Journal of King Saud University – Computer and Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Application of OPTICS and ensemble learning for Database Intrusion Detection

Sharmila Subudhi [a], Suvasini Panigrahi [b,*]

[a] Department of CS&IT, Institute of Technical Education and Research, S'O'A Deemed to be University, Odisha 751030, India
[b] Department of CSE, Veer Surendra Sai University of Technology, Odisha 768018, India

ARTICLE INFO

Article history:
Received 19 October 2018
Revised 8 April 2019
Accepted 1 May 2019
Available online xxxx

Keywords:
Intrusion detection
Database
OPTICS
Outlier factor
Ensemble classifier

ABSTRACT

In this paper, we have proposed a novel approach for detecting intrusive activities in databases by the use of clustering and information fusion through ensemble learning. We have applied OPTICS clustering on the transaction attributes for building user behavioral profiles. A transaction is initially passed through the clustering module for computing its cluster belongingness and an outlier factor that signifies its degree of outlierness. Depending on the outlier factor value, the transaction is classified as genuine or an outlier. Each outlier transaction is further analyzed by passing it onto an Ensemble Learner that applies three different aggregation methods, bagging, boosting and stacking. We have conducted experiments using stochastic models to demonstrate the effectiveness of the proposed system. The performance of the three different ensembles are evaluated and compared based on various metrics. Moreover, our system is found to exhibit better performance as compared to other approaches taken from the literature.

1. Introduction

Database security is a major concern among all, even after the introduction of security measures in several information related fields. According to a study conducted by the Association of Certified Fraud Examiners (*ACFE*) in 2016, a typical organization generally loses approximately 5% of annual revenue in a given year due to fraudulent activities (Global fraud survey, 2017). The statistics show the severity of attacks in information systems, depicting the heavy financial losses faced by the organizations worldwide. Moreover, the intrusive activities can incur a loss of customer's trust on organizations, which may eventually lead to lawsuits.

Traditional database security services provide security features to protect the information systems from intrusive attacks. Despite of these security measures, authorized insiders within the organization can compromise data by misusing their authorization or by the outsiders who can gain access by exploiting authorized identity. An intrusive attack in the database can thus be classified as outside attack and inside attack. An outside attack can be referred as the malicious transactions performed by unauthorized users from outside the organization, who may gain access to the database by misusing system loopholes (Panigrahi et al., 2013). However, an insider, who is fully aware of the security setup of the organization and may have some access permissions to data and system resources (Furnell, 2004), can carry out unauthorized database transactions, which can remain undetected for a long time, thereby causing severe damage to the database system. Thus, the insider threat problem has gained much importance amongst the security research community. Moreover, it is found that the insiders cause the primary security threats to database systems (Murray, 2005). As the existing security measures are not sufficient in preventing such novel attacks, it becomes necessary to develop an efficient Database Intrusion Detection System (DIDS) for protecting information against malicious attacks.

In this paper, a density-based clustering method, namely, OPTICS (Ordering Points To Identify the Clustering Structure) (Ankerst et al., 1999) is applied to the transactional attributes to

* Corresponding author.
E-mail address: spanigrahi_cse@vssut.ac.in (S. Panigrahi).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2019.05.001
1319-1578/

model a system that build profiles for representing normal user behavior accurately. However, the shift in job function of a database user can lead to deviation in database activity that appear as outliers but are not necessarily malicious. This is the main reason for which we have employed an ensemble classifier for further strengthening our initial findings from the clustering module and learning the behavioral changes, thus, minimizing the losses suffered by database owners. In the current work, we have applied the bagging, boosting and stacking ensemble learning methods. To the best of our knowledge, this is the first ever attempt to apply an ensemble of classifiers and comparative analysis of these classifiers for database intrusion detection.

The organization of this paper proceeds as follows. Section 2 describes related work on database intrusion detection. The background of various techniques implemented in our current work are presented in Section 3.We describe the methodology of our intrusion detection system along with the algorithm in Section 4. Section 5 deals with the results obtained from experimental analysis. Finally, Section 6 concludes the paper summarizing the contributions and research outcomes.

2. Related Work

A DIDS cited in Brahma and Panigrahi (2015) employs the Artificial Neuro Fuzzy Inference System (ANFIS) for capturing the user behavioral profiles. The authors have used Sugeno-fuzzy inference method and Artificial Neural Network (ANN) to generate some if-then rules. The incoming transactions that do not comply with these fuzzy rules are marked as malicious. The paper (Singh et al., 2016) describes the importance of association rules and cluster analysis in finding the illegitimate activities from the database usage patterns. Initially, the normal profiles are generated according to the roles of the users by deploying a clustering algorithm. Upon the non-compliance of a new transaction with the existing rules, it is then marked as fraudulent. Ronao and Cho (Ronao and Cho, 2016) have developed an anomaly-based DIDS to detect malicious SQL queries submitted to a Role-based relational database. At first, the Principal Component Analysis (PCA) method is used on the transactions performed by the users to construct the behavioral profiles. Later, the Random Forest with Weighted voting (WRF) classifier is applied on these profiles for detecting intrusive activities.

Another hybrid DIDS has been suggested in Bu and Cho (2017) that employs the Convolutional Neural Network (CNN) and Learning Classifier System (LCS) in tandem. The LCS devise new transactional rules to identify any anomalous events from the database audit logs, while the CNN is used for classification purpose. A database security model suggested in Wei et al. (2018) has deployed the Quantum Oblivious Key Transfer based Private Query (QOKT-PQ) protocol along with the Low Shift and Addition (LSA) technique for examining transactional queries from large sized database systems. Another database security system has been developed by Yesin et al. (2018) that ensures security by establishing universal basis of relations with the database schema. They have considered the following schema objects like, procedures, triggers, tables and packages to construct various rules regarding the access control to the objects and other aspects of database such as, data integrity, abstraction and recoverability. Jayaprakash and Kandasamy (2018) have proposed an anomaly based DIDS that employs the Naïve Bayes supervised classifier for finding out the intrusive SQL queries submitted to the database.

From the above mentioned studies, it is found that all the existing systems aim to detect the intrusive activities in databases accurately. However, a major issue still remains that, while protecting the databases from maligned transactions, the false alarm rate also needs to be attenuated. This fact has been identified in Axelsson (2000) that due to the base-rate fallacy effect, the performance of an IDS is more influenced by its proficiency in reducing the generation of false alarms rather than enhancing the detection rate. Addressing this challenge is one of the main motivating factors while conducting our current investigation.

3. Background study

In order to understand the training and intrusion detection process of our proposed DIDS, we present a brief summary regarding the basic concepts of the techniques used, namely, OPTICS and Ensemble learners.

3.1. Clustering with OPTICS

In this work, we have applied OPTICS (Ordering Points To Identify the Clustering Structure), a density-based approach, which can identify noise (outliers) by segregating the high dense regions from the sparse ones (Ankerst et al., 1999). This method requires two parameters as input – ε which represents the radius and *MinPts* that signifies the number of points essential to build a cluster (Breunig et al., 1999) and produces an orderly sequence of points depicting the clustering structure as output.

Suppose, C= {c_1, c_2,$\cdots$, c_n} be the clusters for representing n behavioral patterns corresponding to n users and A = {a_1, a_2,$\cdots$, a_j} be the set of j attributes used for defining the clusters. The fundamental idea of the OPTICS algorithm is that for each instance k in a cluster C_i, there exist at least a minimum number of instances *MinPts* in N_ε(k). For a constant value of *MinPts*, the clusters based on less value of ε exhibit the properties of a completely density-connected set than the clusters based on higher ε value. An instance k is said to be a core point if at least *MinPts* number of instances are found within its ε-neighborhood denoted by N_ε(k). For each instance, OPTICS compute only two values, namely, core-distance (cd) and reachability-distance (rd).

The core-distance (cd) of an object k ($cd_{\varepsilon,Minpts}(k)$) is the smallest distance between instance k and an object in its ε-neighborhood N_ε(k), which is expressed as:

$$cd_{\varepsilon,Minpts}(k) = \begin{cases} undefined & \text{if } |N_\varepsilon(k)| < MinPts \\ Minpts\_distance(k) & \text{otherwise} \end{cases} \quad (1)$$

The reachability-distance (rd) of k with respect to another core object q ($rd_{\varepsilon,Minpts}(k,q)$) is defined as the smallest distance such that k is directly density-reachable from q.

$$rd_{\varepsilon,Minpts}(k,q) = \begin{cases} undefined & \text{if } |N_\varepsilon(k)| < MinPts \\ max(cd(q), dist(q,k)) & \text{otherwise} \end{cases} \quad (2)$$

The local reachability distance ($lrd_{MinPts}(k)$) of an object k is the inverse of the average reachability-distance from *MinPts*-nearest neighbors of k.

$$lrd_{MinPts}(k) = 1 \Big/ \frac{\sum_{o \in N_{MinPts}(k)} rd_{\infty,MinPts}(k,o)}{|N_{MinPts}(k)|} \quad (3)$$

where, o is the neighbor of k. An object k is detected as an outlier by an outlier factor ($OF_{MinPts}(k)$), which is defined as the average of the ratios of lrds of the *MinPts*-nearest neighbors and k. It is observed

Please cite this article as: S. Subudhi and S. Panigrahi, Application of OPTICS and ensemble learning for Database Intrusion Detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.05.001

that the *OF* value of the instances that lie within a cluster are close to one (Kim et al., 2013).

$$OF_{MinPts}(k) = \frac{\sum_{o \in N_{MinPts}(k)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(k)}}{|N_{MinPts}(k)|} \quad (4)$$

3.2. Ensemble Learning

An ensemble represents combination or aggregation of multiple classifiers to produce better and more robust performance than single learners (Oza and Tumer, 2008). The ensemble classifies the data points and predicts their labels either by voting or averaging of results of each single classifier. It reduces the risk of increase in false positive results by choosing a good combination of different classifiers. There exist various types of ensemble techniques (Oza and Tumer, 2008), but the most popularly employed methods include – Bagging (Breiman, 1996), Boosting (Schapire et al., 2003) and Stacking (Wolpert, 1992).

4. Proposed approach

The proposed DIDS initially applies data preprocessing on the raw dataset, in which all the transactional attributes are mapped into numerals as the computation required to detect intrusive patterns are based on integer values. Secondly, the normalization procedure is applied on all the numeric attribute values for converting them within the range of [0,1]. This is essential for reducing the bias effect of large value data fields. The OPTICS clustering is then used for building normal profiles of users from the historical records. A transaction is considered to be genuine if it adheres to any of the established profiles. On the other hand, transactions that do not comply are passed to an ensemble learner for detecting the possibility of intrusions by combining the information from the current transaction as well as past behavior of the corresponding database user. In order to meet the above said functionalities, the proposed system has been designed with the following three components:

1. Activity Builder (AB)
2. User Activity Database (UAD)
3. Ensemble Learner (EL)

4.1. Activity Builder (AB)

The activity builder deals with the profile building of the database users by considering their past transactional data. We represent each transaction comprising of the following 8 tuples:

$\langle u\_id, transaction\_id, query\_type, table\_list, att\_list, time\_slot, loc, time\_gap \rangle$

where,

- *u_id*: unique identification number to identify each user
- *transaction_id*: unique identification number to identify each database transaction
- *query_type*: denotes the type of queries invoked by the user for performing a transaction.
- *table_list*: Each table in the database schema is assigned a unique ID. This attribute signifies the list of table IDs that are accessed during a transaction.
- *att_list*: specifies the list of attribute IDs that are accessed in a particular transaction.
- *time_slot*: the time slot in which a transaction was carried out during a day. We have sectioned 24 h of a day into 48 slots, each of 30 min duration starting from 00.00 am.
- *loc*: location where the transaction was carried out. We have considered 1 for the transaction carried out from an office location, 2 for home and 3 for other locations.
- *time_gap*: time gap between successive accesses to the database by the same user measured in minutes.

Let us consider an example of a transaction (Tr1), consisting of two queries *Q1* followed by *Q2*, has been submitted to the database by a user having *u_id* = 10 for performing certain task.

Q1: SELECT x, y from table T1 where z = 1

*Q2: DELETE from T2 where w = 1*where, *x*, *y* and *z* are the attributes of table *T1* and *w* is the attribute of table *T2*. The attributes accessed for query *Q1* is <*z*, *x*, *y*> while for query *Q2*, it is <*w*>. In the transaction, the *query_type* is <*SELECT, DELETE*>, *att_list* is <*z*, *x*, *y*, *w*> and the *table_list* is <*T1*, *T2*>. Let, the *transaction_id* for Tr1 is 1. After mapping of the categorical values to integers, suppose the *query_type* <*SELECT, DELETE*> = <1, 4>, *att_list* <*z*, *x*, *y*, *w*> = <40, 23, 12, 6 > and the *table_list* <*T1*, *T2*> = <3, 6 > respectively. Assuming that the user has carried out the transaction from office (*loc* = 1) in between 6.00 pm and 6.30 pm (*time_slot* = 37) with a *time_gap* of 21 min from his/her last transaction. So, the profile of user 10 can be depicted as follows: <10, 1, {1, 4}, {3, 6}, {40, 23, 12, 6}, 37, 1, 21>. A sample database record of two users is given in Fig. 1 for showing how the attributes are taken for the user profiling.

Before the initiation of training and intrusion detection process, we perform data preprocessing of all the features associated with the transactional raw data. After the data preprocessing is over, the activity builder constructs the user profiles by applying the OPTICS algorithm, which takes the processed attributes along with ε and *Minpts* values as input. For each instance present in the dataset, the core-distance (*cd*) and reachability-distance (*rd*) values are computed according to Eq. (1) and Eq. (2) respectively. It is observed that the points having close *rd* values form a group (Ankerst et al., 1999).

4.2. User Activity Database (UAD)

The UAD deals with the transaction storage warehouse in our proposed DIDS that provides information regarding the genuine

u_id	transaction_id	query_type	table_list	att_list	time_slot	loc	time_gap
1	10	select	3, 1, 14	1, 6, 2, 3	9	1	4
	11	alter	17, 33	29, 30, 31, 32, 25	12	2	1
2	78	update	10, 4	4, 6, 50	33	1	5

Fig. 1. Sample Database Records.

Please cite this article as: S. Subudhi and S. Panigrahi, Application of OPTICS and ensemble learning for Database Intrusion Detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.05.001

and intrusive patterns of the database users. A large sized historical records of both normal and intrusive transactions are required for training and building the models. This is done to avoid inconvenience to the legitimate users, those who occasionally deviate from their normal activity. Hence, we have maintained two tables, namely, Genuine History Table (GHT) for storing the genuine transactional activities of each user and a generic Malicious History Table (MHT) comprising of various types of past detected intrusive database transactions.

Whenever a transaction is found to be genuine after passing through the clustering module, we update the GHT in the activity database. On the other hand, when a transaction is identified as genuine/intrusive after the decision making of the Ensemble Learner, we update the GHT/MHT accordingly. This updating procedure of the UAD is done for reducing the misclassification rate. We have presented a few sample records for GHT and MHT in Fig. 2 for better understanding. From the figure, it is quite clear that the GHT stores only the genuine data of each user by considering *transaction_id* as primary key and MHT stores a generic malicious data irrespective of any user.

4.3. Ensemble Learner (EL)

In the proposed work, the ensemble learner is built upon the bagging, boosting and stacking ensemble methods. All these ensembles take the outlier transactions coming from the activity builder and the respective user's profile as input for making the final decision.

Although several machine learning classifiers are available, in this work, we have used the following five machine learning algorithms – Naïve Bayes, Decision Tree (DT), Rule Induction (RI), *k*-Nearest Neighbor (*k*-NN) and Radial Basis Function Network (RBFN) for building the ensembles. The Naïve Bayes classifier is quite simple, which holds the assumption regarding the independence of features present in the dataset and can perform classification on less training data (Rish, 2001). The DT algorithm is simple and easy to interpret and can handle the skewed and non-linear data (Quinlan, 1986). The RI method is used (Cohen, 1995) in order to mine the regular patterns present in the dataset efficiently by utilizing some rules. These inferred rules (rule set) can be presented using IF-THEN clause for easy understanding and are used for classification of new instances. The *k*-NNs are robust to noisy training data and are intuitive in nature as it presumably assigns the new data points to the same class as the training class which are closest to them in the feature mapping space (Cover and Hart, 1967). The RBFNs are tolerant to outliers and can classify the unseen samples effectively with the help of three layered neural network architecture (Schwenker et al., 2001).

The bagging and boosting ensembles construct their respective learning models, while considering the above said classifiers individually. Whenever an inconsistent transaction, coming from the activity builder, is fed to the bagging ensemble, it distinguishes the corresponding transaction as genuine or intrusive by voting out the results of each base classifier. In the boosting ensemble, only the anomalous transactions marked by the previous base classifier are being considered for final decision making. Likewise, the stacking ensemble uses different combinations of classifiers in two layers, namely, layer 0 (base learner) and layer 1 (meta learner). The outcomes of base learner are given to meta learner for achieving improved accuracy and lower false alarms. Once a transaction is marked as genuine or intrusive, the GHT or MHT is updated respectively.

The components of the proposed system along with the flow of events are presented in Fig. 3, while the algorithmic steps of the proposed DIDS has been presented in Algorithm 1.

u_id	transaction_id	query_type	table_list	att_list	time_slot	loc	time_gap	label
1	10	select	3, 1, 14	1, 6, 2, 3	9	1	4	genuine
	11	alter	17, 33	29, 30, 31, 32, 25	12	2	1	genuine
2	78	update	10, 4	4, 6, 50	33	1	5	genuine

(a) Genuine History Table (GHT)

query_type	table_list	att_list	time_slot	loc	time_gap	label
delete	8, 11	16, 24, 5, 16	11	0	2	malicious
alter	55	55, 57, 75, 67	9	2	4	malicious
insert	66	1, 82, 40	27	1	1	malicious

(b) Malicious History Table (MHT)

Fig. 2. Sample Records Maintained in UAD.

Please cite this article as: S. Subudhi and S. Panigrahi, Application of OPTICS and ensemble learning for Database Intrusion Detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.05.001

Algorithm 1 Proposed Ensemble based Database Intrusion Detection Algorithm

Input: $D_u[m][n]$, ε, $MinPts$
Output: Genuine or Fraud

$Norm\_D_u[m][n] = normalize\ (D_u[m][n])$ ▷ Data Normalization
$[Train[m_1][n], Test\ [m_2][n]] = divide\ (Norm\_D_u[m][n])$
▷ Train and Test sets are extracted
▷ $m = m_1 + m_2$
$Prof_u = OPTICS\_cluster\ (Train[m_1][n_1], \varepsilon, MinPts)$ ▷ Using Eq. (1)- Eq. (2)
▷ $Prof_u$= User Profile of User u
$rd[m_2] = OPTICS\_cluster(Test[m_2][n], \varepsilon, MinPts)$
▷ Generate rd values for Incoming transaction
$OF[m_2] = OPTICS\_cluster\ (rd[m_2])$ ▷ Using Eq.4
if $OF[m_2] < 1$ **then**
    Output ("Database Access is allowed")
    Insert_GHT $(Test_x)$ ▷ Insert $Test_x$ in GHT
    ▷ $Test_x \in Test[m_2][n],\ \ x = 1, \cdots, m_2$
else
    goto EL ▷ EL: Ensemble Learner
end if
Collect History Information D(U_x) about user U_x from User Activity Database
EL:
$Bag = Bagging(Test_x, Prof_U)$ ▷ Bag: Bagging model
$Boost = Boosting(Test_x, Prof_U)$ ▷ $Boost$: Boosting model
$Stack = Stacking(Test_x, Prof_U)$ ▷ $Stack$: Stacking model
if $(Test_x \in Bag) \| (Test_x \in Boost) \| (Test_x \in Stack)$ **then**
    Insert_GHT $(Test_x)$ ▷ Insert $Test_x$ in GHT
    Output ("Genuine")
else
    Insert_MHT $(Test_x)$ ▷ Insert $Test_x$ in MHT
    Output ("Malicious")
end if

5. Simulation and Results

The experiments were carried out on a 2.40 GHz i5-4210U CPU system and the proposed DIDS has been implemented in MATLAB 14a.

5.1. Data generation

For showing the efficacy of our proposed model, we have tested it on a large scale synthetic dataset comprising of 41,390 transactional records, due to the unavailability of real life data or any benchmark dataset in this field. Besides, the accessibility of any real dataset is not revealed even after a detailed literature survey of the existing database intrusion detection systems.

In this work, we have used the simulator proposed in (Panigrahi et al., 2013) for generating synthetic transactions that represent the behavior of genuine users as well as intruders. They have developed a transaction simulator in MS-SQLServer 2000 that follows the database schema of standard transactional web bench-

Please cite this article as: S. Subudhi and S. Panigrahi, Application of OPTICS and ensemble learning for Database Intrusion Detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.05.001

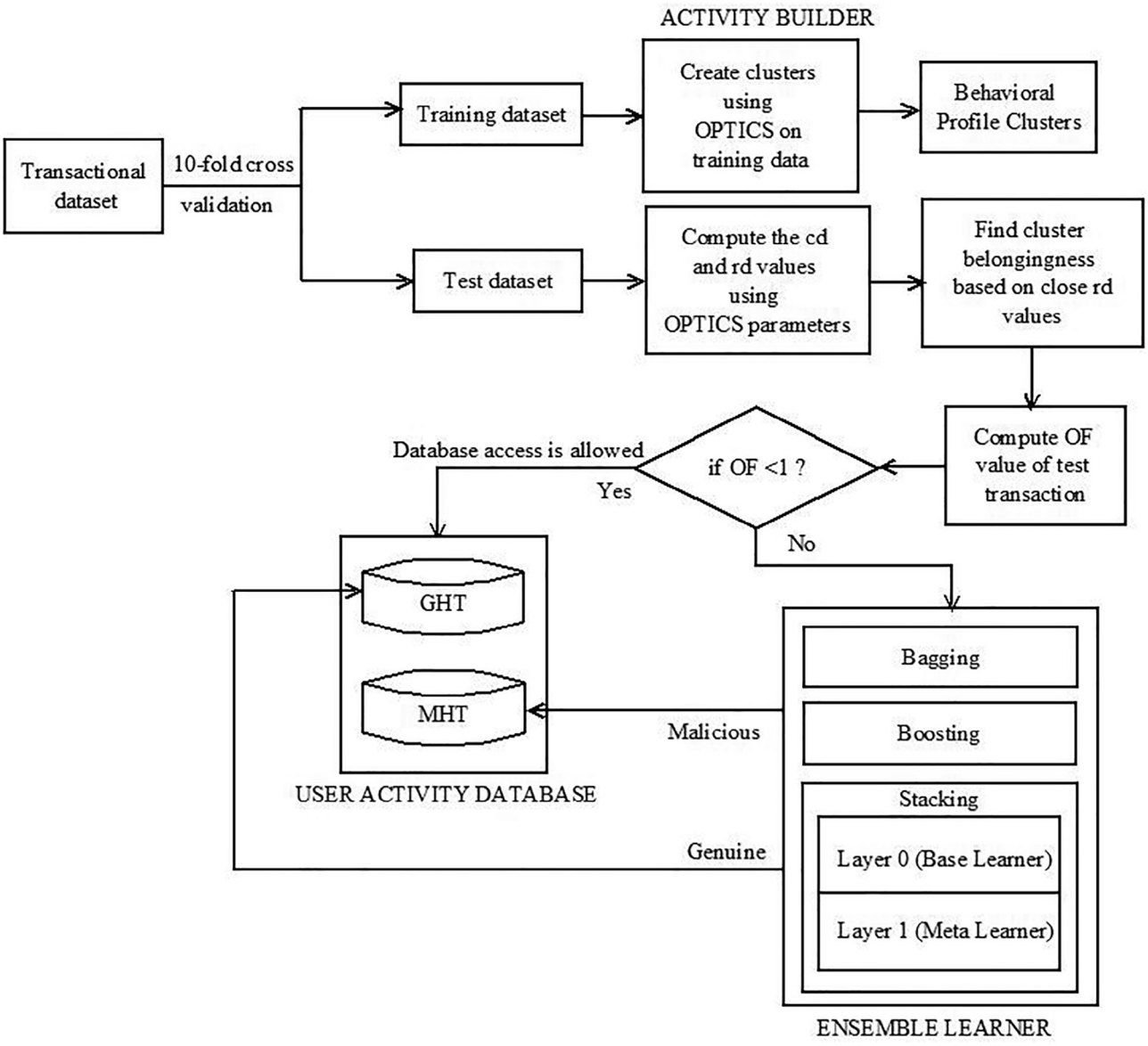

Fig. 3. Flow of Events in the Proposed Database Intrusion Detection System.

mark TPC-W (Transaction Processing Council-Web Commerce). The transaction generation is controlled in the simulator at the granularity level of the table attributes (*att_list*, *table_list*) as well as transactional attributes (*query_type*, *time_slot*, *time_gap*, *loc*). The simulator consists of two different modules, namely, Genuine Transaction Generation Module (GTGM) and Malicious Transaction Generation Module (MTGM) to generate genuine and malicious transactions and another third module called as Markov Modulated Poisson Process Module (MMPPM) to define and control the arrival rates of the transaction requests from the genuine users and intruders. The MMPPM is a dual layered Markov Modulated Poisson Process (MMPP) which consists of a genuine state (S_G) and a malicious state (S_M) with their respective arrival rates λ_{S_G} and λ_{S_M}. The MMPPM is also responsible for the mixture of genuine and malicious transactions by these two states S_G and S_M along with their corresponding transition probabilities $b_{S_GS_M}$ and $b_{S_MS_G}$, while $b_{S_GS_M}$ represents the probability value of transitioning from state S_G to S_M and $b_{S_MS_G}$ denotes the likelihood value of the transition from state S_M to S_G.

Moreover, the generation of genuine transactions done by GTGM in the simulator (Panigrahi et al., 2013) has been regulated by five finite Markov chains – Genuine Select Markov Chain (GSMC), Genuine Insert Markov Chain (GIMC), Genuine Delete Markov Chain (GDMC), Genuine Update Markov Chain (GUMC) and Genuine Transaction Markov Chain (GTMC). The GSMC, GIMC, GDMC and GUMC generate the queries according to the basic four SQL commands – select, insert, delete and update respectively. The number of states required in the Markov chains GSMC, GIMC, GDMC and GUMC are defined by the number of their respective query types, i.e. if the numbers of select queries are *R*, then the number of states in GSMC will also be *R*. The GTMC produces different types of transactions by accumulating multiple queries from GSMC, GIMC, GUMC and GDMC. The states of GTMC are four as we have considered only four different query types.

Similarly, the generation of malicious transactions done by MTGM in the simulator (Panigrahi et al., 2013) has also been controlled by five finite Markov chains – Malicious Select Markov Chain (MSMC), Malicious Insert Markov Chain (MIMC), Malicious

Please cite this article as: S. Subudhi and S. Panigrahi, Application of OPTICS and ensemble learning for Database Intrusion Detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.05.001

Delete Markov Chain (MDMC), Malicious Update Markov Chain (MUMC) and Malicious Transaction Markov Chain (MTMC). All the functionalities of MTGM are similar to that of GTGM. Besides, different Gaussian distribution functions with user-defined mean (μ) and standard deviation (σ) have been used for generating various transactional attributes for simulating different categories of genuine users as well as intruders. The mean and standard deviation of the Gaussian processes are varied during the transaction generation for representing different categories of genuine and malicious users. During the experimentation, we have set the transition probabilities $b_{S_GS_M} = 0$ and $b_{S_MS_G} = 0$ to constrict the generation of genuine and malicious transactions within the states S_G and S_M of the MMPPM respectively. In order to measure the efficiency of the proposed model, eight variants of the simulator are designed by modifying certain parameters, such as $\lambda_{S_G}, \lambda_{S_M}, b_{S_GS_M}, b_{S_MS_G}, \mu$ and σ. Eight different simulation settings (S1 to S8) are presented in Table 1 for showing the variations in the simulation parameters.

5.2. Performance analysis

To analyze the performance of the proposed system, the following performance metrics (Powers, 2011) are used: *Accuracy*, *Precision*, *F1_Score*, True Positive Rate (*TPR*) and False Positive Rate (*FPR*). *Accuracy* is defined as the percentage of correctly classified transactions. *Precision* can be described as the percentage of positive detection that are correct. *TPR* denotes the ratio of true positive samples that are correctly identified by the classifier. *FPR* measures the proportion of falsely rejected genuine samples. *F1_Score* is defined as the harmonic mean of *Precision* and *TPR*.

The testing of the proposed DIDS is initiated with the 10-fold cross validation (Refaeilzadeh et al., 2009) technique for segregating the dataset into the train and test sets while evaluating the performance of our system. The effectiveness of the clustering module of our DIDS is dependent on the two parameters ε and *MinPts*. Thus, to obtain an optimal choice of the parameter values, we have experimented with various combinations of *Minpts* and ε as shown in Table 2. It is clearly visible from the table that the *Accuracy*, *TPR*, *F1_Score* and *FPR* values increases with increase in *Minpts*, while the *Precision* value decreases. It is observed that at the parameter values *Minpts* = 10 and $\varepsilon = 0$, the OPTICS algorithm produces the highest *Precision* = 69.90% and lowest *FPR* = 34.73%. The optimal parameter value (*Minpts* = 10, $\varepsilon = 0$) has been presented in bold for better visualization.

Table 3 represents the comparison of results obtained from ensemble classifiers using bagging, boosting and each individual single classifier over different performance metrics. From the table,

Table 1
Simulator Settings for data generation.

Simulator Settings	λ_{S_G}	λ_{S_M}	$b_{S_GS_M}$	$b_{S_MS_G}$	μ	σ
S1	1	4	5	1	0.50	0.50
S2	1	4	4	1	0.15	0.50
S3	1	4	4	2	0.15	0.70
S4	1	4	3	2	0.10	0.80
S5	1	2	3	3	0.10	0.80
S6	3	1	2	4	0.10	0.90
S7	4	1	1	5	0.05	0.96
S8	8	1	1	5	0.05	0.99

Table 2
Performance of OPTICS with Different Parameter Values.

Performance Metrics (in %)	Parameter Values					
	Minpts = 10 $\varepsilon = 0$	Minpts = 10 $\varepsilon = 5$	Minpts = 50 $\varepsilon = 0$	Minpts = 50 $\varepsilon = 5$	Minpts = 100 $\varepsilon = 0$	Minpts = 100 $\varepsilon = 5$
Accuracy	**56.34**	56.24	58.37	58.27	62.58	62.48
Precision	**69.90**	69.80	68.40	68.30	66.43	66.33
F1_Score	**50.73**	50.60	53.63	53.53	62.30	62.20
TPR	**58.79**	58.69	60.12	60.02	64.30	64.20
FPR	**34.73**	35.09	34.94	34.84	37.11	38.46

Table 3
Comparative Performance of Single Classifier, Bagging and Boosting in Proposed System.

Performance Metrics (in %)	Naïve Bayes			*k*-Nearest Neighbor			Rule Induction			Decision Tree			RBFN		
	Single	Bagging	Boosting	Single	Bagging	Boosting	Single	Bagging	Boosting	Single	Bagging	Boosting	Single	Bagging	Boosting
Accuracy	91.30	91.30	91.55	91.53	**91.97**	91.55	91.54	91.55	91.57	91.53	91.54	91.61	91.55	91.56	91.55
Precision	93.15	93.84	93.85	93.93	94.39	**93.97**	93.76	93.83	93.80	93.77	93.81	93.85	93.79	93.83	93.83
F1_Score	89.95	89.93	90.20	90.22	**90.68**	90.58	90.20	90.21	90.23	90.17	90.19	90.27	90.19	90.21	90.20
TPR	86.34	86.96	86.84	86.76	87.25	**87.47**	86.88	86.89	86.90	86.84	86.84	86.96	86.84	86.85	86.87
FPR	5.22	4.64	4.81	4.63	**4.23**	4.77	4.69	4.61	4.74	4.72	4.61	4.65	4.89	4.62	4.63

Please cite this article as: S. Subudhi and S. Panigrahi, Application of OPTICS and ensemble learning for Database Intrusion Detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.05.001

it is quite clear that the ensemble using *k*-NN yields promising result in bagging as well as boosting by detecting the intruders with higher accuracy and minimum false alarms as compared to individual classifiers and other ensembles as well.

In Table 4, we show the performance of the stacked ensemble classifier comprising of the following five classifiers – Naïve Bayes, *k*-NN, DT, RI and RBFN. The stacking ensemble is experimented by using different combinations of classifiers in base learner and meta learner for achieving maximum classification accuracy along with minimized misclassification rate. It is evident from Table that by using RBFN as a meta classifier and others as base classifier, the proposed model gives the best performance over all metrics – *Accuracy* = 92.17%, *Precision* = 95.27%, *F1_Score* = 90.86% and *FPR* = 3.5%. Therefore, we have chosen this combination of classifiers (visualized in bold) for stacked ensemble. After analyzing the results presented in Table 3 and Table 4, it is quite clear that the stacked ensemble produces the best result as compared to bagging and boosting.

The computational complexity of the proposed DIDS depends on the dimension of the input dataset, i.e., number of tuples and number of attributes considered in each tuple as well as on the size of the training and testing sets. Suppose, m = number of training samples and n = number of features. The time complexity of the proposed system is $O(mn)$.

5.3. Comparative analysis with other work

In this section, we have carried out the performance comparison of our proposed system with two other existing DIDSs (Brahma and Panigrahi, 2015; Ronao and Cho, 2016) found in the literature.

The authors in Brahma and Panigrahi (2015) have developed a DIDS (ANFIS-DIDS) that builds the user profiles by collecting data from the database log files. The Sugeno based ANFIS (Artificial Neuro Fuzzy Inference System) classifier is then used to analyze the current transaction and checks for deviation from their normal profiles. They have considered the following attributes – type of SQL command used, number of tables and attributes accessed in a transaction, time slot and location of the user while performing the transaction for representing the user's behavioral pattern. We have conducted the experiments with ANFIS on the synthetic dataset (Panigrahi et al., 2013) by setting the required parameters: number of nodes = 161, number of linear parameters = 448, number of nonlinear parameters = 36, total number of parameters = 484, number of training data pairs = 18 and number of fuzzy rules = 64 as per the heuristics followed in (Brahma and Panigrahi, 2015).

Likewise, in Ronao and Cho (2016) the authors have suggested an anomaly-based DIDS, named as PCA-WRF (Principal Component Analysis – Random Forest with Weighted voting) that mainly focuses on using the Role Based Access Control (RBAC) architecture for finding out the intrusive transactions performed by the intruder. In order to achieve their goal, they have initially used the PCA technique for extracting the necessary and uncorrelated attributes and constructed the user profiles from the queries submitted to the database. After building the profiles, the WRF classifier is used for analyzing the incoming transactions along with the user profiles and classifying the malicious transactions. We have conducted the comparative assessment of their approach with our system using the dataset given in Panigrahi et al. (2013) by following the heuristics described in Ronao and Cho (2016).

Table 4
Performance of Proposed System using Stacked Ensemble Classifier.

Base Learner	Meta Learner	Accuracy (in %)	Precision (in %)	F1_Score (in %)	TPR (in %)	FPR (in %)
Naïve Bayes k-NN Rule Induction Decision Tree	**RBFN**	**92.17**	**95.27**	**86.85**	**90.86**	**3.50**
RBFN Rule Induction Decision Tree k-NN	Naïve Bayes	91.23	95.29	85.04	89.87	3.51
Naïve Bayes k-NN Rule Induction RBFN	Decision Tree	91.83	94.39	86.91	90.50	4.23
Naïve Bayes Decision Tree k-NN RBFN	Rule Induction	91.81	94.42	86.84	90.47	4.21
Naïve Bayes Decision Tree Rule Induction RBFN	k-NN	90.24	85.17	93.07	88.94	11.84

Please cite this article as: S. Subudhi and S. Panigrahi, Application of OPTICS and ensemble learning for Database Intrusion Detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.05.001

Table 5
Comparative Analysis of Proposed Ensemble DIDS with Other Systems.

Performance Metrics (in %)	Ensemble DIDS			ANFIS-DIDS	PCA-WRF
	DT Bagging	DT Boosting	Stacking		
Accuracy	91.55	91.57	**92.17**	85.07	87.01
Sensitivity	86.89	86.90	**90.86**	86.24	89.62
FPR	4.61	4.75	**3.51**	10.15	5.23

We have computed the *Accuracy*, *TPR* and *FPR* for all the DIDSs as well as the ensembles and presented the results in Table 5. Among the bagging and boosting ensembles, we have chosen to show the Decision Tree (DT) based ensemble as it is closely related to the Random Forest classifier described in Ronao and Cho (2016). It is found from the table that the stacked ensemble (with RBFN as meta classifier) is able to detect the intrusive behavior more effectively while keeping the *FPR* at lowest.

Additionally, it is observed that the computational complexity of Brahma and Panigrahi (2015) and Ronao and Cho (2016) are higher as compared to the proposed DIDS. This is because we have minimized the complexity by filtering most of the regular database transactions with the use of OPTICS clustering method. Only the suspicious transactions (transactions that are found to be deviating from the respective user's normal profile) are passed through the Ensemble Learning component for final classification. Hence, the number of transactions processed by the DIDS in our case is quite less as compared to the volume of incoming transactions.

6. Conclusions

Securing the database from the intruders has a significant importance in any organization. In this research, an intrusion detection system has been proposed which involves the application of OPTICS clustering and ensemble learning for identifying intrusive activities in databases. The intrusion detection methodology includes two phases – training and testing. In the training phase, the features of the input dataset are preprocessed. OPTICS clustering is then applied on the preprocessed attributes for building behavioral profiles. In the testing phase, a transaction is initially passed to the clustering module for testing its belongingness with any of the profiles. If the transaction is found to deviate from the normal profiles, then it is further processed by an ensemble learner for confirmation regarding the inconsistency. We have applied three different ensemble methods, namely, bagging, boosting and stacking, consisting of five different classifiers – Naïve Bayes, DT, RI, *k*-NN and RBFN for final decision making of the outlier transactions using information stored in the user activity database.

Experiments were carried out on a large scale synthetic data for analyzing the efficiency of our proposed system. It is found that the*k*-NN bagging ensemble gives better results while comparing the performance of bagging with other classifiers, boosting and individual classifiers. Further, the results obtained from stacking clearly depict that using RBFN as a meta classifier and the other four as base classifiers outperforms all other ensembles. Besides, the superiority of our system is verified by comparing it with two other approaches proposed in the literature.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of Competing Interest

None.

Acknowledgements

The authors are highly grateful to the Department of Computer Science and Engineering, Veer Surendra Sai University of Technology (Formerly University College of Engineering), Burla, Sambalpur, India for providing the required amenities and support for making this investigation successful.

References

Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J., 1999. OPTICS: ordering points to identify the clustering structure. ACM Sigmod Record, vol. 28. ACM, pp. 49–60.
Axelsson, S., 2000. The base-rate fallacy and the difficulty of intrusion detection. ACM Trans. Inform. Syst. Security (TISSEC) 3 (3), 186–205.
Brahma, A., Panigrahi, S., 2015. A new approach to intrusion detection in databases by using artificial neuro fuzzy inference system. Int. J. Reasoning-based Intelligent Syst. 7 (3–4), 254–260.
Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.
Breunig, M., Kriegel, H.P., Ng, R., Sander, J., 1999. OPTICS-OF: identifying local outliers. In: Aytkow, J., Rauch, J. (Eds.), Principles of Data Mining and Knowledge Discovery, Lect. Notes Comput. Sci., Vol. 1704. Springer, Berlin Heidelberg, pp. 262–270.
Bu, S.-J., Cho, S.-B., 2017. A hybrid system of deep learning and learning classifier system for database intrusion detection. In: International Conference on Hybrid Artificial Intelligence Systems. Springer, pp. 615–625.
Cohen, W.W., 1995. Fast effective rule induction. In: Proceedings of the twelfth international conference on machine learning, pp. 115–123.
Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13 (1), 21–27.
Furnell, S., 2004. Enemies within: the problem of insider attacks. Comput. Fraud Security 2004 (7), 6–11. URL: http://www.sciencedirect.com/science/article/pii/S1361372304000879.
Global fraud survey, 2017. 2016 report to the nations on occupational fraud and abuse. copyright 2016 by the association of certified fraud examiners, inc. http://www.acfe.com/rttn2016.aspx, accessed: 30 January, 2018..
Jayaprakash, S., Kandasamy, K., 2018. Database intrusion detection system using octraplet and machine learning. In: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE, pp. 1413–1416..
Kim, S., Cho, N., Lee, Y., Kang, S.-H., Kim, T., Hwang, H., Mun, D., 2013. Application of density-based outlier detection to database activity monitoring. Inform. Syst. Front. 15 (1), 55–65.
Murray, A.C., August 2005. The threat from within network computing. http://www.networkcomputing.com/careers-and-certifications/the-threat-from-within/d/d-id/1213620, accessed:19 February, 2015..
Oza, N.C., Tumer, K., 2008. Classifier ensembles: Select real-world applications. Inform. Fusion 9 (1), 4–20. special Issue on Applications of Ensemble Methods.
Panigrahi, S., Sural, S., Majumdar, A., 2013. Two-stage database intrusion detection by combining multiple evidence and belief update. Inform. Syst. Front. 15 (1), 35–53.
Powers, D.M., 2011. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation..
Quinlan, J.R., 1986. Induction of decision trees. Machine learning 1 (1), 81–106.
Refaeilzadeh, P., Tang, L., Liu, H., 2009. Cross-validation. Encyclopedia of database systems. Springer, pp. 532–538.
Rish, I., 2001. An empirical study of the naive bayes classifier. IJCAI 2001 workshop on empirical methods in artificial intelligence, vol. 3. IBM, New York, pp. 41–46.
Ronao, C.A., Cho, S.-B., 2016. Anomalous query access detection in rbac-administered databases with random forest and pca. Inf. Sci. 369, 238–250.
Schapire, R., 2003. The boosting approach to machine learning: an overview. In: Denison, D., Hansen, M., Holmes, C., Mallick, B., Yu, B. (Eds.), Nonlinear

Estimation and Classification, Lecture Notes in Statistics, vol. 171. Springer, New York, pp. 149–171.
Schwenker, F., Kestler, H.A., Palm, G., 2001. Three learning phases for radial-basis-function networks. Neural Netw. 14 (4), 439–458.
Singh, I., Darbari, V., Kejriwal, L., Agarwal, A., 2016. Conditional adherence based classification of transactions for database intrusion detection and prevention. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, pp. 42–49.
Wei, C.-Y., Cai, X.-Q., Liu, B., Wang, T.-Y., Gao, F., 2018. A generic construction of quantum-oblivious-key-transfer-based private query with ideal database security and zero failure. IEEE Trans. Comput. 67 (1), 2–8.
Wolpert, D.H., 1992. Stacked generalization. Neural Netw. 5 (2), 241–259.
Yesin, V.I., Yesina, M.V., Rassomakhin, S.G., Karpinski, M., 2018. Ensuring database security with the universal basis of relations. In: IFIP International Conference on Computer Information Systems and Industrial Management. Springer, pp. 510–522.

Please cite this article as: S. Subudhi and S. Panigrahi, Application of OPTICS and ensemble learning for Database Intrusion Detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.05.001