Hindawi
Security and Communication Networks
Volume 2020, Article ID 5874935, 14 pages
https://doi.org/10.1155/2020/5874935

Research Article

Preserving Privacy in Multimedia Social Networks Using Machine Learning Anomaly Detection

Randa Aljably,[1,2] Yuan Tian,[3] and Mznah Al-Rodhaan[2]

[1]*Computer Department, Shaqra University, Shaqra 11911, Saudi Arabia*
[2]*Computer Science Department, King Saud University, Riyadh 11543, Saudi Arabia*
[3]*School of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, China*

Correspondence should be addressed to Yuan Tian; ytian@njit.edu.cn

Received 30 January 2020; Revised 23 May 2020; Accepted 3 June 2020; Published 20 July 2020

Academic Editor: Zhaoqing Pan

Nowadays, user's privacy is a critical matter in multimedia social networks. However, traditional machine learning anomaly detection techniques that rely on user's log files and behavioral patterns are not sufficient to preserve it. Hence, the social network security should have multiple security measures to take into account additional information to protect user's data. More precisely, access control models could complement machine learning algorithms in the process of privacy preservation. The models could use further information derived from the user's profiles to detect anomalous users. In this paper, we implement a privacy preservation algorithm that incorporates supervised and unsupervised machine learning anomaly detection techniques with access control models. Due to the rich and fine-grained policies, our control model continuously updates the list of attributes used to classify users. It has been successfully tested on real datasets, with over 95% accuracy using Bayesian classifier, and 95.53% on receiver operating characteristic curve using deep neural networks and long short-term memory recurrent neural network classifiers. Experimental results show that this approach outperforms other detection techniques such as support vector machine, isolation forest, principal component analysis, and Kolmogorov–Smirnov test.

1. Introduction

Anomaly detection is an important paradigm in information security, especially in technically challenging platforms like multimedia or online social networks (OSNs). To detect anomalous users in social networks, proposed models identify a pattern of what is considered normal behavior, by examining a series of events either from each individual user or from a collected set of users [1, 2]. Intrusion detection system (IDS) is used to identify deviations or anomalies in entities' behavior. The detection system identifies threats such as fraud, disturbance, network intrusion, information leakage, and privacy violation [2]. Among IDSs, machine learning techniques showed promising results in detecting users with high accuracy and low false rates. Supervised machine learning approaches the detection problem from a statistical point of view [3]. Bayesian classifiers detect any change in communication patterns between users and treat it as a counting process following time increments [4], while unsupervised deep neural networks extract features from the system logs, calculate the probabilities for these feature vectors to make future predictions, and detect abnormalities [5].

Despite the major achievements that machine learning techniques have achieved in detecting outliers, there still exist some challenges that impede preserving the user's privacy in OSNs [6], namely, the lack of cooperation between detection algorithms and other security measures such as access control [7]. The isolation deprives the detection models from using available user attributes to rule out false alarms.

In this study, we extend an attribute-based access control model to act as an extension to the detection model, and further examine anomalous users using each user's treats to adjust authorization privileges. Furthermore, the algorithm will invoke user awareness in each set of connected users.

Simulation results compare the Bayesian, deep neural network (DNN) [8] and long short-term memory recurrent neural network (LSTM-RNN) [4] performance to other detection systems such as Kolmogorov–Smirnov test (KSE) [9], support vector machine (SVM) [10], principal component analysis (PCA) [11], and isolation forests [12]. The results showed that the algorithm meets the privacy requirement of preserving user's confidentiality and availability on the OSN.

The contributions of this work are summarized as follows:

(1) We define a dynamic approach to incorporate the services provided by machine learning detection models to enhance access control performance and improve the overall performance without increasing complexity. We show that the resulting models' security level is higher than the integrated components.

(2) We demonstrate how the proposed integrated model contributes to the preservation of privacy by alerting users and detecting and preventing abnormal activity. If a user is labeled as anomalous, they will be denied access quickly and other users will be warned to change their security settings.

(3) The model uses rich and fine-grained security policy techniques to guide the anomaly submodel in order to meet the privacy preservation goal.

Once granted access, the model tracks the behavior of every entity in the OSN [13]. If anything deviates from the normal behavior baseline, it is flagged and logged as anomalous. Furthermore, the activity pattern is fed into the system online.

The rest of the paper is organized as follows: in Section 2, we present a review of studies that have previously approached the privacy problem and proposed solutions. In Section 3, we briefly formulate the problem at hand. In Section 4, we describe the proposed algorithm and privacy analysis. Performance evaluation is reported in Section 5. We conclude the paper and present our future work in Section 6.

2. Related Work

In this section, we shed some light on important research into access control and anomaly detection. Usually, information protection relies primarily on basic mechanisms, such as authentication, access control, and auditing. However, attempts to combine different security structures have demonstrated that the protection level can be increased [14–16].

The studies in [17–19] combine generic access control models with intrusion detection layers. The approaches involve specifying security policies and extend them to identify intrusions and suspicious behavior occurring at the application level. These policies are responsible for guiding intrusion detection and response. The proposed systems enforce such policies to monitor access requests to vulnerable applications and support the detection of attacks that are not visible at the application level.

2.1. Attribute-Based Access Control in OSNs. Various researchers have focused on analyzing the online behavior of users and evaluating the amount of disclosed information or the usage of privacy settings. The results of such approaches showed that only a minimal percentage of users change their highly permeable privacy preferences [20].

Some approaches that aimed to detect security violations focused on specific attributes within the model. Previous studies [2, 21, 22] proposed using location-centric attribute-based access control (ABAC) or relationship-based access control (ReBAC) to enhance their capabilities and allow finer-grained controls. Both ABAC and attribute-based encryption (ABE) are used for end-to-end confidentiality. Unfortunately, the latency added by access control when using a large database is higher, because it involves one database lookup apart from user authentication and authorization. This processing delay and the lack of multiple enterprise investigations are considered drawbacks of these systems [2, 6, 23].

2.2. Anomaly Detection. In recent years, deep learning has drawn attention as one of the most popular machine learning (ML) techniques, producing remarkable results for a given range of supervised and unsupervised tasks. The main reason that deep learning outperformed its peers is its ability to automatically learn high-level representations from the data with little or no need of manual feature engineering.

In this study, we propose a model to detect anomalous behavior. It is based on an unsupervised neural network that can identify unknown new attacks. It feeds on knowledge provided in an online and offline manner [24]. Then the model uses the network's classification to prevent unauthorized access to shared resources. The research in [25, 26] showed that RNN and LSTM-RNN outperform anomaly detection baselines such as PCA isolation forest and SVM with an average anomaly score in the 95.53 percentile.

The work in [27] proposed using exponentially weighted moving average which narrows the gap between local diversities in the data stream. The model was not affected by anomalous fluctuations in the data because it learned the parametric statistical model of the changing stream distribution.

Gavai et al. [28] examined anomaly detection using social and online activity data of enterprise employees. The authors compared two detection approaches. The first is an unsupervised approach that used an isolation forest. The model statistically identifies malicious behavior with respect to extracted features. The second is a supervised approach that employed an expert developed classifier. The first model achieved a ROC score of 0.77 while the classification accuracy of the supervised approach was 73.4%.

In more detail, [26, 29] used LSTM-CNN for detecting user's anomalous behavior as an insider threat. The LSTM extracted temporal features from the user's actions and then converted them to fixed-size feature matrices. The CNN used

these feature matrices to detect anomalous behavior which it considered insider threat. Experimental results showed AUC of 0.9449. [29].

As promising as anomaly-based IDS (ABIDS) is, it is still facing many challenges that, if not well resolved, may impede its fast growth. ABIDS collects data from networks and systems for analysis and detection. The collected datasets have an uneven number of anomalous behavior compared to normal behavior, resulting in unbalanced classes. In addition, the anomalies are not labeled and they are distributed all over the dataset. Furthermore, it is normal for a user's behavior to evolve and change over time. However, this change may be misinterpreted by the detection system as anomalous.These challenges raise great concerns for ABIDS in protecting information and detecting threats.

3. Problem Formulation

3.1. System Model. Traditionally, the intrusion detection system (IDS) and access control system (ACS) work independently of each other. The isolation and lack of coordination prevent the detection of sophisticated high-level attacks and slow the real-time response to ongoing attacks. Also, this independence may cause the IDS to falsely classify activities and perhaps terminate authentic processes or deny services to legitimate users.

In this paper, we propose a model that leverages both systems to strengthen intrusion detection (ID) in a way that effectively and accurately helps keep unwanted visitors outside the multimedia social network (MSN), while allowing the administration to monitor and control access inside the MSN. The architecture of our model is shown in Figure 1.

In Figure 1, the cloud represents the MSN. In the access control part of the model, the focus is on specifying (organization- or application-level) security policies, as is done with dynamic policy techniques. The attribute-based access control submodel compares access requests with security policies. The process is extended with the ability to identify intrusions and suspicious behavior in the IDS component, on the basis of Bayesian anomaly detection methods. Thus, this approach builds a complete picture of the attacks. The objective of this model is to support the prevention and detection of and the automatic response to intrusion attempts.

The remaining parts of the model are illustrated in Figure 1, and their functions are as follows:

(1) The security database contains up-to-date, accurate, multisource information related to security events.

User activity profiles: user activities are monitored to construct a pattern of regular behavior. This pattern is audited and assembled into a user profile for future comparison with any unusual or appropriate behavior.
Signature actions: these contain information about known misuse or intrusion scenarios.

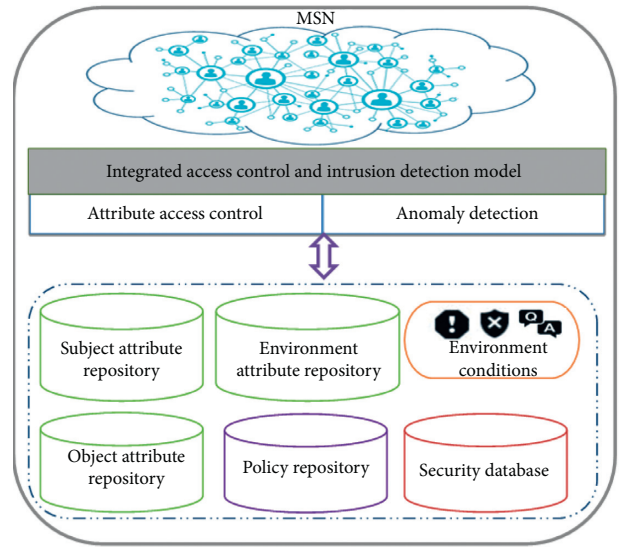

Figure 1: The integrated Bayesian attribute-based access control (B-ABAC) architecture consisting of multimedia social network (MSN), where the integrated access control and intrusion detection model requests and stores data in the following repositories: green repositories represent attribute repositories, the purple repository is the policy repository administrated by a policy manager, and the red repository is the security database repository.

Account auditing: periodic auditing of activities is performed to keep track of unauthorized activities or accounts.

(2) A policy repository stores three types of policies:

An access control policy: this specifies how to handle a user's access request and how to handle unauthorized access. It also describes the environmental conditions that must be met to authorize access.
A misuse/anomaly detection policy: this specifies the parametric values to be used in the detection process, specifies the implementation of detection methods, compares a user's activity with their user profile and attack scenarios, and specifies procedures to follow when an unauthorized access request or an anomalous user is detected.
A privacy policy: this specifies how to protect the confidentiality of data, for example, by encrypting attributes or blocking access to resources.
A response policy: this defines the response actions and countermeasures to access requests (authorizing a request, denying a request, blocking a user, blocking a feature, disabling a profile, generating an alarm, or sending warning messages).

(3) Other repositories in the model are as follows:

Object attributes repository: this contains attributes describing the object such as the security level of an object, its date, owner, and content.
Subject attributes repository: this contains attributes describing the subject requesting access to a

specific object. It stores attributes such as the subject's role and clearance.
Environment attribute repository: this contains technical details associated with the request, including the server location, time, and IP address.
Communication interfaces: these represent the mechanism that supports bidirectional communication between two agents.

4. The Proposed Algorithm

The algorithm combines attribute-based access control with the classification power of a Bayesian statistical model to assign permissions to application users for access control requirements. In this section, we describe the proposed algorithm by describing the procedure it follows, and then we explain the algorithm design.

4.1. Design Rationale. When a subject or user requests access to the MSN or to a specific object or requests permission to act, the MSN refers to the algorithm for a decision. The algorithm evaluates the requestor's status from previous anomaly detection rounds. If there is no previous information, it invokes the Bayesian classifier for anomaly testing. Then the algorithm passes the result to the access control submodel, which checks the request against the rules, subject and object attributes, and environmental conditions found in the policy repository to compute a decision, as shown in Figure 2. The figure illustrates the steps the algorithm follows to check the authorization of each request.

This work adopted a multistage approach. In this approach, the dynamic anomaly detection submodel scans the network for anomalous nodes. Then, the timing process models the number of communications between individuals over time; every increment will follow a Bayesian discrete time counting process with conditionally independent increments. The number of interactions between each pair of individuals is the equivalent weight of their relationship in the network.

Typically, anomaly detection involves processing the edge connections between the graph's nodes to determine whether individuals (represented as nodes) are connecting with each other more or less frequently than usual or are creating new connections with other individuals. Nonetheless, the focus in large dynamic networks, such as social networks, is the application of methods with acceptable computational demands.

Bayesian models deal with discrete and continuous variables; such discrete variables can represent the time at which we observe the networks.

4.2. Construction. The following steps describe the algorithm and the roles of the MSN and the users. The attribute-based access control subsystem expands policy evaluation by comparing every received access request with security policies. Thus, there needs to be an authorization policy to grant or deny access rights to objects. The decision is accomplished using three types of environmental conditions: preconditions that must be true before the execution of access authorization, midconditions that must be true during the execution of access authorization, and postconditions that must be true by the completion of access authorization (otherwise, if the postconditions are false, they will trigger postexecution actions).

Each request is assigned an authorization status (request result), and it has the values of "Yes" when it is granted, meaning that the request fulfilled all the mentioned conditions; "No" if it did not match them and was denied; and "*X*" in cases of uncertainty [14]. In addition to other variables in the state records, the model stores the request results, which are classified as prerequest results, midrequest results, and postrequest results, corresponding to each respective condition set.

The proposed model follows three steps: before execution of the access rules, during execution, and after execution:

(i) Step 1: when the subsystem receives a request, it inspects the operation, metadata (user ID, type of action, access time), and specific object information provided in the request. It then invokes the Bayesian anomaly detection subsystem for a diagnosis of user behavior. It accesses the policies and authorization rules, saved in the policy database, to retrieve the objects' security policies, including the state records and conditions for comparison. The request is granted if sufficient credentials are provided, and the precondition parameters are passed to the next phase to execute the permission.

(ii) Step 2: after request authorization on the basis of midconditions, the flow moves to the third phase.

(iii) Step 3: any necessary actions, such as sending warning messages to users, are executed.

We provide an example of a scenario for the implementation of these three phases in the algorithm description.

The integrated Bayesian network starts with a network scanning process [4].

(i) Step 1: model the number of communications between pairs of nodes as a Bayesian counting process. Since each pair is parallel, the increments can be conditionally independent. The algorithm uses a Bayesian probability function:

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}. \tag{1}$$

Here $P(c \mid x)$ is the posterior probability, $P(c \mid x)$ is the likelihood, $P(c)$ is the class prior probability, and $P(x)$ is the predictor prior probability. A discrete random variable x is said to have a Poisson distribution with parameter $\lambda > 0$, if, for $k = 0, 1, 2, \ldots$, the probability mass function of x is given by

$$f(k; \lambda) = \Pr(x = k) = \frac{\lambda^k e^{-\lambda}}{k!}. \tag{2}$$

For numerical stability, the Poisson probability mass function should be evaluated as

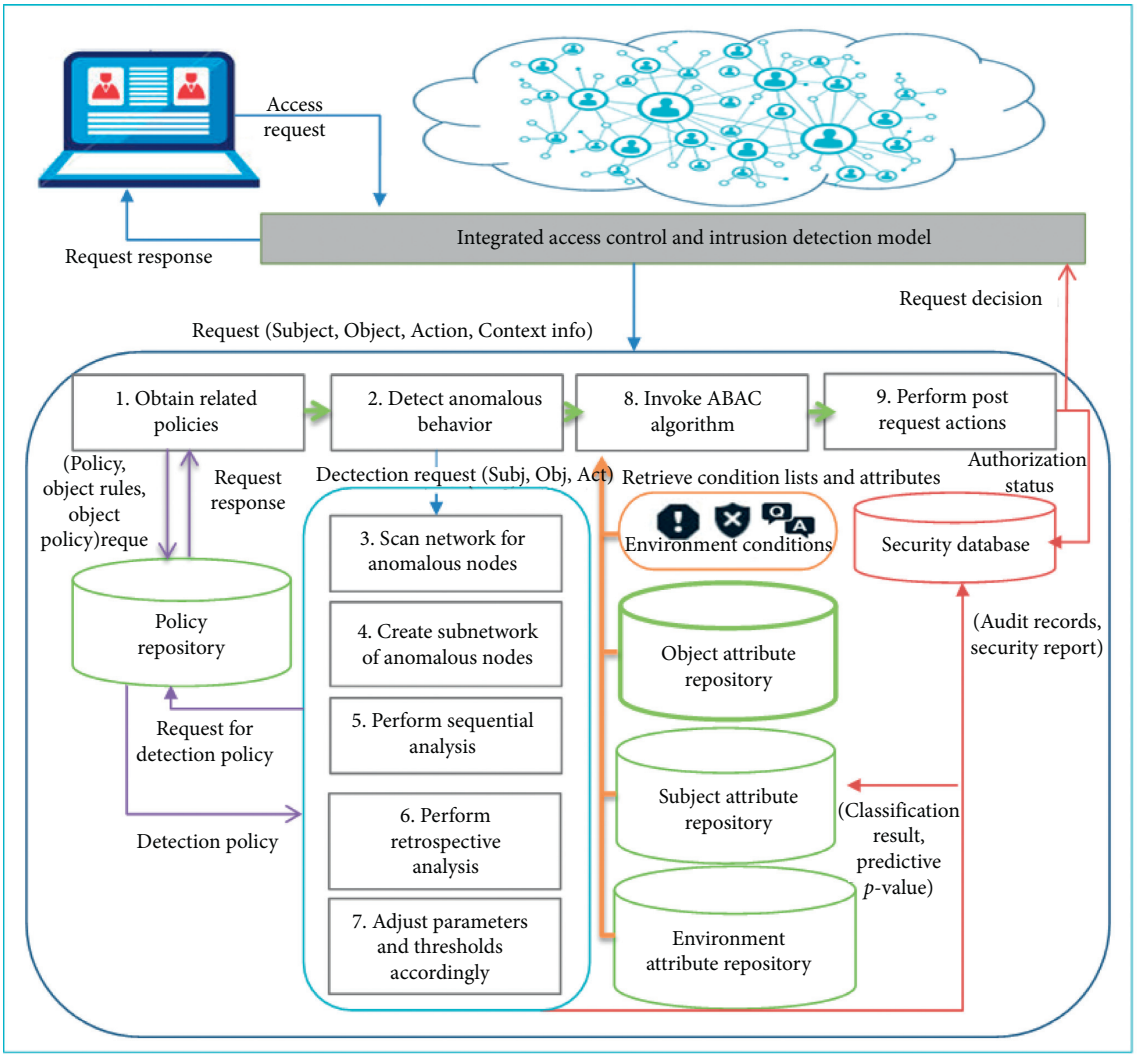

FIGURE 2: The implementation details of the B-ABAC. The access control submodel invokes the Bayesian detection algorithm at step 2 to update its authorization decisions based on the detection results.

$$f(k;\lambda) = \exp\{k \ln \lambda - \lambda - \ln \Gamma(k+1)\}. \quad (3)$$

Choose $P_{ij} = \text{Poisson}(\lambda_{ij})$ for an unknown rate parameter $\lambda_{ij} > 0$. Choose a gamma prior distribution for λ_{ij}; this ensures the posterior predictive distribution for a future period is calculable as a simple ratio of Poisson gamma mass functions.

Steps to apply Dirichlet process: when given a measurable set S, a base probability distribution H, and a positive real number α, the Dirichlet process $\text{DP}(H, \alpha)$ is a stochastic process whose sample path is a probability distribution over S.

For any measurable finite partition of S: $(B_i)_{i=1}^n$, and if $X \sim \text{DP}(H, \alpha)$, then

$$(X(B_1), \ldots, X(B_n)) \sim \text{Dir}(\alpha H(B_1), \ldots, H(B_n)). \quad (4)$$

Here Dir is the Dirichlet distribution. The notation $X \sim \text{DP}(H, \alpha)$ means that the random variable X is distributed according to the distribution $\text{DP}(H, \alpha)$, i.e., according to the Dirichlet process with the parameter base distribution H and the real number α.

The Dirichlet distribution of order $k \geq 2$ with parameters $\alpha_1, \ldots, a_k > 0$ has a probability density function with respect to the Lebesgue measure in the Euclidean space $R^{k=1}$ given by

$$f(x_1, \ldots, x_k; \alpha_1, \ldots \alpha_k) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} x_i^{\alpha_i - 1}. \quad (5)$$

(ii) Step 2: represent the number of communications as the weights assigned to communicating nodes in the network.

(iii) Step 3: since some of the time intervals might have a communication mean of zero (indicating that the nodes are not communicating at this time), there is a

need to use independent models for the hurdle model. These independent models provide variables to determine whether pairs are communicating.

(iv) Step 4: perform an individual-based analysis, as described in [30, 31]; in this step we assume that $dN_{ij}(t)$ is the number of communications from i to j starting at time 0 until discrete time t. The increments determine the out-degree and in-degree of node i, and we represent the number of outgoing communications as

$$N_{i.}(t) = \sum_{j \neq i} N_{ij}(t). \tag{6}$$

The incoming communications over time for individual i are represented as

$$N_{i.}(t) = \sum_{j \neq i} N_{ji}(t). \tag{7}$$

We then calculate the total activity analysis by finding the degree sum of the network over time $N_{..}(t)$.

(v) Step 5: from the population, a sample of size n is selected, in which the random variable of interest X is the number of anomalous individuals in the sample, M is the number of anomalous individuals in the population, and N is the set of communicating individuals:

$$P(X = x) = \frac{(x)\binom{N-M}{n-x}}{\binom{N}{n}}. \tag{8}$$

Since the proposed detection model is based on a statistical approach, it has the advantage of pure computation. Each node pair will be examined in isolation by assuming independence of the processes in certain circumstances. When the conjugate Bayesian models are chosen carefully, they allow this inferential process to be analytically tractable.

In case the anomaly detection is done using a DNN, the steps would be as follows:

(i) Step 1: extract and aggregate numeric features from the auditing records and the user's metadata. This process produces one feature vector for user/day.

(ii) Step 2: the features vectors are normalized using a batcher to feed the minibatches into the network to create a set of DNNs or LSTM-RNNs.

(iii) Step 3: the networks learn normal behavior to predict the next vector in each sequence.

(iv) Step 4: the networks calculate the joint and conditional probability for each vector. Let P_θ be the joint probability computed over user count $(\widehat{x}_t^u)$ and categorical features. R_t^u denotes the first of the categorical features which is the role of user u at time t, and S_t^u denotes the last categorical feature which is the supervisor. Then,

$$P_\theta\left(x_t^u \mid h_{t-1}^u\right) = P_\theta\left(\widehat{x}_t^u, R_t^u, \ldots, S_t^u \mid h_{t-1}^u\right). \tag{9}$$

(v) Step 5: the model generates anomaly scores, which are used to rank user/days from most to least anomalous.

(vi) Step 6: the ABAC subsystem receives the listed anomalous users concerning the target requestor and the surrounding network of users. According to the access control and response policies after inspecting the related attributes, if the user proves to be anomalous, they are denied access and flagged, and the pattern of behavior is stored for future reference. Otherwise, they will be authorized.

4.3. Algorithm Design. First, we list the explanation of the symbols used in the attribute-based access control (ABAC) submodel, Table 1.

Next, we introduce the algorithm that the attribute access control submodel follows to decide on the authorization request it received from a particular user. The algorithm is implemented in three parts. The first part is preexecution, which occurs before authorization as demonstrated in Algorithm 1. While the remaining two parts occur during and after the authorization decision, as seen in Algorithm 2.

5. Integrated Model Results

5.1. Descriptive Scheme. Our goal was to apply the anomaly detection methods to MSNs platforms such as Facebook, Twitter, LinkedIn, Instagram, and Snapchat. We chose Facebook because it was deemed the top most used OSN in the year 2017. The proposed security scenario classifies the type of communication between nodes. Possible interactions include sending and receiving private messages, friend requests, and replies; joining friends' lists; publishing wall posts or commenting on posts; liking and disliking posts; and tagging or being tagged in pictures.

In this experiment, we used MATLAB 2019b to detect anomalous behavior according to the number of messages between users. The attributes selected from Facebook API for policy enforcement are shown in Figure 3. These attributes are mostly connected to the main entities in each user account, such as the social network of other related users, attributes related to the messaging feature, and what the user can do with these messages.

5.2. Access Control Results. One of our objectives in this scenario is to differentiate between innocent users who are facing hard times (like Haitian victims) and spammers or feature abusers. The secondary objective is to make use of most of the attributes saved in the OSN and allocate the minimum amount of space for ABAC attributes. Preserved allocation allows for faster retrieval, better performance, and

Table 1: Description of symbols used in ABAC submodel.

Symbol	Description
R	Request operation
Cntxt-info	Contextual information (user identity)
Pre_Con_L	A list of attributes and conditions that must be true to approve a request
Mid_Con_L	A list of attributes and conditions that must be true between receiving a request and processing it
Post_Con_L	A list of attributes and conditions that must be true after processing a request. It is used to activate postexecution actions
Result_Con_L	Attributes and conditions modified according to the request result
Authorization_status	Evaluation status of the conditions; the request decision (Yes: authorization granted, No: not granted)
Post_ authorization _status	Evaluation status of the postconditions (Yes: granted, No: not granted, *X*: uncertain)

```
Input: receiving Req = R (Subj, Obj, Action, Cntxt-info)
Output:
 (1) Authorization_status {Yes, No}
 (2) Request _decision = {Accept, Reject}
 (3) Procedure:
 (4) Preexecution Algorithm:
 (5)    Access Policy Repository
 (6)    Set_Policy = Get_object_policy_info (Obj)
 (7)       If (Policy = = NULL)
 (8)          Set_authorization_status = (NO)
 (9)          Request_decision = "Reject"
(10)       Else
(11)       Loop through Pre_Con_L items
(12)          Set Pre_conditions = Get_preconditions (Pre_Con_L)
(13)          Set Post_conditions = Get_postconditions (Post_Con_L)
(14)             If (Pre_conditions = = NULL)
(15)                Set Pre_authorization_status ("Yes")
(16)             Else
(17)             Pre-authorization_status = Check_authorization (Pre_conditions)
(18)             Post_result = Check_authorization (Post_conditions)
(19)             End
(20)       End
(21)       If (Pre_authorization_status = ("Yes")
(22)          Set authorization_status_L = (Post_result, Pre_authorization_status)
(23)       Else
(24)          Request_decision = "Reject"
(25)          Invoke Privacy_policy (Req, authorization_status_L)
(26)          Invoke Response _Policy (Req, authorization_status_L)
(27)       End
(28)    End
(29)    If (authorization_status_L = ("Yes, Yes")
(30)       Call During execution Algorithm
(31)    Else
(32)          Invoke Privacy_policy (Req, authorization_status_L)
(33)          Invoke Response _Policy (Req, authorization_status_L)
(34) End
```

Algorithm 1: Preexecution phase of the ABAC submodel.

minimal memory requirements. The third objective is to eliminate the role of the data normalizer as a translator between IDS and ABAC, thereby establishing a direct invoke-response relationship.

After logging in, once a user tries to access their account or any of the OSN features, the access control model will run the scenario and check the user's communication pattern. If abnormal activity is detected or the user is found in the list of anomalous users, further examination is conducted to determine the access rights.

In our dataset, from the list of anomalous users obtained from the anomaly detection model, two users were found to be anomalous. One of these users had two different accounts on FB to communicate with other users. This user was

```
(1) During-execution Algorithm
(2)    Access Object Attribute Repository
(3)    Access Subject Attribute Repository
(4)    Access Environment Attribute Repository
(5) Set Mid_conditions_L = Get_Mid_conditions (Obj, Action, User_info)
(6) Loop through Mid_Con_L until check_authorization (Mid_condition_L) = "False"
(7)          Authorization_status = Check_authorization (Mid_condition_L)
(8) End
(9) If (Authorization_status = (No)
(10)       Invoke Privacy_policy (Req, authorization_status_L)
(11)       Invoke Response _Policy (Req, authorization_status_L)
(12)       Set_authorization _status = "No"
(13)       Request_decision = "Reject"
(14) Else
(15)       Set_authorization _status = "Yes"
(16)       Request_decision = "Accept"
(17)       Call Post–execution algorithm
(18) End
(19) Post–execution Algorithm
(20) If (Post_condition = Null) fig
(21)       Set_authorization_status ("Yes")
(22) Else
(23)    Loop through set of Post_Con_L
(24)    Authorization_status = Check_authorization (Post_conditions_L)
(25)    Invoke Response _Policy (Req, authorization_status) Post_execution_ action (Result_Con_L)
(26)    End
(27)    END
```

Algorithm 2: During- and postexecution algorithms.

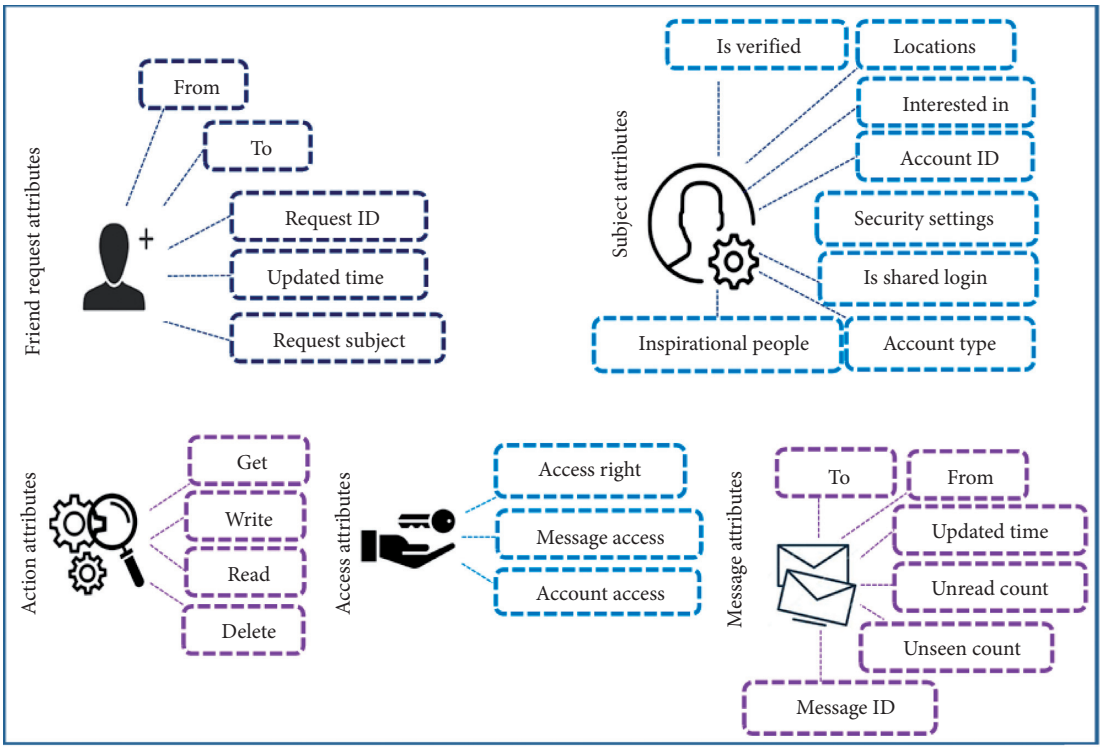

Figure 3: Facebook attributes used in tests of the B-ABAC.

sending messages in an abnormal pattern. In this case, the proposed attribute-based access control subsystem sent him a message, as shown in Figure 4.

For the users receiving messages from an anomalous user, if the ABAC submodel determines that the user has weak security settings, the model warns the user with the message in Figure 5.

Five of the remaining individuals were declared to be significant in the social network for misusing the messaging feature. Once abnormal activity is discovered, it is a straightforward task to identify the opposite communicators and analyze their activities or observe their security settings. If the suspicious user's account does not have sufficient information for a decision, the model informs users and asks them for further information, as shown in Figure 6.

To measure the accuracy of this model, we posted a questionnaire to the users in our dataset to identify their present status in FB (restricted or not restricted from sending messages). We eliminated those who were blocked or restricted for reasons other than over-messaging. A remarkable observation from the survey results was that those who were blocked or restricted from accessing any feature on Facebook answered that they did not know why they were restricted or blocked, even if their account was disabled, which indicates a considerable gap in communication between the website and its users.

According to the users' answers, we linked the labels of their account situations and compared them with the labels resulting from our proposed models.

The results show that those who were restricted by our approach were likewise restricted by FB, but FB restricted an additional user that our model did not. The reason for FB's restriction was that the user showed a sudden increase in messages sent to specific accounts. In our system, this particular user was not restricted because the ABAC did not rule them as suspicious on the basis of their account information and neither did the other communicating parties. After manual inspection, the user was, in fact, harmless, and the increase in his messaging rate was due to them studying abroad, leading to increased communication with their family after getting an Internet connection.

5.3. Privacy Analysis. In this section, we show how our design functions to preserve the privacy of the MSN and users' information and how it prevents unauthorized users from disclosing confidential information.

The privacy of the SN's data: when social networks attempt to access a user's history of preferences, the request is compared with a set of conditions set by the user; if a condition prohibits the collection of data by any subject other than the user himself, the request should be rejected, and the user notified.

The privacy of users' data: any requests from unauthorized users who have been labeled anomalous by the detection system are prevented from accessing the data, and the user is notified and advised to change their security settings.

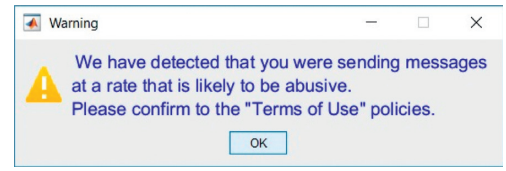

Figure 4: Anomalous user's warning message.

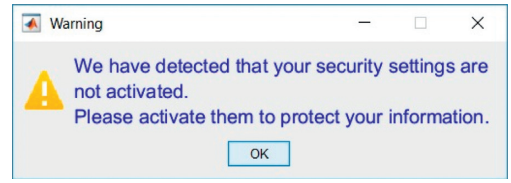

Figure 5: Warning message for communication recipients to activate their security settings.

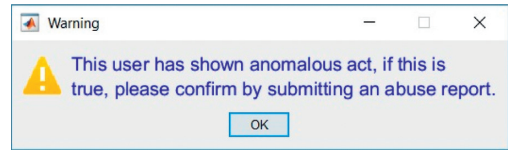

Figure 6: A message is displayed to obtain additional information from other parties.

The continuity of access to data for authorized users: the accuracy of detecting anomalous behavior on the Facebook dataset using the Bayesian submodel reached 89%. The list of anomalous users resulting from the Bayesian submodel was further inspected by the ABAC submodel to evaluate each user independently. Some of the users on the anomalous list were declared to be harmless and thus would be given permission to continue using the platform.

As described in this section, we conducted experiments by applying the integrated anomaly detection methods with access control models to real and simulated Facebook data. The first dataset contained threads of message sent between 25 November 2015 and 4 December 2015. There were 400 individuals located at 30 places in the network during data collection. However, due to the scattered nature of social network graphs, not all of these individuals were connected. Therefore, the dataset contained 10,000 of the possible 79,800 node connections.

In addition, we applied unsupervised NN methods to the CERT Insider Threat Dataset v6.2 [32]. It includes log files collected from an organizations network, with data such as http traffic, e-mail traffic, operations done on files, logon and logoff operations, and the usage of an external storage device, in addition to categorical features (attribute metadata) such as department, functional unit, role, and project. There were 135,117,169 events collected over 516 days from 4,000 users.

Messaging activity was chosen for detecting anomalies in user behavior because of its capability of structuring a user's activity profile [31, 32]. During the first analysis phase, our approach checked all 30 locations for anomalous users by applying a multinomial model with a sequential Dirichlet

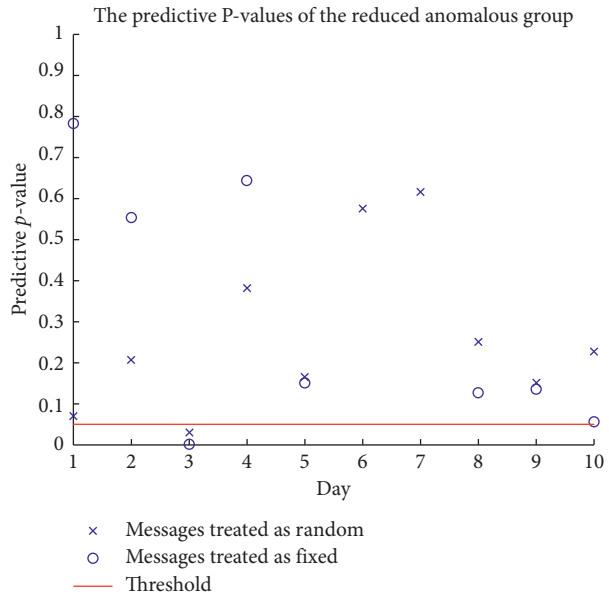

Figure 7: The p values of the anomalous nodes under the multinomial model, reproduced from Aljably et al. [36].

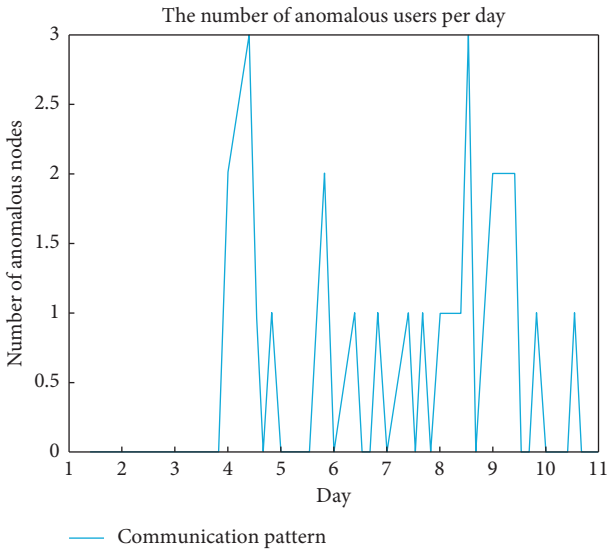

Figure 8: Number of anomalous nodes in each daily time interval under the Markov–Bernoulli model, reproduced from Aljably et al. [36].

process and an uninformative negative binomial base measure [33]. The Bernoulli process and Markov chain [34, 35] were used on all network users, with mean values of [0.63, 0.48], respectively, and a threshold of 0.05, to get a better understanding of the messaging patterns and their variability. This phase monitors the activities of anomalous nodes as a group. The p values for this group from the multinomial model can be seen in Figure 7. The circular points are the p values when the number of messages is treated as random; the crosses consider the number of messages as a known quantity.

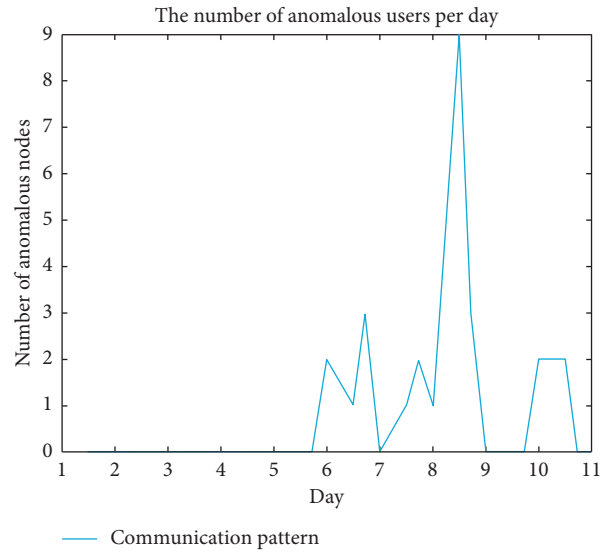

Figure 9: The number of anomalous nodes after dividing the day to 3 subintervals that have an approximate equal number of messages.

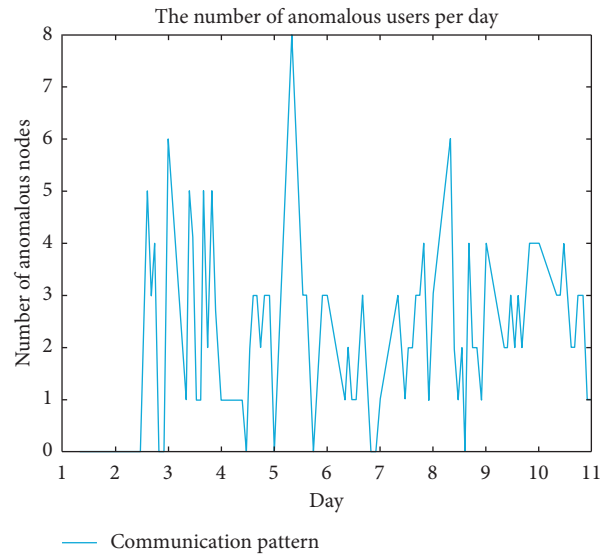

Figure 10: The number of anomalous nodes after dividing the day to 10 subintervals that have an approximate equal number of messages.

In Figure 8, the number of possible anomalous users who had highly negative martingale residuals increased on the fourth, eighth, and ninth days. These results suggest that the use of sequential analysis incorporated all the data in an approach to real-time anomaly detection.

The model used both the end-time of each interval and the anomalous nodes found in stage 1 to reduce the network's size and find the graph of users contacted by the anomalous users. The division of each day into a specific number of intervals plays a role in the detected number of anomalous nodes and the reduced subnetwork. The network was analyzed over a sequence of discrete time series, each giving a localized view of network communication, and the complete view was obtained by accumulating these series. By

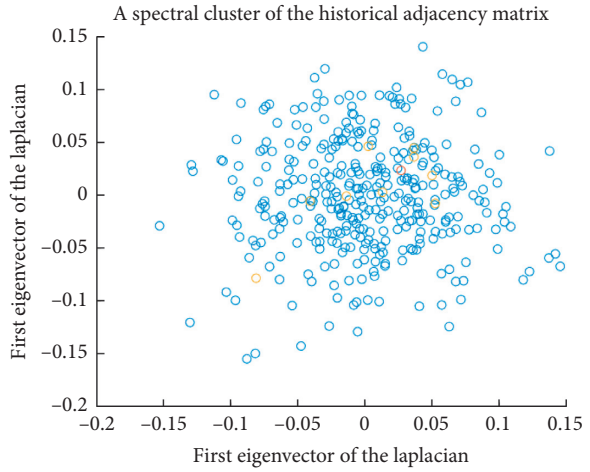

Figure 11: The sequential analysis predictive p values in the subnetwork of anomalous nodes and nodes which they communicated with using 3 intervals, respectively; the color represents the calculated p values.

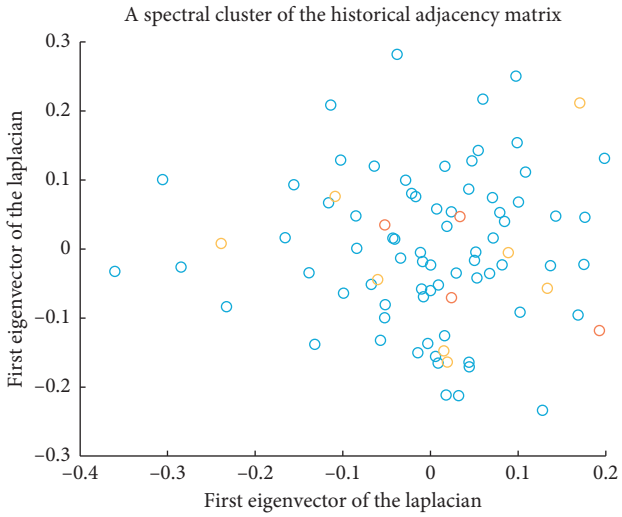

Figure 12: The sequential analysis predictive p values in the subnetwork of anomalous nodes and nodes which they communicated with using 10 intervals, respectively.

using 3, 5, and 10 intervals for ten days, we obtained 30, 50, and 100 homogeneous periods for the analysis, as shown in Figures 9 and 10. When using 3 intervals, the Bayesian classifier did not flag any anomalous activity in the first 5 days; on the other hand, the classifier flagged more anomalous activity when using 10 intervals. The increased anomalous classification increased the false alarm rate.

The spectral cluster plot made using two components of the symmetric Laplacian of the historical adjacency matrix is given in Figures 11 and 12 for 3 and 10 subintervals. The data is centered around the origin; as seen in these two figures, we are interested in the relative location between data points, which changed when using different subintervals, due to the change of the spectral clustering of the p values calculated by the Bayesian classifier. The x-axis and y-axis represent the

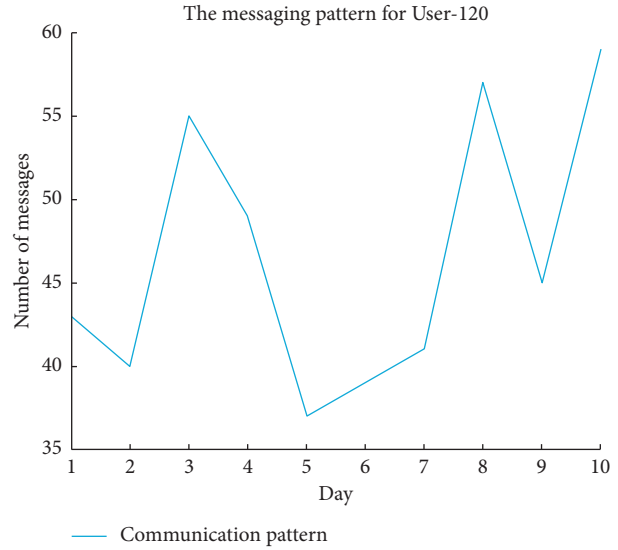

Figure 13: Visualization for User-120 communication pattern during the 10-day period.

values when each user is projected onto Laplacian-eigenvectors.

Using an example of an anomalous node from the network analyzed in Figures 9 and 10, we briefly present the communication pattern of User-120. In the second week (ending on the 4th of December 2015), the user showed higher activity than usual (although not their highest ever). The highest peak of connectivity was in the first week starting from the second day, as shown in Figure 13.

User-0, for example, is a very famous and popular account on Facebook (FB) but was not detected as anomalous by our method. Looking at the communication log of the set of detected anomalous nodes reveals User-0 to be the most frequent communicator in the network with this group.

We compared these results with results obtained by applying Kolmogorov–Smirnov tests. These tests are frequently used for detecting changes in observations with applications in networks and radioactivity, among others. The procedure was adapted from [33], by first resampling at random, with replacement from the 400 statistics, and then by fitting a gamma distribution to this new sample and calculating its 99th percentile. Lastly, we repeated this process until reaching a stable result (100 times). Using the parameters $N(0, 1)$ where N is the average Kolmogorov–Smirnov (KS) statistic and H is the sum of 1000 squared standard Cauchy random variables, we computed the outlier score for each p-dimensional data point. In Figures 14 and 15, the highest scores between [0, 1] are possible outliers. These scores closely correspond to the anomalous nodes found by the Bayesian method.

Traditionally, neural networks are fed sequences of batches or minibatches. Then, they use back propagation to update the weight in order to reduce predictive error. This training iterates over the data until convergence, and then it is tested on new data to classify it. However, this approach is not suitable for anomaly detection purposes. The DNN and LSTM-RNN hyperparameters were tuned using random search, with 256

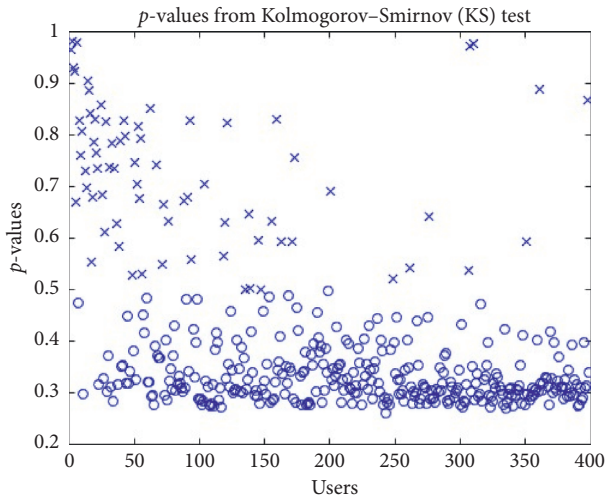

Figure 14: The p-dimensional data points of the anomalous nodes. The circular points are the normal nodes; the crosses are considered anomalous nodes.

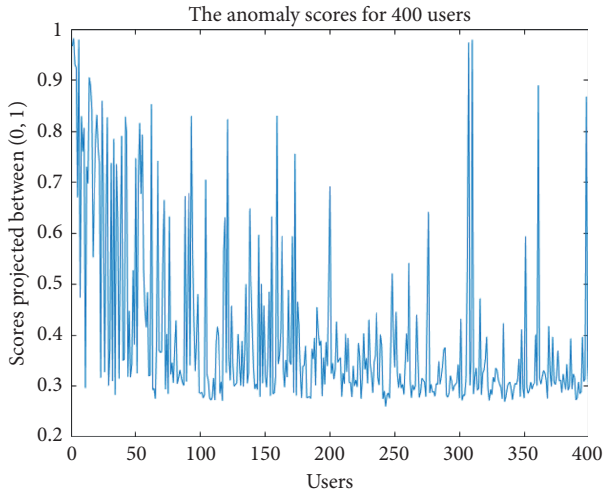

Figure 15: The anomaly scores for the users using KSE test.

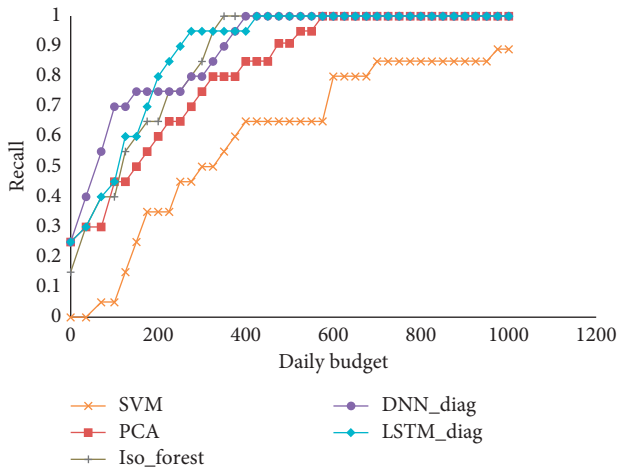

Figure 16: Recall curves computed on the testing set.

batch samples in DNN and 256–8,092 RNN samples. The learning rate was 0.00–0.01 and [3–40] back propagate time steps. If the model performance does not improve after 10 training steps, it stops training. We used Tanh as activation function and Adam as Stochastic gradient descent. Random search was also used to tune baseline hyperparameters. A range of [20–300] trees were used in the isolation forest, with [0–0.5] contamination and 1.0 max features. The number of principal components in PCA was [1–20]. Radial basis, linear, poly, and sigmoid functions were used as SVM kernels.

With this in mind, Figure 16 shows the recall curves for DNN and LSTM models, compared with SVM, isolation forest, and PCA. The DNN and RNN-LSTM algorithms were implemented using Python [12] and a set of libraries, including TensorFlow, NumPy, SciPy, Sklearn, and Matplotlib. Since the dataset was anonymized by its publisher, we assumed a certain level of data loss that might have had a negative influence on the agent's performance.

As seen in the figure when the daily budget is between 200 and 400, LSTM obtains 95% recall, but over 400, LSTM and DNN with a diagonal covariance matrix obtain 100% recall, followed by isolation forest in the third place.

As the RNNs implicitly identified the characteristics of the data and extracted the underlying features, this approach achieved higher accuracy compared to isolation forest. The key idea of isolation forest is that it randomly picks a feature and then randomly selects a split value between maximum and minimum values of the selected feature. Thereby, isolation forests do not learn what is normal; instead they attack anomalies directly. Since no distance metric is used in isolation forest, it tends to save time and computation. On the other hand, it is considered a simple model that suffers from misclassification with complex data and it does not consider time, so it cannot detect intrusion that spans over a great temporal range.

SVM, on the other hand, maps the data points in the training examples into feature space using a kernel function and then attempts to find a hyperplane with maximum margins that separates the mapped vectors [32]. It optimizes performance by trying to find the best machine for a set of examples by maximizing the correctness of machine regarding the training dataset and maximizing the classification capability of machine for the testing set. However, the standard SVM is sensitive to noise, leading to poor generalization ability [32], and consequently one class SVM is known to be sensitive to outliers and thus does not perform very well for outlier detection.

The PCA baseline projected the feature vector on the first k principal component, after which it was mapped back to the original feature space. The hyperparameter k was tuned on the development set, thereby meaning that the anomaly score was proportional to the error in the reconstruction. The PCA achieved good results because it tackles the anomaly detection problem as a dimensionality reduction rather than outlier identification [11].

6. Conclusion and Future Work

In this paper, we have proposed a model that aimed to protect the user's privacy in OSNs and prevent unauthorized insiders

from accessing and tampering with the user's data. Our proposed scheme leverages attribute-based access control to dynamically adjust permission assignments according to behavior-related information. Also, it integrates the access control model with machine learning supervised and unsupervised anomaly detection, thereby employing multiple stages to detect a user's anomalous behavior. This scheme contributed to the preservation of privacy by alerting users and detecting and preventing abnormal activity. It takes advantage of the existing attributes and generates only a few attributes. We experimented with real and simulated data adapted from social networks and measured the number of anomalous users communicating using these platforms. The false rate results of anomalous users were lower than reported, since the model used additional information derived from user and system attributes. This indicates the importance of assessing each user's condition individually.

In the future, we intend to generalize our model to apply multiple access control models and incorporate additional attributes for access authorization. We also intend to experiment using adversarial training process with generative adversarial network (GAN) classifiers to detect anomalies. The classifier generator maps training samples from a noise prior to data and implements a discriminator that classifies real and generated samples.

Data Availability

The model data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by a grant from the "Research Center of the Female Scientific and Medical Colleges," Deanship of Scientific Research, King Saud University.

References

[1] A. C. Hsu and I. Ray, "Specification and enforcement of location-aware attribute-based access control for online social networks," in *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control-ABAC*, vol. 16, New Orleans, LA, USA, March 2016.

[2] A.-C. Enache and V. Sgârciu, "Enhanced intrusion detection system based on bat algorithm-support vector machine," in *Proceedings of the 11th International Conference on Security and Cryptography*, Vienna, Austria, August 2014.

[3] M. Shehab, A. Squicciarini, G.-J. Ahn, and I. Kokkinou, "Access control for online social networks third party applications," *Computers & Security*, vol. 31, no. 8, pp. 897–911, 2012.

[4] Z. Pan, C.-N. Yang, V. S. Sheng, N. Xiong, and W. Meng, "Machine learning for wireless multimedia data security," *Security and Communication Networks*, vol. 1–2, 2019.

[5] H. Rong, T. Ma, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "Deep rolling: a novel emotion prediction model for a multi-participant communication context," *Information Sciences*, vol. 488, pp. 158–180, 2019.

[6] G. S. Okeeffe and K. Clarke-Pearson, "The impact of social media on children, adolescents, and families," *Pediatrics*, vol. 127, no. 4, pp. 800–804, 2011.

[7] W. Stallings, R. Perlman, S. Bellovin et al., "Authentication, Access Control, and Intrusion Detection," *Computer Science Handbook*, CRC Press, Boca Raton, FL, USA, 2nd edition, 2004.

[8] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network traffic anomaly detection techniques and systems," *Computer Communications And Networks Network Traffic Anomaly Detection And Prevention*, pp. 115–169, Springer, Berlin, Germany, 2017.

[9] D. M. D. Reis, P. Flach, S. Matwin, and G. Batista, "Fast unsupervised online drift detection using incremental Kolmogorov-Smirnov test," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, August 2016.

[10] T. Ko, J. H. Lee, H. Cho, S. Cho, W. Lee, and M. Lee, "Machine learning-based anomaly detection via integration of manufacturing, inspection and after-sales service data," *Industrial Management & Data Systems*, vol. 117, no. 5, pp. 927–945, Dec. 2017.

[11] C.-P. Tran and D.-K. Tran, "Anomaly detection in POSTFIX mail log using principal component analysis," in *2018 10th International Conference On Knowledge And Systems Engineering (KSE)*, Ho Chi Minh City, Vietnam, November 2018.

[12] Python® Machine Learning, *Getting Started with Scikit-Learn for Machine Learning*, pp. 93–117, Python® Machine Learning, Herndon, VA, USA, Aug 2019.

[13] B. Al-Otaibi, N. Al-Nabhan, and Y. Tian, "Privacy-preserving vehicular rogue node detection scheme for fog computing," *Sensors*, vol. 19, no. 4, p. 965, 2019.

[14] T. Ryutov, C. Neuman, D. Kim, and L. Zhou, "Integrated access control and intrusion detection for web servers," in *Proceedings of the 23rd International Conference on Distributed Computing Systems*, Taipei, Taiwan, April 2003.

[15] J. Wu and S. Shimamoto, "Integrated UCON-based access control and adaptive intrusion detection for wireless sensor networks," in *Proceedings of the 2010 IEEE Global Telecommunications Conference GLOBECOM*, Miami, FL, USA, December 2010.

[16] T. L. Norman, "Access control system servers and workstations," *Electronic Access Control*, Chapter 18, Butterworth-Heinemann, Oxford, UK, 2nd edition, 2017.

[17] S. Liu, J. Zhang, and Y. Xiang, "Statistical detection of online drifting twitter spam," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security-ASIA CCS*, vol. 16, Xi'an, China, May 2016.

[18] A. Odesile and G. Thamilarasu, "Distributed Intrusion detection using mobile agents in wireless body area networks," in *2017 Seventh International Conference On Emerging Security Technologies (EST)*, Canterbury, UK, September 2017.

[19] V. C. Hu, D. C. Ferraiolo, R. C. Kuhn et al., *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*, National Institute of Standards and technology, U.S Department of Commerce, Gaithersburg, MA, USA, 2014.

[20] C. Mazumdar, "Enterprise information system security," *Handbook Of Research On Social And Organizational Liabilities In Information Security*, pp. 118–132, IGI, New York, NY, USA, 2009.

[21] J. Esteveztapiador, P. Garciateodoro, and J. Diazverdejo, "Anomaly detection methods in wired networks: a survey and

taxonomy," *Computer Communications*, vol. 27, no. 16, pp. 1569–1584, 2004.

[22] A. Perrin, "Social media usage: 2005–2015," *Pew Research Center: Internet*, 2015.

[23] Y. Cheng, J. Park, and R. Sandhu, "Attribute-aware relationship-based access control for online social networks," *Lecture Notes In Computer Science Data And Applications Security And Privacy XXVIII*, pp. 292–306, Springer,, Berlin, Germany, 2014.

[24] T. Ma, H. Rong, Y. Hao, J. Cao, Y. Tian, and M. A. Al-Rodhaan, "A novel sentiment polarity detection framework for Chinese," in *Proceedings of the IEEE Transactions on Affective Computing*, p. 1, Piscataway, NJ, USA, July 2019.

[25] A. Tuor, K. Samuel, H. Brian, N. Nicole, and R. Sean, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," in *Proceedings of the In Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, March 2017.

[26] F. Meng, F. Lou, Y. Fu, and Z. Tian, *Deep Learning Based Attribute Classification Insider Threat Detection for Data Security*, in *Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, Guangzhou, China, June 2018.

[27] K. M. Carter and W. W. Streilein, "Probabilistic reasoning for streaming anomaly detection," in *Proceedings of the 2012 IEEE Statistical Signal Processing Workshop (SSP)*, Ann Arbor, MI, USA, August 2012.

[28] G. Gavai, K. Sricharan, D. Gunning, R. Rolleston, J. Hanley, and M. Singhal, "Detecting insider threat from enterprise social and online activity data," in *Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats-MIST*, vol. 15, Denver, CL, USA, July 2015.

[29] F. Yuan, Y. Cao, Y. Shang, Y. Liu, J. Tan, and B. Fang, "Insider threat detection with deep neural network," *Lecture Notes in Computer Science Computational Science–ICCS*, pp. 43–54, Article ID 10860, 2018.

[30] S. Watanabe, "Definition of bayesian statistics," *Mathematical Theory of Bayesian Statistics*, Chapter 1, CRC Press, Boca Raton, FL, USA, 1st edition, 2018.

[31] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "Towards detecting compromised accounts on social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 447–460, Jan. 2017.

[32] J. Glasser and B. Lindauer, "Bridging the gap: a pragmatic approach to generating insider threat data, 2013," in *Proceedings of the IEEE Security and Privacy Workshops*, San Francisco, CA, USA, May 2013.

[33] N. A. Heard, D. J. Weston, K. Platanioti, and D. J. Hand, "Bayesian anomaly detection methods for social networks," *The Annals of Applied Statistics*, vol. 4, no. 2, pp. 645–662, 2010.

[34] A. Reinhart, V. Ventura, and A. Athey, "Detecting changes in maps of gamma spectra with Kolmogorov–Smirnov tests," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 802, pp. 31–37, 2015.

[35] T. Yuan, M. M. Kaleemullah, M. A. Rodhaan, B. Song, A. Al-Dhelaan, and T. Ma, "A privacy preserving location service for cloud-of-things system," *Journal of Parallel and Distributed Computing*, vol. 123, pp. 215–222, 2019.

[36] R. Aljably, Y. Tian, M. Al-Rodhaan, and A. Al-Dhelaan, "Anomaly detection over differential preserved privacy in online social networks," *PloS one*, vol. 14, p. 4, 2019.