2,2 >	select < 1,1 >
R2.F, R2.H	< 2 >< 2 >		
FROM R1, R2		< 1,1 >< 1,1 >	< [1,0,1,0], [0,1,0,1] >
WHERE R1.B = R2.F			< 1,1 > [0,1,0,0], [0,1,0,0]

Two clustering techniques, K-centers and K-M, were used in [5] for the clustering phase. Further, two ways of performing the ID task were used for every clustering algorithm: the naïve Bayes classifier (NBC) and the outlier detection technique.

Our results are reported for each n-quiplet representation and are compared using the two clustering methods. However, the TSASM algorithm is used in this paper for clustering data. In addition, a simple classifier based on Euclidean distance 1 is used for the ID task.

The main idea of the NBC is that each instance *x* of the data is defined as the attribute value conjunction, and the target function *f*(*x*) only has values from several finite set *V*. The attributes match a set of observations and the distinct classes are the elements of *V* that are associated with these observations. A set of training examples DT can be provided for the classification problem. Moreover, a new instance with values of attributes (*a*1,…,*an*) is specified, to predict the class or target value for the new instance. The most probable class value VMAP is assigned to the new instance and given the attributes (*a*1,…,*an*).

The outlier detection technique is dependent on the median of absolute deviations (MAD) test [34]. There are *n* data points as profile quiplets that can be assumed using the MAD test. Assume that *di* denotes a data point's distance *i* from the cluster center to which is belongs. Assume *d* denotes the median value of *dis* for *i* = 1,2,…,*n*. *MAD* can then be calculated using Eq. (6):

$$MAD = median_i \left(\left| d_i - \bar{d} \right| \right) \tag{6}$$

Moreover, for every point i we compute:

$$Z_i = \frac{0.6745\ (d_i - \bar{d})}{MAD} \tag{7}$$

Checking if $|Zi| > D$, then d_i can be considered an outlier, which means that point i is an outlier. D denotes a constant that is experimentally evaluated, where its value is assumed to be 1.5.

The FN and FP rates are used for comparisons. FN represents the fraudulent transaction percentage identified as genuine and FP represents the genuine transaction percentage identified as malicious [35].

The proposed algorithm is dominated all results of quiplet representations in the FN rates as shown in Figs. 3(b), 3(d), 3(f), 3(h), 4(b), 4(d), 4(f), 4(h), 5(b), 5(d), 5(f) and 5(h). The results indicate that our proposed algorithm can result in very low FN rates.

A comparison between the proposed algorithm and c-quiplet representation is shown in Figs. 3(a) and 3(c) using the NBC via the K-M and K-centers approaches respectively. Strong performance can be achieved using the NBC classifier as s is increased for both algorithms. However, our proposed algorithm can result in low FP rates, although performance is reduced if a simple classifier is used.

A comparison between the proposed algorithm and c-quiplet representation is shown in Figs. 3(e) and 3(g) using the outlier detection methodology via the K-M and K-centers approaches respectively. The results indicate that our proposed algorithm provides better results for all s values.

A comparison between the proposed algorithm and m-quiplet representation is shown in Figs. 4(a) and 4(c) using the NBC via the K-M and K-centers approaches respectively. The results shown in these two figures indicate that the NBC can lead to smaller FP rates than our proposed algorithm. When the s value is increased, the FP rate is reduced because of simple classifier performance.

A comparison between the proposed algorithm and m-quiplet representation is shown in Figs. 4(e) and 4(g) using the outlier detection methodology via the K-M and K-centers approaches respectively. The results indicate that our proposed algorithm gives better results for all s values.

A comparison between the proposed algorithm and f-quiplet representation is shown in Figs. 5(a) and 5(c) using the NBC via the K-M and K-centers approaches respectively. The results indicate that our proposed algorithm results in smaller FN rates than does the NBC algorithm for s = [0,2]. Once the value of s is increased above 2, the f-quiplet provides more desirable FP rates. Moreover, it can lead to lower FT rates for s = [0,4], or s = [2] if applying the NBC with K-centers.

A comparison between the proposed algorithm and f-quiplet representation is shown in Figs. 5(e) and 5(g) using the outlier detection methodology via the K-M and K-centers approaches respectively. The results indicate that our proposed algorithm gives better results for all s values.

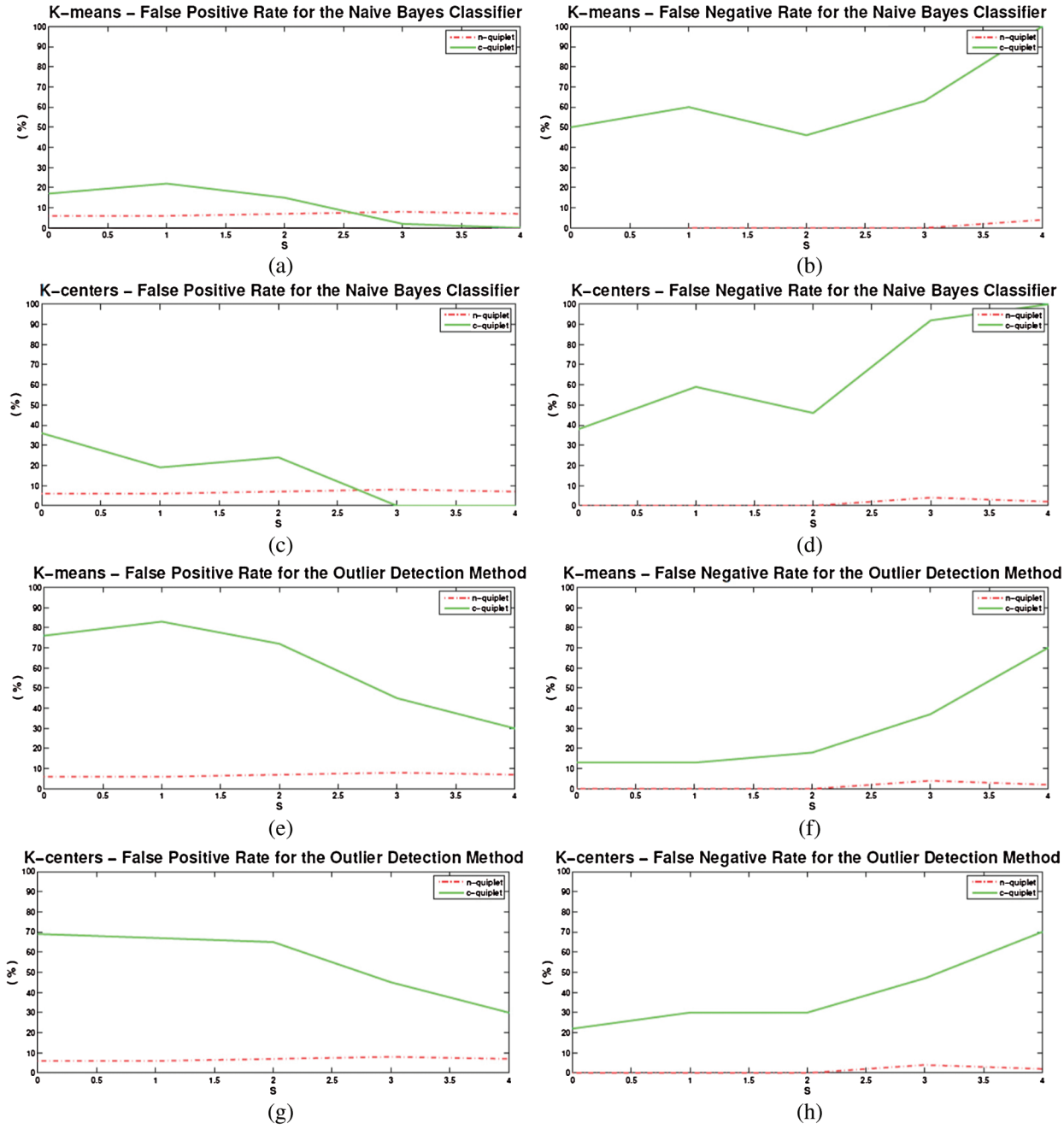

Figure 3: C-quiplet representation results

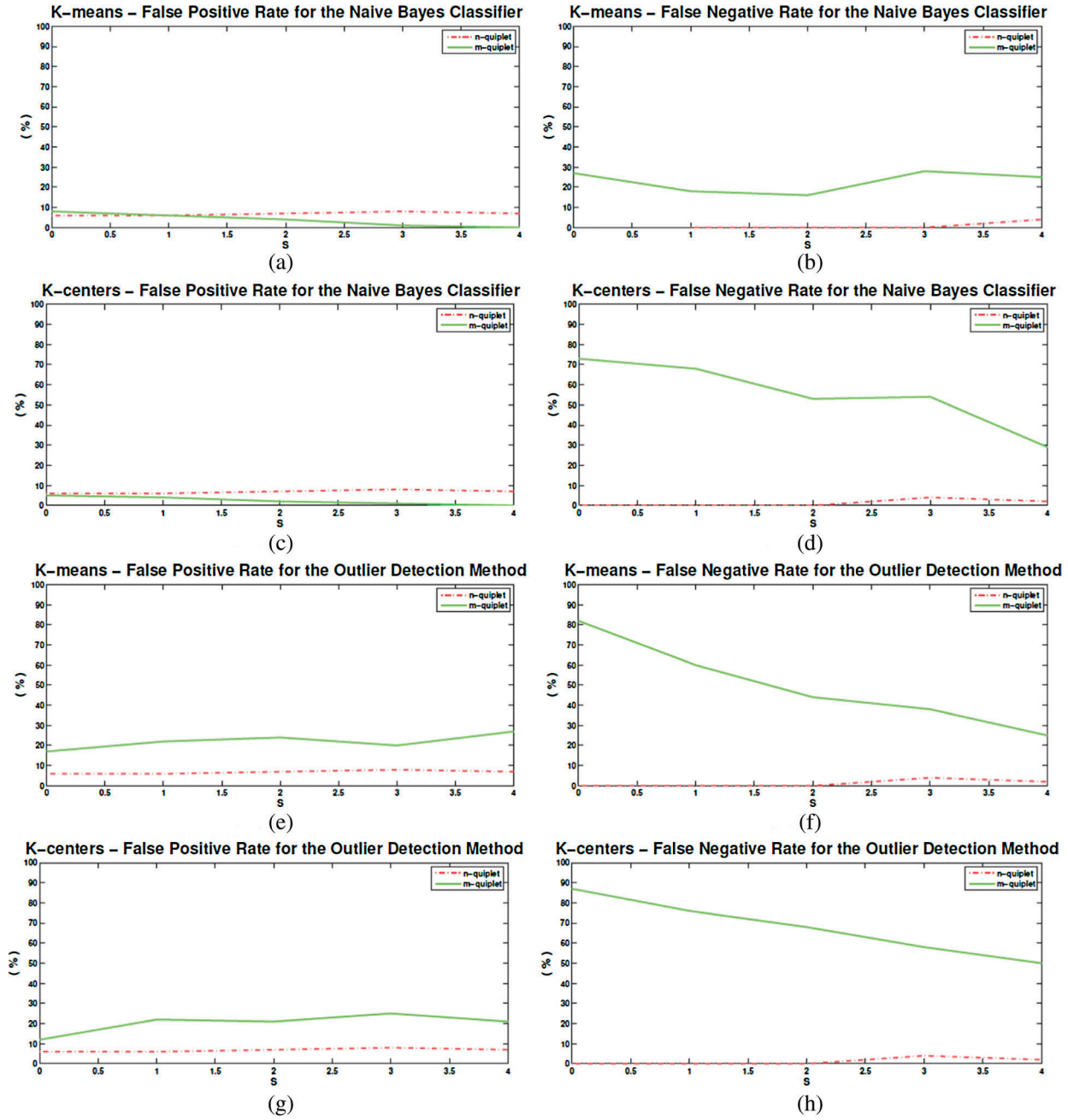

Figure 4: M-quiplet representation results

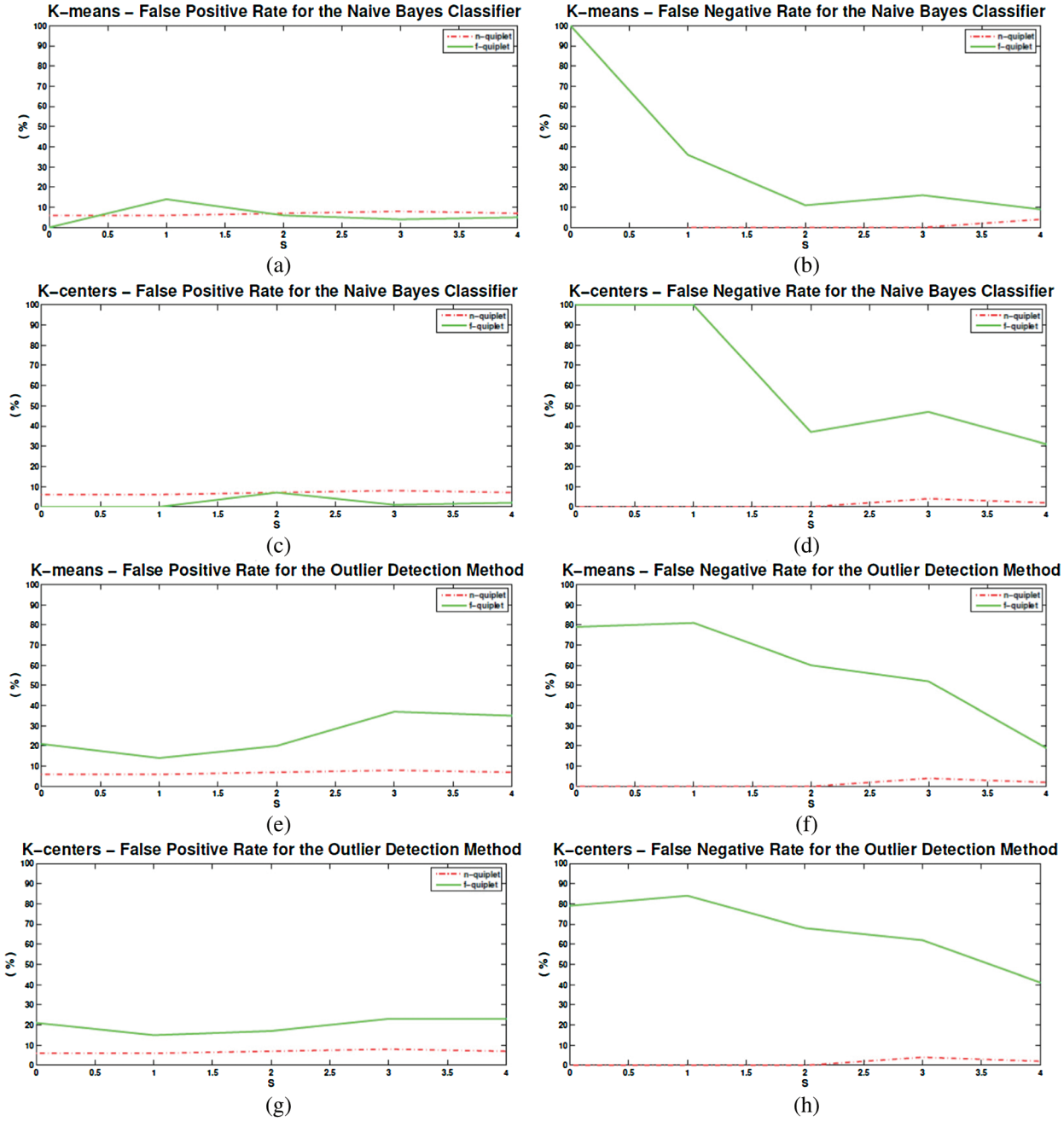

Figure 5: F-quiplet representation results

5 Conclusion

New challenges facing DBMS are presented in this paper. It explains how traditional DBMS can be effective for guaranteeing security appropriate for high-value data. A new DBMS architecture is proposed in this paper. The ID technique is used to protect the database against internal and external threats. IDS is used by organizations that do not have a clear role architecture assigned to every user. The TSASM clustering algorithm can be used to generate profiles from intrusion-free log files. A machine learning algorithm is used for the ID task for new issued queries. A synthetic database is built to represent real

data log files. The experimental results indicate that the proposed method performs effectively against malicious transactions. The proposed method can lead to extremely low FN and FP rates; thus database security can be increased. However, this paper has the limitation that we only tested our proposed algorithm on one data set. Thus, in future, our algorithm should be tested with more than one large real data set.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The author declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. Ali and S. Gressel, "Information and reformation in KM systems: big data and strategic decision-making," *Journal of Knowledge Management*, vol. 21, no. 1, pp. 71–91, 2017.

[2] K. Imdadul, Q. T. Vien, T. A. Le and G. Mapp, "A comparative experimental design and performance analysis of snort-based intrusion detection system in practical computer networks," *Computers*, vol. 6, no. 1, pp, 1–15, 2017.

[3] H. Vajiheh and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.

[4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat *et al.,* "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[5] L. Qi, W. Li, J. Wang and M. Cheng, "A SQL injection detection method based on adaptive deep forest," *IEEE Access*, vol. 7, pp. 145385–145394, 2019.

[6] A. Nicolas, P. Bonnet, L. Bouganim, B. Nguyen, P. Pucheral *et al.,* "Personal data management systems: The security and functionality standpoint," *Information Systems*, vol. 80, pp. 13–35, 2019.

[7] Y. Xiang, Z. Tian, J. Qiu, S. Su and X. Yan, "An intrusion detection algorithm based on feature graph," *Computers, Materials & Continua*, vol. 61, no. 1, pp. 259–273, 2019.

[8] L. D. Wang, "Big data in intrusion detection systems and intrusion prevention systems," *Journal of Computer Networks*, vol. 4, no. 1, pp. 48–55, 2017.

[9] C. Carmen, U. Thakore, A. Fawaz, B. Chen, W. G. Temple *et al.,* "Data-driven model-based detection of malicious insiders via physical access logs," *ACM Transactions on Modeling and Computer Simulation*, vol. 29, no. 4, pp. 1–25, 2019.

[10] G. Rajesh, S. Tanwar, S. Tyagi and N. Kumar, "Machine learning models for secure data analytics: A taxonomy and threat model," *Computer Communications*, vol. 153, pp. 406–440, 2020.

[11] C. J. Paul, Y. Kaji and N. Yanai, "RBAC-SC: role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12240–12251, 2018.

[12] R. Markus, S. Wunderlich, D. Scheuring, D. Landes and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.

[13] C. J. Francisco, D. Gil, H. Mora, B. Volckaert and A. M. Jimeno, "Scheduling framework for distributed intrusion detection systems over heterogeneous network architectures," *Journal of Network and Computer Applications*, vol. 108, pp. 76–86, 2018.

[14] A. Ahmed and I. Alsmadi, "Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach," *Journal of Network and Computer Applications*, vol. 80, pp. 152–164, 2017.

[15] V. L. J. García, A. L. S. Orozco and J. M. Vidal, "Advanced payload analyzer preprocessor," *Future Generation Computer Systems*, vol. 76, pp. 474–485, 2017.

[16] M. Ibéria, M. Beatriz, N. Neves and M. Correia, "SEPTIC: Detecting injection attacks and vulnerabilities inside the DBMS," *IEEE Transactions on Reliability*, vol. 68, no. 3, pp. 1168–1188, 2019.

[17] B. Muhammet and R. Das, "A novel honeypot based security approach for real-time intrusion detection and prevention systems," *Journal of Information Security and Applications*, vol. 41, pp. 103–116, 2018.

[18] S. Mina, M. M. Javidi and S. Hashemi, "Detecting intrusion transactions in database systems: A novel approach," *Journal of Intelligent Information Systems*, vol. 42, no. 3, pp. 619–644, 2014.

[19] B. Luca, M. Cello, M. Marchese, E. Mariconti, T. Naqash *et al.,* "Statistical fingerprint-based intrusion detection system (SF-IDS)," *International Journal of Communication Systems*, vol. 30, no. 10, e3225, 2017.

[20] L. Wenjuan, W. Meng and H. H. Ip, "Developing advanced fingerprint attacks on challenge-based collaborative intrusion detection networks," *Cluster Computing*, vol. 21, no. 1, pp. 299–310, 2018.

[21] K. G. Rajesh, N. Mangathayaru and G. Narsimha, "An approach for intrusion detection using novel gaussian based kernel function," *Journal of Universal Computer Science*, vol. 22, no. 4, pp. 589–604, 2016.

[22] M. Islabudeen and M. K. Devi, "A smart approach for intrusion detection and prevention system in mobile *ad hoc* networks against security attacks," *Wireless Personal Communications*, vol. 112, no. 1, pp. 1–32, 2020.

[23] B. Andrea, A. Bartolini, M. Lombardi, M. Milano and L. Benini, "A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 634–644, 2019.

[24] R. C. Ann and S. B. Cho, "Anomalous query access detection in RBAC-administered databases with random forest and PCA," *Information Sciences*, vol. 369, pp. 238–250, 2016.

[25] R. U. Pratap and N. K. Singh, "Weighted role based data dependency approach for intrusion detection in database," *IJ Network Security*, vol. 19, no. 3, pp. 358–370, 2017.

[26] K. Ashish and E. Bertino, "Design and implementation of an intrusion response system for relational databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 875–888, 2010.

[27] S. Asmaa, E. Bertino, S. R. Hussain, D. Landers, R. M. Lefler *et al.,* "DBSAFE—An anomaly detection system to protect databases from exfiltration attempts," *IEEE Systems Journal*, vol. 11, no. 2, pp. 483–493, 2015.

[28] B. Seok-Jun and S. B. Cho, "A convolutional neural-based learning classifier system for detecting database intrusion via insider attack," *Information Sciences*, vol. 512, pp. 123–136, 2020.

[29] R. A. Alsowail and T. Al-Shehari, "Empirical detection techniques of insider threat incidents," *IEEE Access*, vol. 8, pp. 78385–78402, 2020.

[30] Y. Yuqing, D. Zhou and X. Yang, "A multi-feature weighting based K-means algorithm for MOOC learner classification," *Computers, Materials & Continua*, vol. 59, no. 2, pp. 625–633, 2019.

[31] T. Ling, C. Li, J. Xia and J. Cao, "Application of self-organizing feature map neural network based on K-means clustering in network intrusion detection," *Computers, Materials & Continua*, vol. 61, no. 1, pp. 275–288, 2019.

[32] L. R. Costa, D. Aloise and N. Mladenović, "Less is more: basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering," *Information Sciences*, vol. 415, pp. 247–253, 2017.

[33] G. Daniel and T. Vidal, "HG-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering," *Pattern Recognition*, vol. 88, pp. 569–583, 2019.

[34] R. I. Adeyanju, M. H. Lee, M. Riaz, M. R. Abujiya and N. Abbas, "Outliers detection models in shewhart control charts; an application in photolithography: A semiconductor manufacturing industry," *Mathematics*, vol. 8, no. 5, pp. 1–17, 2020.

[35] K. M. Reza, M. B. Shirzad and S. Mehmandoost, "CID: A novel clustering-based database intrusion detection algorithm," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2020.