A Self-Adaptive Bell–LaPadula Model Based on Model Training With Historical Access Logs

Zhuo Tang, Xiaofei Ding, Ying Zhong, Li Yang, and Keqin Li, *Fellow, IEEE*

***Abstract*— In currently popular access control models, the security policies and regulations never change in the running system process once they are identified, which makes it possible for attackers to find the vulnerabilities in a system, resulting in the lack of ability to perceive the system security status and risks in a dynamic manner and exposing the system to such risks. By introducing the maximum entropy (MaxENT) models into the rule optimization for the Bell–LaPadula (BLP) model, this paper proposes an improved BLP model with the self-learning function: MaxENT-BLP. This model first formalizes the security properties, system states, transformational rules, and a constraint model based on the states transition of the MaxENT. After handling the historical system access logs as the original data sets, this model extracts the user requests, current states, and decisions to act as the feature vectors. Second, we use k-fold cross validation to divide all vectors into a training set and a testing set. In this paper, the model training process is based on the Broyden–Fletcher–Goldfarb–Shanno algorithm. And this model contains a strategy update algorithm to adjust the access control rules dynamically according to the access and decision records in a system. Third, we prove that MaxENT-BLP is secure through theoretical analysis. By estimating the precision, recall, and F1-score, the experiments show the availability and accuracy of this model. Finally, this paper provides the process of model training based on deep learning and discussions regarding adversarial samples from the malware classifiers. We demonstrate that MaxENT-BLP is an appropriate choice and has the ability to help running information systems to avoid more risks and losses.**

***Index Terms*— Adversarial sample, BLP, machine learning, mandatory access control, maximum entropy model, rule optimization.**

I. Introduction

THERE are many systems that are security-sensitive, such as military sectors, government, and banks, whose resources are usually partitioned according to the security levels while the entire system uses strict mandatory access control (MAC) to keep the data confidential and integral [1]. The Bell-LaPadula model (BLP) [2] is such a computer security model proposed by David Bell and Leonard La Padula to imitate military security policy, which has been widely applied and studied in high security systems, such as SELinux [3] and database security systems [4].

Due to the diversity of requests, especially in the cloud computing environment with mass traffic, the current security theorems and rules in BLP cannot adapt well to the variation of a distributed system [5]. Because most systems lack the abilities of environmental awareness and system self-adaptability, it usually causes the following situations:

First, the security of a traditional BLP model is based on the principle of stationarity, which requires the security levels of subjects and objects to remain unchanged once they are created. Practical applications show that this will result in strong constraints on subject behaviour, and it will also bring many security vulnerabilities and risks, which cannot to be monitored. For example, the representative strategy "*read down*" and "*write up*" in BLP can prevent information flow to insecure subjects or objects effectively. However, if we lack the capacity to identify threats in a dynamical procedure and adjust the access rules in time, some malwares can access the information that you do not want to publish by modifying their security levels at run time.

Second, even though there are no malwares and attackers to modify their security levels or objects' security levels forcibly, some inherent defects still exist in the traditional BLP model and its later improved versions, which can also be found in all other traditional access control models, such as Discretionary Access Control (DAC) and Role based Discretionary Access Control (RBAC). For example, the subject range for accessing the sensitive objects is usually defined beforehand, and this will destroy the system availability. In practice, the sensitive objects are neither always read by the subjects in the higher security levels nor always written by the subjects in the lower security levels at all times. Hence, the systems need to know about changes of the sensitiveness of the subjects and objects in time.

The above mentioned defects of the traditional BLP model are inherent because it is one of the typical traditional mandatory accesses control mechanisms, which all have the drawbacks of unchanged access rules once generated. That is, even though the rules are strictly enforced, the risks and threats still exist in these systems employing the BLP model, which are even unrecognized in the system. The traditional BLP

Manuscript received July 18, 2017; revised December 26, 2017; accepted February 14, 2018. Date of publication February 19, 2018; date of current version April 16, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61572176 and Grant L1624040, and in part by the National Key Research and Development Program of China under Grant 2017YFB0202201. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Eduard A. Jorswieck. *(Corresponding author: Zhuo Tang.)*

Z. Tang, X. Ding, and Y. Zhong are with the College of Information Science and Engineering, Hunan University, Changsha 410082, China, and also with the National Supercomputing Center in Changsha, Hunan University, Changsha 410082, China (e-mail: ztang@hnu.edu.cn; lkl@hnu.edu.cn).

L. Yang is with the College of Computer and Communication Engineering, Changsha University of Science and Technology, Hunan 410076, China.

K. Li is with the College of Information Science and Engineering, Hunan University, Changsha 410082, China, also with the National Supercomputing Center in Changsha, Hunan University, Changsha 410082, China, and also with the Department of Computer Science, State University of New York at New Paltz, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2018.2807793

models in the current applications lack the ability to perceive the system security status and risks in a dynamic manner. Fortunately, the system access logs can record the transactions and state transitions, which can be used to compensate the BLP model limitations regarding environmental perception and self-optimization based on data analysis and mining.

Hence, the operations need to be manually annotated as "secure" or "insecure" to judge the security situation, which can be the pre-processed training data after the historical access logs are annotated based on the empirical knowledge. It would be natural to employ the machine learning model to identify and predict the unrecognized threats that may be raised by authorized access requests from legitimate users.

As the core of artificial intelligence, machine learning [6] is an effective means of self-learning by computers, which are able to gain regulations by performing an automatic analysis on a known set of data and apply them to predict unknown data. For example, in the area of behaviour recognition, the basic methods are learning the rules or generating the decisions by model training [7]. For this target, there have already been a series of specific algorithms, such as Bayesian classification [8], the K-Nearest Neighbor algorithm (KNN) [9], Support Vector Machine (SVM) [10], the Hidden Markov Model (HMM) [11], Maximum Entropy Models (MaxENT) [12], and Conditional Random Fields (CRFs) [13].

In most running applications controlled by the BLP security polices, system decisions are not only based on the previous step but also depend on the current system states, security rules, and user requests; there actually are not Markov properties in the processing of state transitions. Compared to the HMM and CRFs models, MaxENT model is an appropriate selection for the model inference [14].

Except for the above shallow learning and clustering techniques, to identify unrecognized threats from the historic access logs, a deep learning model such as Deep Neural Network (DNN), is also appropriate for this purpose. The advantages are listed as follows to illustrate the reason why the MaxENT model is used in our works:

- Compared to other shallow learning models, the feature selections in MaxENT are more flexible, and the performance of the maximum entropy classifier is also usually superior to that of KNN, SVM, etc.
- Compared to the deep learning model, because the access log has a small number of features but a large number of records, the advantage of MaxENT can be appropriate for this type of data better than DNN. By adjusting the adaptation to the unknown parameters and the fitting levels for the known parameter, MaxENT can solve the problem of over-fitting, which often appears in deep learning models.
- Compared to other application scenarios with model training based on BFGS or deep learning, because any slight disturbance would make the current sample completely different from the original one, adversarial samples are hard to generate from the input feature data. The mode training of MaxENT based on BLP feature values is hard to attack with conventional adversarial samples.

There is usually much hidden knowledge in the system logs, which can be used to construct the training set [15]. Based on the MaxENT theory, this paper proposes an improved BLP model: MaxENT-BLP, which takes advantage of machine learning methods to train the history access logs and attempts to decrease the existing defects in the access rules of the traditional BLP model. By analysing and annotating the current system logs, the MaxENT-BLP establishes a self-optimizing model based on machine learning methods that can learn from empirical data, and thus the security policies and rules can be adjusted dynamically according to the historical and current secure states in the running systems. The main contributions of the paper are summarized as follows.

- We propose an improved mandatory access control model. Because the access control polices in MaxENT-BLP can be adjusted dynamically according to the current security states and events in a system, it improves the security and environmental perception for traditional security model.
- We prove the security of MaxENT-BLP in theory, and implement this model based on massive access logs in a practical system. Experimental results show that MaxMNT-BLP not only has good ability of threat recognition, but also has good precision and recall.

The rest of this paper is organized as follows. Section 2 surveys the related works. Section 3 proposes the model definitions. Section 4 illustrates the model training and solution through the historical access logs. Section 5 gives the security analysis and proof for this proposed model. Experiments and analysis that support our contributions are presented in Section 6. Section 7 proposes the solution of model training based on a deep learning method and discusses the defence capability for the malware classifier. Finally, Section 8 concludes this paper.

II. Literature Review

Because solutions to security problems usually rely on the empirical data, the use of machine learning to solve system threats and vulnerabilities is becoming an important means for the security analysis of distributed systems over these years. In recent years, the researches on adapting security models primarily involves intrusion detection and authorization & authentication systems.

A. Threat Discovery & Intrusion Detection

The vulnerability scanning and threat discovery often discover unknown risks that rely on existing knowledge. Hence, model training on the empirical data seems a natural and effective means to achieve this goal [18]. Yamaguchi *et al.* [16] used principal component analysis and a text mining technique to obtain defects in source code. Bozorgi *et al.* [17] proposed a method for training SVM and used it to predict system vulnerabilities. Puttini *et al.* [20] employed a Bayes classification model for real-time intrusion detection and modelled the user behaviours with a parameterized model. Tian *et al.* [21] designed a group of sequence

kernels to implement SVM-based anomaly detection of system calls.

Yeung and Ding [29] adopt an anomaly detection approach by detecting possible intrusions based on program or user profiles built from normal usage data using the hidden Markov model (HMM) and the principle of maximum likelihood. Because the feature selection is critical to knowledge-based authentication, Chen and Liginlal [30] adopt the principle of maximum entropy to implement the feature selection, which can be formulated as an optimization problem to maximize the Kullback-Leibler divergence between the guessing distribution and the true empirical distribution over the same space of a selected feature subset.

There are also various excellent research studies regarding how to detect anomalies and malicious files among the antimalware based on shallow learning and deep learning. Recent state of art in machine learning reported the issues that can occur if shallow learning classification or clustering techniques are deployed [26]–[28]. To take advantage of both shallow learning and deep learning models, these two types of training methods are all considered in our works to solve the aforementioned problems.

B. Adapting Access Control & Security Polices

In the study of adaptive access control, role mining is an effect way to match the appropriate privileges to users [22]. Recently, research studies have provided many approaches that automatically mine likely properties from a policy via the technique of association rule mining [23], [24]. Role mining is more suited to the role-based access control mechanism, in which the role takes the core position in building and maintaining the architecture of the RBAC system and migrating the non-RBAC system to the RBAC system [25]. In our model, because BLP is a typical MAC model, the principle of least privilege can be obeyed through the policy optimization based on MaxENT-BLP, even though there is no concept of role in the MAC model.

Moreover, there are many research studies on adapting security policies. Lo and Chen [31] proposed a method to evaluate the risk levels of information security under various security controls. In their works, the fuzzy linguistic quantifiers-guided maximum entropy order-weighted averaging operator is used to aggregate impact values assessed by experts, and applied to diminish the influence of extreme evaluations such as personal views and drastic perspectives. Mosenia *et al.* [19] described a novel continuous authentication system that authenticates users based on biomedical signal streams. By the way of machine learning, they can provide the high accuracy levels of biomedical signals.

From the above discussions, because the security and validity of created access rules need to be tested in the operational process of practical systems, it is feasible to adjust the model using existing experience data that comes from the running model processes. Because the representative research works on resolving the security state perception concern almost the entire information system, most of them lack the ability to mine and detect the vulnerabilities and risks for the specific secure policies.

III. MaxENT-BLP Model

In this section, we first provide the basic model description, which includes three security properties and element definitions, and then propose two specific improvements for the original BLP rules. Finally, by treating the request and state transition of BLP as the feature functions, the MaxENT-BLP model is constructed based on the computational process for the expectation of feature function of the MaxENT.

A. Basic Definitions

The BLP model defines a set of security properties to constrain the system states and the transition rules among different states. The main function of the rules in BLP is to ensure that the system state v is always a security state in the system, if and only if the state v satisfies the following three security properties [32].

1) Discretionary Property (ds-Property for Short): A state $v = (b, M, f)$ that satisfies the *ds-property* requires: $(s_i, o_j, x) \in b \Rightarrow x \in M_{i,j}$.

2) Simple Security Property (No Read Up, ss-Property for Short): A state $v = (b, M, f)$ that satisfies the *ss-property* requires: $(s, o, x) \in b$ *iff* (i) $x = e$ *or* $x = a$ *or* $x = c$ (ii) $(x = r$ *or* $x = w)$ *and* $L(s) \geqslant L(o)$.

*3) *-Property (No Write Down):* Suppose S' is a subset of S. A state $v = (b, M, f)$ satisfies **-property* to S' iff

$$s \in S' \Rightarrow \begin{cases} o \in b(S:a) \Rightarrow L(o) \geqslant L(s) \\ o \in b(S:w) \Rightarrow L(o) = L(s) \\ o \in b(S:r) \Rightarrow L(o) \leqslant L(s) \end{cases}$$

where, $b(S : x_1, x_2)$ represents the object set in b that subject S can access with x_1 or x_2 permission.

Thus, in this model, the security theorem defines the security of the current state in the system and decides whether the state transition rule is secure or not. The element definitions of this model are listed in Table I.

There are ten basic rules in the classic BLP model [2]. For instance, the first rule can be presented as Eq. (1), which formalizes subject s_i applying for permission r to access object o_j.

$$\rho_1(R_k, v) = \begin{cases} (?, v), & R_k \notin (\emptyset, g, s_i, o_j, r) \\ (yes, v^*), & R_k \in (\emptyset, g, s_i, o_j, r)\ \& \\ & r \in M_{i,j}, \quad L(s_i) \geq L(o_j)\ \& \\ & o \in b(s_i : w, a), \quad L(o_j) \leq L(o) \\ (no, v), & \text{otherwise} \end{cases} \tag{1}$$

where $v^* = \{b \cup \{s_i, o_j, r\}, M, f\}$. $\sigma_1 = \emptyset$, $\gamma = g$, $p = r$, $\sigma_2 \neq \emptyset$ is made to verify whether the domain of x satisfies the form of the tuple: $(\emptyset, g, s_i, o_j, r)$, and $r \in M_{i,j}$, $L(s_i) \geq L(o_j)$ is made to verify whether it obeys the *ds-property* and *ss-property*, which means that S_i has permission r to o_j in the access matrix M and its security level can dominate the security level of the object. $o \in b(s_i : w, a)$, $L(o_j) \leq L(o)$ are made to verify whether it obeys **-property*, which means that subjects cannot move information from an object with a higher classification to an object with a lower classification. Symbol "$\cup$" in $b \cup \{s_i, o_j, r\}$ represents the granting of permission r to subject s_i to access o_j.

TABLE I
ELEMENT DEFINITIONS OF MAXENT-BLP

Set	Elements	Meaning
S	$\{s_1, s_2, \ldots, s_n\}$	subjects: user, processes, and the programs in execution
O	$\{o_1, o_2, \ldots, o_n\}$	objects: data, files, or subjects
C	$\{c_1, c_2, \ldots, c_n\}$ $c_1 \leqslant c_2 \leqslant \cdots \leqslant c_n$	classification: clearance level of a subject, classification of an object
K	$\{k_1, k_2, \ldots, k_n\}$	need-to-know categories: project numbers, access privileges
A	$\{r, w, e, a, c\}$	access attributes: read, write, execute, append, or control
b	$b \in (S, O, A)$	the current access set
M	$m_{i,j} \in A$	access matrices: $m_{i,j}$ denotes s_i's access attributes to the object o_j
F	$f \in F$ $f = (f_1, f_2, f_3, f_4)$	security level function
$L(S)$	$f_1(s)$, $f_3(s)$	subject security level: f_1: classification, f_3: categories
$L(O)$	$f_2(o)$, $f_4(o)$	object security level: f_2: classification, f_4: categories
V	(b, M, f)	states
D	$\{yes, no, error, ?\}$	decisions. yes: accept, no: refuse error: system error, ?: request error
R	$\{r_1, r_2, \ldots, r_n\}$	request: input, commands, requests for accessing to the objects
req	(S^*, RA, S^*, O, X) $\{\sigma_1, \gamma, \sigma_2, o_j, x\}$ $req \in R$	$\sigma_1, \sigma_2 \in S^*$ is the licensor and licensee, $\gamma \in RA$ is request element, $RA = \{g, r, c, d\}$
ω^*	$\{\omega^{(1)}, \ldots, \omega^{(n)}\}$	the parameter values of the MaxENT Model need be optimized
$p_\omega*$	$\{p_\omega{}^{(1)}, \ldots, p_\omega{}^{(n)}\}$	the distribution evaluation of the MaxEnt Model need be optimized

B. Learning Ability of the Improved BLP Model

The state transitions in the BLP model are defined by current requests, rules, and the logical functions: ρ: $R \times V \rightarrow D \times V^*$. For a given state and a request, the BLP rules determine the response and the state transition. In this model, the state set is $V = \{v_1, v_2, v_3, \cdots, v_n\}$, the request set is $R = \{r_1, r_2, r_3, \cdots, r_n\}$, the decision set is $D = \{yes, no, error, ?\}$, and the set of results is represented as: $Y = \{secure, insecure\}$.

Moreover, in this expression $R \times V \rightarrow D \times V^*$, R represents the access request, V and V^* are the system status set before and after transitions respectively, and D represents the decision. In the MaxENT model, X represents the model inputs, Y represents the label results, and $f(x)$ represent the feature function. To build a bridge between the BLP and MaxENT model, we set R and V as the input X of the training data, and hence one of the basic element of the training data with the MaxENT model can be denoted as this tuple: $x = (r_i, v_i)$.

In this processing, the decision set D is regarded as the results of the data labels. As the output of the MaxENT model, the value scope of the set Y is equal to the decision set D in effect. To define an improved BLP model with some learning ability based on MaxENT, we first establish the feature vectors based on the original rules of the original BLP model. In this model, the training set is formalized as Eq. (2):

$$\mathrm{T} = \{[(r_1, v_1), y_1], [(r_2, v_2), y_2] \cdots [(r_n, v_n), y_n]\} \quad (2)$$

where y_i is the labelled result as the output. Then, we can obtain the empirical distribution as Eq. (3):

$$\begin{aligned} \widetilde{p}(x, y) &= \widetilde{p}[(r_i, v_i), y] = \frac{f(\mathrm{X} = x, Y = y)}{N} \\ &= \frac{f[(R = r_i, V = v_i), Y = y]}{N} \end{aligned} \quad (3)$$

and,

$$\widetilde{p}(X = x) = \frac{V(X=x)}{N} \quad (4)$$

where N is the total number of samples, which is the size of the annotated data. Accordingly, the basic form of the feature functions is designed as Eq. (5):

$$f_i[(r_i, v_i), y] = \begin{cases} 1, \ \rho_i(r_i, v_i) = yes, y = secure \\ 0, \text{ others} \end{cases} \quad (5)$$

where $i \in 0, 1, 2, \cdots, n$, $1 \leq i \leq 10$, and v_i consists of a three-tuple (b, M, f). When a system receives a request $r \in R$ under a state $v \in V$, it will output a decision such as "*yes*" according to a state transition rule ρ_i, which is based on the characteristic to keep the system always in secure state. Based on this, we can finally determine whether the feature values are set as 0 or 1. In this manner, the computation process for expectation of feature function for the empirical distribution $\widetilde{p}[(v_i, r_i), y]$ is as shown in Eq. (6):

$$\begin{aligned} E\widetilde{p}(f) &= \sum_{x,y} \widetilde{p}(x, y) f(x, y) \\ &= \sum_{(r_i, v_i), y} \widetilde{p}[(r_i, v_i), y] \, f_i[(r_i, v_i), y] \end{aligned} \quad (6)$$

And the expectation of feature function under both the model $p[y|(r_i, v_i)]$ and the empirical distribution $\widetilde{p}(r_i, v_i)$ can be calculated as Eq. (7):

$$Ep_i(f) = \sum_{(r_i, v_i), y} \widetilde{p}(r_i, v_i) p[y|(r_i, v_i)] f_i[(r_i, v_i), y] \quad (7)$$

From the above, the objective function and constrain conditions of the MaxENT-BLP model could be expressed as follows:

$$\begin{aligned} &\max_{p \in C} \ H(p) = -\sum_{(r_i, v_i), y} \widetilde{p}(r_i, v_i) p[y|(r_i, v_i)] \log p[y|(r_i, v_i)] \\ &\text{s.}t. \ \sum_y [y|(v_i, x_i)] = 1 \\ &\quad E\widetilde{p}(f_i) = Ep(f_i), 1 \leq i \leq 10 \end{aligned} \quad (8)$$

IV. MODEL TRAINING & STRATEGY OPTIMIZATION

This section firstly provides the overall framework of the data flow in MaxMNT-BLP model, and then presents the detailed course of computing the weights of the feature function. The output probabilities of these weights are also the probabilities of BLP decisions.

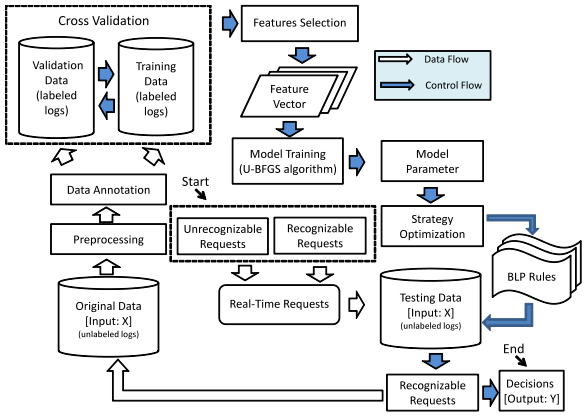


Fig. 1. The process diagram of MaxENT-BLP.

A. Process Description

The purpose of the model training is to obtain the knowledge to judge the secure state for the system decisions. Fig. 1 illustrates a simple but concrete example to explain the decision making process of the BLP model based on model training. After handling the historical system access logs as the original data sets, through pre-processing and data annotation, we first divide the data sets with the recognizable requests into the training data and the testing data, and then complete the features selection, finally extract the requests and current states to act as the feature vectors.

After the model training based on parameter estimation, we can complete the strategy optimization. In this way, the analogous requests that have a certain number of the same features with the recognizable request labelled "*insecure*" will be infused by the optimized strategies, even though that the initial BLP rules allow this request, but other requests with an automatic label "*secure*" will be validated with a decision "*yes*". The entire process is a closed loop with reinforcing feedback, this is because the requests with decisions based on model inference would be supplemented into the original logs. According to their actual secure states, these logs would be annotated with the label "*insecure*" or "*secure*".

MaxMNT-BLP represents the system action logs with the user requests as the input sequence set. For the requests at any time, if they do not exist in the original system access logs, this request will be manually annotated. In this process, the annotated results will be divided into three parts: the decision that is returned to the requesters, the total records supplemented into the original logs, and the extracted records supplemented into the current training set as the new training data. In this way, MaxMNT-BLP can be adjusted dynamically according to the current security states and events in a system, and it can improve the security and environmental perception for traditional security models.

B. Parameter Estimation

The MaxENT-BLP Model is essentially a typical optimization problem, which belongs to non-linear programming with linear constraints. Because our goal is to train the model parameters ω by the maximum likelihood estimation, this process is effectively the equivalent of searching the max values of the logarithm likelihood function. In this paper, an improved Iterative Scaling (IIS) algorithm [33] is used to solve the above problems. Suppose the current parameter vector of MaxENT-BLP is $\omega = (\omega_1, \omega_2, \cdots, , \omega_n)^T$, then a new parameter vector $\omega + \delta = (\omega_1 + \delta_1, \omega_2 + \delta_2, \cdots, \omega_n + \delta_n)^T$ is required, and it can make the value of the logarithmic model likelihood function increase. If there is such a method $\tau : \omega \to \omega + \delta$ to update the parameter vector, then it can be reused until we find the maximum number of the likelihood function.

For a given empirical distribution $\widetilde{p}((r_i, v_i), y)$, the change of the logarithm likelihood function with the parameter increment of δ can be calculated as Eq. (9):

$$\begin{aligned} L(\omega+\delta) - L(\omega) &= \sum_{(r_i,v_i),y} \widetilde{p}((r_i,v_i),y) \log p_{\omega+\delta}(y|(r_i,v_i)) \\ &\quad - \sum_{(r_i,v_i),y} \widetilde{p}((r_i,v_i),y) \log p_{\omega}(y|(r_i,v_i)) \\ &= \sum_{(r_i,v_i),y} \widetilde{p}((r_i,v_i),y) \sum_{i=1}^{n} \delta_i f_i((r_i,v_i),y) \\ &\quad - \sum_{x} \widetilde{p}(r_i,v_i) \log \frac{Z_{\omega+\delta}(r_i,v_i)}{Z_{\omega}(r_i,v_i)} \end{aligned} \tag{9}$$

And the lower bound of the increment of the log-likelihood function can be obtained based on this inequality: $-\log\alpha \geq 1-\alpha, \alpha > 0$:

$$\begin{aligned} L(\omega+\delta) - L(\omega) &\geq \sum_{(r_i,v_i),y} \widetilde{p}((r_i,v_i),y) \sum_{i=1}^{n} \delta_i f_i((r_i,v_i),y) + 1 \\ &\quad - \sum_{x} \widetilde{p}(r_i,v_i) \frac{Z_{\omega+\delta}(r_i,v_i)}{Z_{\omega}(r_i,v_i)} \\ &= \sum_{(r_i,v_i),y} \widetilde{p}((r_i,v_i),y) \sum_{i=1}^{n} \delta_i f_i((r_i,v_i),y) + 1 \\ &\quad - \sum_{x} \widetilde{p}(r_i,v_i) \sum_{y} p_{\omega}(y|(r_i,v_i)) \exp \\ &\quad \sum_{i=1}^{n} \delta_i f_i((r_i,v_i),y) \end{aligned} \tag{10}$$

Letting $A(\delta|\omega)$ represents the result of the lower bound obtained in Eq. (10), we have: $L(\omega+\delta) - L(\omega) \geq A(\delta|\omega)$.

If there exist an appropriate δ to raise the value of $A(\delta|\omega)$, that also makes the log-likelihood function increase. However, because δ in function $A(\delta|\omega)$ is a vector that consists of a plurality of variables, these variables are not easily optimized at the same time.

To lower the actual value of $A(\delta|\omega)$, the IIS algorithm is suitable for optimizing one of the variables δ_i while fixing the other variables δ_j, $i \neq j$. It also introduces the variable $f^{\#}((r_i,v_i),y) = \sum_i f_i((r_i,v_i),y)$. Here, f_i is a binary function, and $f^{\#}((r_i,v_i),y)$ represents the frequency of all the features appearing in $((r_i,v_i),y)$. Thus, $A(\delta|\omega)$ can be

rewritten as Eq. (11):

$$A(\delta|\omega) = \sum_{(r_i,v_i),y} \widetilde{p}((r_i,v_i),y)\sum_{i=1}^{n}\delta_i f_i(x,y) + 1 - \sum_{x}\widetilde{p}(r_i,v_i)\sum_{y} p_\omega(y|(r_i,v_i)) \times \exp\left(f^{\#}((r_i,v_i),y)\sum_{i=1}^{n}\frac{\delta_i f_i((r_i,v_i),y)}{f^{\#}((r_i,v_i),y)}\right) \quad (11)$$

Considering the convexity of the exponential function, for an arbitrary state or request, holds that: $\frac{f_i((r_i,v_i),y)}{f^{\#}((r_i,v_i),y)} \geq 0$ and $\sum_{i=1}^{n}\frac{f_i((r_i,v_i),y)}{f^{\#}((r_i,v_i),y)} = 1$. According to the Jensen inequality, we can obtain the following relationship as Eq. (12):

$$\exp\left(\sum_{i=1}^{n}\frac{f_i((r_i,v_i),y)}{f^{\#}((r_i,v_i),y)}\delta_i f^{\#}((r_i,v_i),y)\right) \leq \sum_{i=1}^{n}\frac{f_i((r_i,v_i),y)}{f^{\#}((r_i,v_i),y)}\exp\left(\delta_i f^{\#}((r_i,v_i),y)\right) \quad (12)$$

Based on Eq. (12), Eq. (11) can be rewritten as:

$$A(\delta|\omega) \geq \sum_{(r_i,v_i),y} \widetilde{p}((r_i,v_i),y)\sum_{i=1}^{n}\delta_i f_i(x,y) + 1 - \sum_{x}\widetilde{p}(r_i,v_i)\sum_{y} p_\omega(y|(r_i,v_i)) \times \sum_{i=1}^{n}\frac{f_i((r_i,v_i),y)}{f^{\#}((r_i,v_i),y)}\exp\left(\delta_i f^{\#}((r_i,v_i),y)\right) \quad (13)$$

In Eq. (13), the right part of this inequality is denoted as $B(\delta|\omega)$, and $\mathrm{L}(\omega+\delta) - L(\omega) \geq B(\delta|\omega)$. Therefore, $B(\delta|\omega)$ becomes a new lower bound of the amount of change of the logarithmic likelihood function. We can use Eq. (14) to work out the partial derivative of $B(\delta|\omega)$ to δ_i:

$$\frac{\partial B(\delta|\omega)}{\partial\delta_i} = \sum_{(r_i,v_i),y}\widetilde{p}((r_i,v_i),y) f_i((r_i,v_i),y) - \sum_{x}\widetilde{p}(r_i,v_i)\sum_{y} p_\omega(y|(r_i,v_i)) f_i((r_i,v_i),y) \times \exp\left(\delta_i f^{\#}((r_i,v_i),y)\right) \quad (14)$$

To detect the extreme points, the value of the expression of the partial derivatives is set to zero. Hence, we can obtain Eq. (15):

$$\sum_{(r_i,v_i),y}\widetilde{p}((r_i,v_i))\, p_\omega(y|(r_i,v_i))\, f_i((r_i,v_i),y) \times \exp\left(\delta_i f^{\#}((r_i,v_i),y)\right) = E_{\widetilde{p}}(f_i) \quad (15)$$

Hence, the traditional MaxEnt model can be formalized as follows:

$$p_\omega(y|x) = \frac{1}{Z_\omega(x)}\exp\left(\sum_{i=1}^{n}\omega_i f_i(x,y)\right) \quad (16)$$

where

$$Z_\omega(x) = \sum_{y}\exp\left(\sum_{i=1}^{n}\omega_i f_i(x,y)\right) \quad (17)$$

In Eq. (17), $Z_\omega(x)$ is called the normalization factor, and $f_i(x,y)$ is the feature function. ω_i represents the weight of the feature, which is the parameter vector of the MaxENT model. Thus, the value of δ_i can be worked out by solving Eq. (15), and then further to calculate the weights of ω, which can be plugged into Eq. (16) and Eq. (17) to obtain the optimal estimate of MaxENT-BLP expediently.

C. Strategy Optimization Algorithms

In this section, the Quasi-Newton method [34] can be applied to solve the learning processing of MaxENT-BLP. $p_\omega(y|x) = p_\omega[y|(r_i,v_i)]$ can be calculated as Eq. (18):

$$p_\omega[y|(r_i,v_i)] = \frac{\exp\left(\sum_{i=1}^{n}\omega_i f_i[(r_i,v_i),y]\right)}{\sum_{y}\left(\sum_{i=1}^{n}\omega_i f_i[(r_i,v_i),y]\right)} \quad (18)$$

The objective function can be established as Eq. (19):

$$\min_{\omega\in R^n} f(\omega) = \sum_{x}\widetilde{p}(r_i,v_i)\log\sum_{y}\exp\left(\sum_{i=1}^{n}\omega_i f_i[(r_i,v_i),y]\right) - \sum_{x,y}\widetilde{p}[(r_i,v_i),y]\sum_{i=1}^{n}\omega_i f_i[(r_i,v_i),y] \quad (19)$$

Then, the equation of the gradient can be calculated as:

$$\mathrm{g}(\omega) = \left(\frac{\partial f(\omega)}{\partial\omega_1}, \frac{\partial f(\omega)}{\partial\omega_2}, \cdots, \frac{\partial f(\omega)}{\partial\omega_n}\right)^T \quad (20)$$

where:

$$\frac{\partial f(\omega)}{\partial\omega_i} = \sum_{x,y}\widetilde{p}(r_i,v_i)\, p_\omega[y|(r_i,v_i)] f_i[(r_i,v_i),y] - E_{\widetilde{p}}(f_i), \quad i = 1,2,\cdots,n \quad (21)$$

Because the MaxENT is a typical probabilistic graph model, we can solve this model through an improved Broyden Fletcher Goldfarb Shanno (BFGS) algorithm [34], which is an iterative method for solving unconstrained nonlinear optimization problems. For a real-valued function $f: R_n \to R$, $g(x)$ and $G(x)$ denote the gradient and the Hessian matrix of the function f at x respectively. For simplicity, they are often denoted by g_k and G_k. In this model, function $f: R_n \to R$ is continuously differentiable, and for a vector $x \in R_n$, $\|x\|$ denotes its Euclidean norm.

For the unconstrained optimization problem: $minf(x)$, $x \in R_n$, a sequence x_k can be generated by the iterative scheme by BFGS: $x^{(k+1)} = x^{(k)} + \lambda_k \times p_k$, $k = 0,1,2,\cdots,M$, where p_k is the direction obtained by solving the linear equation: $B_k p_k + g_k = 0$. The matrix B_k can be updated as Eq. (22) during the following iterations:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T\delta_k} - \frac{B_k\delta_k\delta_k^T B_k}{\delta_k^T B_k\delta_k} \quad (22)$$

where, $y_k = g_{k+1} - g_k$, $\delta_k = x^{(k+1)} - x^{(k)}$.

In Eq. (22), B_{k+1} inherits the positive definiteness of B_k as long as ${y_k}^T \times s_k > 0$, which is guaranteed to hold if the step size λ_k is determined by an equation of line search as Eq. (24):

$$f(x^{(k)} + \lambda_k p_k) = \min_{\lambda \geq 0} f(x^{(k)} + \lambda p_k) \tag{23}$$

Algorithm 1 illustrates the entire process to solve MaxENT-BLP by the BFGS method. The computational cost of algorithm 1 depends on the dimension N of the symmetric matrix B, thus the time complexity of algorithm 1 is $O(N \times N)$.

Algorithm 1 BFGS Algorithm of Maximum Entropy Model Learning

Input:
Feature functions $f_1, f_2, ..., f_n$;
Empirical distribution $\widetilde{p}[(r_i, v_i), y]$;
Objective function $f(\omega)$;
Gradient $g(\omega) = \bigtriangledown f(\omega)$;
Accuracy ε.
Output:
Optimal parameter values ω^* ;
Optimal model $p_{\omega^*}[y|(r_i, v_i)]$.
//begin procedure£º
Initial point $\omega^{(0)}$, set B_0 as positive definite symmetric matrix, set $k = 0$.
while TRUE **do**
Calculate the p_k according to $B_k p_k = -g_k$
Calculate the λ_k according to Eq. (22)
Calculate the $g_k = g(\omega^{(k)})$ according to Eq. (19)
if $||g_k|| < \varepsilon$ **then**
Get the iteration final $\omega^* = \omega^{(k)}$
Break;
end if
$k \leftarrow k + 1$
Update the value of ω_i according to : $\omega^{(k+1)} \leftarrow \omega^{(k)} + \lambda_k p_k$
Update the value of B_k according to Eq. (21)
Update the value of y_k according to : $y_k = g_{k+1} - g_k$
end while
return $p_{\omega^*}[y|(r_i, v_i)]$.

Based on the results of the model training, as a strategy update processing, Algorithm 2 is designed to adjust the access control rules dynamically according to the current security states and events in a system. In this process, the records that cannot be identified automatically according to historical experience will be exported to be marked by manual tagging. After that, we can put these all marked records into the original record set iteratively. Finally, the necessary model parameters can be obtained via a training process by Algorithm 1. Actually, Algorithm 2 just re-calculates the new weights of the feature function with the original data vectors. The output probabilities of these weights are also the probabilities of BLP decisions. In this way, the MaxENT Model can adjust the access control rules adjusted dynamically according to the current security states and events in the running process of the system.

TABLE II
DEFINITION OF REQUEST FEATURE VECTOR

Element	Value	Meaning
subject σ_1	0	assigned a null value
	S_i	a specific subject s
request element γ	$g/r/c/d$	take a random value in RA
subject σ_2	0	assigned a null value
	S_i	a specific subject s
object O_j	0	assigned a null value
	O_j	a specific subject s
x	$r/w/e/a/c$	take a random value in access property set
	0	assigned a null value
	F	security level of subject and object

Algorithm 2 UBFGS: Strategy Update Algorithm

Input:
Parameter K;
Origin Data vector set D_0;
Unable to distinguish Data vector set D_1;
Data vector set D_2, D_3, D_4.
Output:
New optimal parameter values $\omega^{\#}$;
New optimal model $p_{\omega^{\#}}[y|(r_i, v_i)]$.
//begin procedure£º
//D_0 represent the sum record before current moment, D_1 represent
//the sum record whose request Unable to identify in current a period of
// time
Merger D_0 and D_1, denote as a new set D_2;
// K-fold cross-validation can be used to divide training set and test set
Using K-fold cross-validation divide D_2 into two part, training set D_3 and test set D_4;
//D_3 is the new training set after being mergered
Using D_3 and algorithm 1 to get the value $p_{\omega^{\#}}[y|(r_i, v_i)]$;
return $p_{\omega^{\#}}[y|(r_i, v_i)]$.

D. Feature Selection

Because the historical system access logs could be served as the original data set in MaxENT-BLP, requests and current state should be extracted to act as the feature vectors. The value of the feature vector is composed of a quintuple: $(\sigma_1, \gamma, \sigma_2, o_j, x)$, and the meanings of each value in the feature vector are listed in Table II.

In this model, the current security state of a system can be formalized as a triple (b, M, f). As in the following log shown in Table III, for a certain system log in our experiments, the pre-treated data can be represented as five components, and more detailed analysis for the experimental data is presented

TABLE III
A PRE-PROCESSING LOG WITH DECISION: *yes*

B	$s3o2e$ 0 0 0 0
M	$s1o1r$ $s2o5wea$ $s1o5wea$ $s2o2werca$ $s3o10$ $s1o2wea$ $s2o1rc$ $s3o5wea$ $s3o2e$
F	$o110$ $s12310$ $o23310$ $o54310$ $s233210$ $s343210$
R	0 g $s1$ $o5$ w
Decision	*yes*

TABLE IV
A PRE-PROCESSING LOG WITH DECISION: *no*

B	$s2o2wr$ $s2o1r$ $s3o2e$ $s2o5r$ 0 0
M	$s2o3c$ $s2o2wrca$ $s3o10$ $s2o1rc$ $s3o3c$ $s3o2we$ $s3o5wrca$ $s2o5wa$
F	$o110$ $o23310$ $s233210$ $s343210$ $o323$ $o543210$
R	0 g $s2$ $o5$ r
Decision	*no*

in Section 5.1. The first part is the current access-set b, string "*s3o2e*" indicates that subject s_3 has permission e on object o_2 at present.

The second part is the current system privilege matrix M. Because we selected 3 subjects and 3 objects from the access logs, M contains 9 columns in the inner data structure. The third part denotes the set of current security level function F. There are 6 columns in the same manner, including 3 subjects and 3 objects. The value "*s12310*" in F shows the security classification is 2 and the department set is $\{3, 1, 0\}$. The fourth part describes the current system request R. It contains 5 columns, which means subject s_1 asking for permission w to access object o_5.

The last part is the decision set that needs to be labelled after training in this model, which is to label the system decisions for a special input, including "*yes*, *no*, *?*" and "*error*". In rule 1, the training data cannot be labelled "*yes*" until the following conditions are satisfied: the request should be legal (the fourth part), the security levels of subjects must dominate those of objects (the third part), the requested permissions must belong to the privilege matrix (the second part), and **-property* must be satified (the first part and third part). Therefore, there are a total 25 features in the data set for the five columns in these five parts.

Given this, the paper adds a list of features to the data set in MaxENT. The following example in Table IV represents subject s_2 asking to read object o_5, but the system rejects this request because the security level of o_5 dominates that of s_2.

The feature vectors are extracted from the original data and denoted as (R, V, D, Y), where R represents the requests of access, V represents the states of the MaxENT-BLP system, D represents the decisions of the request, and Y represents the set of annotations.

In this model, the decision set D can be divided into the secure and insecure parts: D_s and D_u respectively. Therefore, the data format after processing is $(R_i, V_j, D_k, Y), (0 \leq i \leq m_0, 0 \leq j \leq m_1, k = 0, 1)$. For example, (r_2, v_1, d_s, y_1), where $d_s \in D_s$ and $y_1 \in Y$, denotes that the execution operation from subject S_0 to object O_0 in the case of system state v_1 is secure.

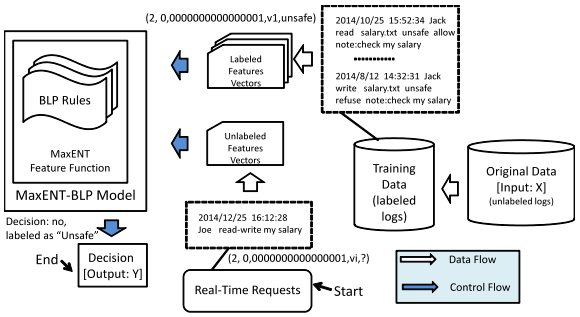


Fig. 2. An instance of MaxENT-BLP decision process.

After processing the original data, K-fold cross validation is used to the model training for MaxENT-BLP. In this method, because R and V in the test set are regarded as the input, we can obtain a decision set D' as the output. If D' is consistent with D in the test set, the system is effective.

E. Instance Analysis

In our application environment, there is a distributed office automation (OA) system in a large manufacturing corporation.[1] The access control model and security polices in this system are basically based on the BLP model. Actually, in most larger corporations, the BLP model usually can bring administrative convenience for the massive users and resources.

This section describes a typical process regarding how some operations from validated users are identified as a threat to the system over time. In this instance, we describe the overall process of how an information system generates the new rules and modifies the existing rules based on the processed access logs dataset. The system rejects the request from Jack to access the salary file of the company, because based on the historical access logs, this type of operations should be recognized as a threat according to some of its features. To illustrate this process in detail, the basic data flow of this instance is shown in Fig. 2.

In this system, the items in the original access logs were recorded as follows:

- *2014/12/23 11:32:34 Amy read only salary.txt secure allow note:routine inspection*
- *2014/12/25 16:12:28 Joe read-write salary.txt null null note:check my salary*
- *2014/10/25 15:52:34 Jack read salary.txt insecure allow note:check my salary*
- *2014/8/12 14:32:31 Jack write salary.txt insecure refuse note:check my salary*

Table V and Table VI give the partial user information and authorization configuration in this company. As shown in these records, Amy is a general manager, and she has all permission for the files that belong to this database company. Bob is

[1] https://www.szzt.com.cn/en/

TABLE V
BASIC INFORMATION OF USERS

Cid	Cname	Cbirthday	Cduty	Cgroup
20160001	Amy	1971/03/20	General manager	1
20160002	Bob	1981/10/10	Department manager	2
20160003	Jack	1986/06/26	Software Engineer	3
20160004	Joe	1989/01/15	Senior Clerk	3

TABLE VI
THE AUTHORIZATION CONFIGURATION FOR THE OBJECT SALARY.txt

Cgroup	read only	read-write	execute	append	control
1	1	1	1	1	1
2	1	0	0	0	0
3	1	0	0	0	0

TABLE VII
ENCODING OF OPERATIONS IN ACCESS LOGS

BI	FA	R_1	W_1	E_1	A_1	C_1	R_2	W_2	E_2	A_2	C_2
1	0	0	1	0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	1	0	0	0	0

a department manager, so he does not have the permission to access salary.txt. In addition, Joe and Lily are ordinary employees, and thus they just have read only permission to the responding file.

The traditional BLP model lacks the ability to adjust the rules according to the system running status. The allowed requests at the beginning would never be rejected if the security levels stay the same, despite the same type of accesses being annotated as a threat many times.

In this instance, the operations from Jack have been repeatedly labelled as threats even though he has legal permission to read his own salary records. The traditional BLP model is unable to recognize the same type of risks. That is, despite of the high similarity of actions between Jack and Joe for many of the same parts of features, the operations from Joe are still allowed for his legitimate identity.

Fortunately, the advanced access mechanisms based on MaxENT-BLP can recognize previously unknown threats. Table VII quantize the original records, where BI denotes whether the subject is in the basic information table, and FA denotes whether the subject is authorized with the relevant permissions. In these fields, 0 represents "no", and 1 represents "yes".

Based on the above privileges numeralization, we use $BF = \{BI, FA\}$ to denote the basic information set, $RP = \{R_1, W_1, E_1, A_1, C_1\}$ to denote the requested permission for the objects, and $AP = \{R_2, W_2, E_2, A_2, C_2\}$ to denote the authorized permission by the authorization system. The characters in the two sets represent the following corresponding permissions: "*R: read; w: write; E: execute; A: append; C: control*". In this manner, the records in the training set are all encoded and labelled as this formalization, which are the inputs of the training process in the MaxENT-BLP model.

The first row in Table VII is from a specific request: "$2015/12/25\ \ 15 : 52 : 34\ \ Jack\ \ write\ \ salary.txt$", which represents a record of requesting the write permission to the object "0000 0000 0000 0001". Because Jack is in the basic information, $BI = 1$. For its group $Cgroup = 3$, we have: $FA = 0$, $BF = BI$, $FA = 10 = 2$.

For the feature vector (R, V, D, Y), $R = (BF, RP\&AP, objects)$, and D denotes the artificial label, with the values belonging to this set: $\{secure, insecure\}$. In this case, because $BF = 10$, $RP = 00010$, $AP = 10000$, and $RP\&AP = 0$, we have $R = BF, RP\&AP, objects = 2, 0, 0000000000000001$. Because this record is manually labelled as "$insecure$", the feature vector based on the previous template can be written as: $(2, 0, 0000000000000001, v_1, insecure)$.

The second row in Table VII gives the encoding for another unrecognized request: "$2016/10/25\ \ 05 : 52 : 34\ \ Joe\ \ add\ \ salary.txt$". The original record is shown as an unknown threat with out a manual label. Similar to the first record, we also have: $BF = BI$, $FA = 10 = 2$, $RP = 00100$ for requesting the write permission, and hence its feature vector can be represented as: $(2, 0, 0000000000000001, v_i, ?, ?)$.

Finally, by through putting this vector to the MaxENT-BLP model, the full feature vector $(2, 0, 0000000000000001, v_i, insecure, no)$ can be obtained.

V. SECURITY ANALYSIS & DISCUSSION

MaxENT-BLP improves the system transition rules to enhance its availability. These improvements are based on the reservation or reinforcement of the original security of the BLP model. Because these security rules still obey the information flow direction of "*No read up, No write down*" in the traditional BLP model, this improvement just appends stricter restrictions, so the improved security model also does not violate the basic characteristics, *ss-property*, and **-property*.

In essence, the *ss-property* limits the execution of "*read only*" when the subject security levels are higher than that of objects. In MaxENT-BLP, the random accesses from subject s to an object o will not cause information to flow from a high security level to low security level. In this way, the *ss-property* will be kept in the entire transition. **-property* limits the rules to allow the *add* operation when the security levels of objects are higher those of subjects and allows the *read/write* operation when the security levels of objects are equal to those of subjects. In this improved model, the operations *restricted read/write* are just for specific objects, and should satisfy the essential condition: "*security level of object dominates that of subject*", so it still satisfies **-property* in the BLP model. Furthermore, because users with low security levels cannot modify the specific data in a high security level, this not only improves the integrity of the BLP model but also enhances the usability of the original BLP model.

In the MaxENT-BLP model, the state transition rules are all originated from the BLP model. Thus, it is necessary to provide the security proof for this new rule. Based on the following theorems, we can prove that the security of the MaxENT-BLP model is equivalent to that of the BLP model.

Definition: A state v is a secure state iff v satisfies *ss-property*, **-property*, and *ds-property*.

TABLE VIII
NUMBER OF DECISION IN DATASET

DataSet	Secure	Insecure
17,640,731	17,628,970	11,761

TABLE IX
NUMBER OF DECISION IN TRAINING SET AND TESTING SET (NUMBER/PORTION)

Cross-validation	Tag	Training set	Testing set
2-fold	Secure	11,752,648	5,876,323
	Insecure	7,840	3,920
	(Sum)	11,760,648/66%	5,880,243/33%
4-fold	Secure	13,221,728	4,407,243
	Insecure	8,820	2,940
	(Sum)	13,230,548/75%	4,410,183/25%
8-fold	Secure	15,425,348	2,203,621
	Insecure	10,291	1,471
	(Sum)	15,435,639/82.5%	2,205,092/17.5%

Theorem: Rules after training still satisfy MaxENT-BLP security constraints, and the training process has no effect on the security of the model.

Proof: The training dataset is derived from preprocessing of system access logs, but the preprocessing procedure does not change the security properties of each element in system logs. Therefore, the satisfaction of the basic features of MaxENT-BLP in these log data is reserved.

For some special requests, such as skip-level accesses, by reducing the read-write scope of some objects, stricter limitations are imposed to avoid breaking the basic features of the MaxENT-BLP model, including *ss-property* and **-property*. In this way, some skip-level operations are forbidden in this model, and just the objects belonging to the same security levels or adjacent security levels can be accessed. In conclusion, rules after training still satisfy the security constraints of the traditional BLP model. This completes the proof of the theorem.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Conditions and Method

This section illustrates a group of experiments to verify the effectiveness of MaxENT-BLP. The related programs in these experiments are based on a BFGS tool package [35]. MaxENT-BLP can be widely used in the access control modules with the independent logs sub-systems. In this paper, the original data are from a real distributed office automation system based on a MySQL cluster, and the processed dataset consists of 17,640,731 record vectors. For this dataset, there are two types of annotations for this feature set: $Secure$ and $Insecure$. Table VIII shows the distribution of the dataset.

Table IX illustrates the decisions distribution of the training set and testing set. The numbers in this table count the same decisions in the data set.

To assess the performance accurately, we use three cross-validation techniques to train and evaluate the dataset. In 2-fold cross-validation, we divide 17,640,731 request records into two sets: The first set contains 11,760,648 record vectors, and the second set contains 5,880,243 record vectors. Table IX lists the number of decisions in the training set and testing set. The tags "Secure" and "Insecure" denote that the related operations are secure or insecure for the system respectively, which are also the evidence for the decisions regarding the related operations made by the system. In this case, three-fourths of the vectors (2/3 of each decision) are used for training and one-fourths for testing (1/3 of each decision). The same analysis method can be used to explain the 8-fold cross-validation in these experiments.

TABLE X
THE RESULTS OF THE 8-FOLD VALIDATION TECHNIQUES

Dataset	Decision	Precision	Recall	F1-Measure
DS1	Secure	99.72%	92.54%	95.99%
	Insecure	93.64%	100%	96.72%
DS2	Secure	98.64%	95.66%	97.13%
	Insecure	94.52%	98.69%	96.56%
DS3	Secure	98.89%	93.27%	95.99%
	Insecure	96.59%	99.88%	98.20%
DS4	Secure	99.79%	92.32%	95.91%
	Insecure	99.85%	94.32%	97.01%
DS5	Secure	100%	98.33%	99.16%
	Insecure	96.76%	99.98%	98.34%
DS6	Secure	99.99%	94.55%	97.19%
	Insecure	96.53%	100%	98.23%
DS7	Secure	100%	98.56%	99.27%
	Insecure	95.68%	99.68%	97.64%
DS8	Secure	100%	98.34%	99.16%
	Insecure	95.63%	98.53%	97.06%

TABLE XI
COMPARISON OF AVERAGE P, R, F1 FOR TWO CROSS VALIDATION TECHNIQUES

cross validation	Average (%)		
	Precision	Recall	F1-Measure
2-fold cross validation	93.44%	93.44%	93.4%
4-fold cross validation	95.54%	95.54%	95.54%
8-fold cross validation	97.40%	97.40%	97.40%

B. Analysis of Experimental Results

Table X lists the results of 8-fold cross-validation techniques. In Table XI, the resulting correct differentiation rates represent the average of F1-Measure, and it reflects that MaxENT-BLP could make a relatively appropriate system decisions when it receives the users access requests from another perspective. In these experiments, the accuracies are up to 97.40% with 8-fold cross-validation.

The comparison experiments between traditional BLP and MaxENT-BLP validate the effectiveness of the proposed model. Because the data in Fig. 3(a) denotes the accumulated number of all previous days, the number of illegal accesses of BLP presents linear growth. However, with the model training and self-learning capacity, some previously unknown illegal accesses will be recognized and then be intercepted along with the MaxENT-BLP system module running. Therefore,

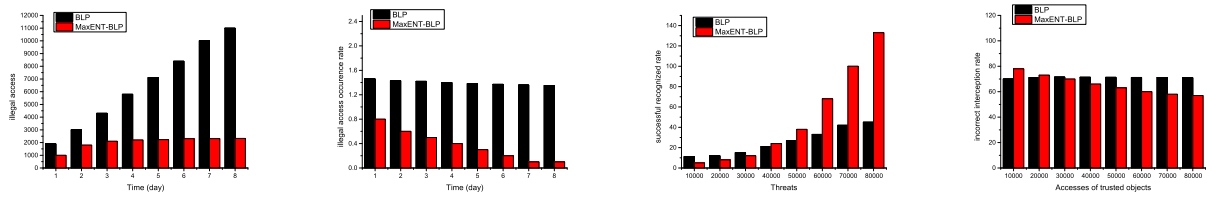


Fig. 3. The security evaluation of MaxENT-BLP. (a) Illegal accesses. (b) Illegal accesses occurrence rate. (c) Successful recognized rate. (d) Incorrect interception rate of secure access.

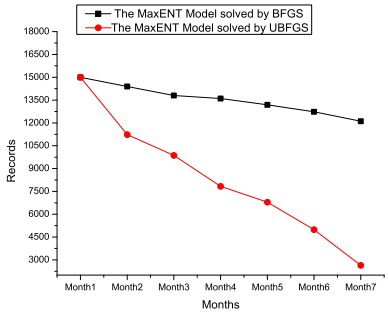


Fig. 4. The number of unrecognized requests.

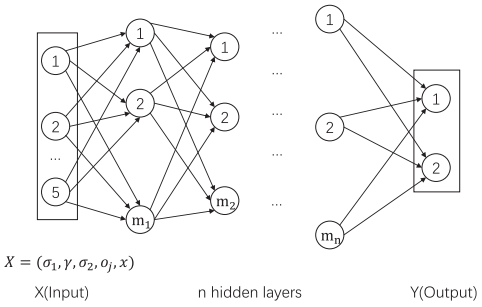


Fig. 5. The deep learning model of our task.

the growth rates of accumulated number of illegal accesses are lower compared to the traditional BLP model.

Fig. 3(b) reflects the comparison of protection effect to the system between the traditional BLP model and the MaxENT-BLP model. The y-axis denotes the occurrence rate of the illegal accesses for each day during the experiment, which can be calculated by dividing the number of illegal accesses by the total access number of each day.

In these experiments, because the cause of illegal accesses is that the threats usually cannot be recognized, the relationship between the threats and the illegal accesses are illustrated clearly in Fig. 3(c). With the minimal data, for the imprecise record samples, the recognition rate of the MaxENT-BLP model is lower than that of the traditional BLP model instead. However, with the growth of the training data, due to its positive feedback mechanism, MaxENT-BLP becames more and more precise at recognizing the threats as the system runs.

Fig. 3(d) shows the incorrect interception percentage of secure access. The figure indicates that as time and access records increase, the percentage of incorrect interception of the accesses to the secure objects will remain steady in BLP. However, on the contrary, MaxENT-BLP can make the incorrect interception of secure access lower and lower to some extent. In the initial stages, for a small number of record samples, the incorrect interception percentage of MaxENT-BLP is a lightly larger than that of BLP. However, through the rules optimization after the model training and rules learning process, the same types of accesses that were intercepted by mistake will be allowed into the system. As a result, the incorrect interception percentage of secure access in MaxENT-BLP will be decreased with the increment of the system traffic.

Fig. 4 illustrates that, if we just adopt the BFGS algorithm to implement MaxENT, for these non-existent requests in the original system logs, because the requests are unable to be identified, these requests that cannot be handled will grow in number. The experimental results show that with the effect of Algorithm 2, the related polices can be adjusted dynamically according to the system states, which can decrease the unrecognized request effectively. Meanwhile, the comparison between the two curves in Fig. 4 also illustrates that the effect of model training under Algorithm 2 is better than under the original BFGS algorithm because UBFGS can re-recognize the training data by supplementing the new annotated data periodically.

It is well known that for all supervised machine learning models, the accuracies of the model inferences are almost up to the quality of the data in training sets. With more training samples, better predicting results for the testing set can be obtained. Because the model training process in MAXENT-BLP is also typically supervised learning, we can conclude that the effect of the proposed model is up to the quantities of the information in the historical logs.

VII. Discussion

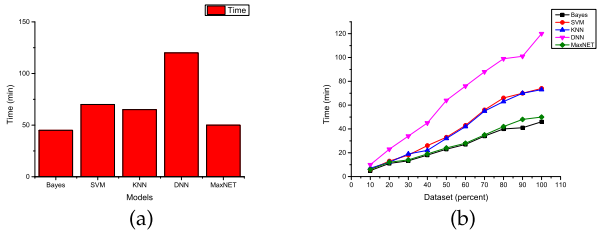
A. Model Training Based on Deep Learning

Without a doubt, there are several traditional classification approaches and deep machine learning methods that can be used to increase the environment aware abilities of the BLP model based on the data training from the historical logs. In particular, deep learning models are more robust than many traditional machine learning models when they are faced with malicious attacks. Therefore, we also apply the deep learning model to process our task and it can make our model more robust in protecting itself from the attack.

The nature of the data annotation and training can be treated as a problem of binary classification: the historical logs should be divided into a secure and insecure group automatically. To address this issue, we adopt a deep neural networks (DNN) as a classifier. The model architecture, shown in Fig. 5, is a

TABLE XII
RESULTS OF OUR MAXENT MODEL AGAINST OTHER METHODS

Model	Precision	Recall	F1-Measure
Bayes	87.51%	90.31%	90.91%
KNN	89.51%	90.32%	89.91%
SVM	91.32%	90.42%	94.20%
DNN	92.32%	95.42%	95.24%
MaxNET	**97.40%**	**97.40%**	**97.40%**



(a) (b)

Fig. 6. The training time of five models. (a) The training time. (b) The training time under different proportions datasets.

slight variant of the DNN architecture for our task. The feature vectors selected in Section 4.4 are used as input for our deep learning model. Let $X = (\sigma_1, \gamma, \sigma_2, o_j, x)$ be a 5-dimensional feature vector as our model's input. The value of each dimension is different. As shown in Fig. 5, each neuron in the input layer is used to accept one dimension's value of the feature vector.

Our deep learning model consists of n hidden layers and each hidden layer contains m neurons. These neurons, which apply activation functions, are linked to each other to learn models. A neural network model F can be defined as follows:

$$\begin{aligned} X &= (\sigma_1, \gamma, \sigma_2, o_j, x) \\ \mathbf{F} : X &\mapsto f_n(\ldots f_2(f_1(X, \theta_1), \theta_2) \ldots, \theta_n) \end{aligned} \tag{24}$$

where each vector θ_i parameterizes layer i of the network F and includes weights for the links connecting layer i to layer $i-1$. These parameters $\theta = \{\theta_i\}$ are learned during model training. We use stochastic gradient descent to train our model.

DNN is an appreciate method that can be used to implement the secure/insecure classification. To illustrate the effectiveness of the method objectively, we have tested the models of DNN and MaxENT based on the same training data. In addition, to compare with some traditional machine learning models, we also apply shallow methods including Bayes, KNN, and SVM.

KNN: This algorithm has a major shortcoming for classification. It is sensitive to very unbalanced datasets. When most entities belong to one or a few classes, the infrequent classes will dominate in most neighborhoods.

SVM: Support Vector Machine is one of the most efficient machine learning algorithms for classification. However, it is not perfect. The disadvantages are that the theory only really covers the determination of the parameters for a given value of the regularization and kernel parameters and choice of kernel.

In the research works, we have compared the effectiveness of these methods by working out the Precision, Recall and F1-score of these models, and recorded the training time of the two models as the efficiency indexes.

The average performances of different methods are displayed in Table XII. We use the 8-fold cross-validation technique to train and evaluate the dataset. The resulting correct differentiation rates represent the average of F1-Measure. The accuracy of our model is up to 97.40% with 8-fold cross-validation.

As we can see from the results, our learning-to-rank approach based on MaxENT achieves significantly better results compared with traditional classification approaches. The SVM model, which is believed to be suitable for classification, did not perform well in our setting. The comparison between MaxENT and DNN shows the effectiveness of our model. It reflects that MaxENT-BLP could make relatively appropriate system decisions when receiving user access requests from another perspective.

Fig. 6(a) gives the execution time of the other four traditional classification methods to train all dataset with 8-fold cross-validation. Fig. 6(b) displays the training time under different proportions of the datasets. These figures indicate that Deep Neural Networks are very time-consuming method, and the same goes for SVM and KNN. Bayes is efficient but not accurate. However, MaxENT-BLP is time-saving with a satisfactory effect. These experimental results prove that the model of MaxENT is sometimes better than deep learning methods for some massive training dataset with rare features.

B. Discussion for the Adversarial Samples From Malware Classifier

There are some attacks often instantiated by adversarial examples: legitimate inputs altered by adding small, often imperceptible, perturbations to force a learned classifier to misclassify the resulting adversarial inputs, while remaining correctly classified by a human observer [27].

Literature [27] pointed out that the parameter training can resist the attacks from the adversarial sample if the processing satisfies the following three conditions: (a) the capabilities required are limited to observing output class labels; (b) the number of labels queried is limited; and (c) the approach applies and scales to different machine learning classifier types.

In our parameter training, the attackers only need to know the output tag set of the target model, which is the result of the model inference in the MaxENT-BLP, and this processing is easily undetected due to small input data. Because the output is a finite set, it is obvious that the limited output itself satisfies our requirements for system security. Because the original model is based on traditional machine learning methods, the adversarial regions generated by this method between the classification boundary and the real boundary are easily obtained by computing the gradient for the output results (similar to the algorithms in [36] and [37]).

Actually, the attack based on adversarial samples for BFGS/L-BFGS cannot be prevented by the distillation techniques. This is largely due to the distillation method springing from the compression of the neural network, and this

technology is only suitable for reducing and preventing the attacks based on adversarial samples for deep learning models.

After further thinking, we found that the attacks based on adversarial samples generally achieve their purposes by modifying some feature values without affecting the subjective judgement of the sample content. For instance, in deep learning models, the classification features are usually all of the pixels in a picture, and for text classification by semantics, each word in an article is a feature. The slight changes in some very small parts of the features for this type of samples cannot change the sample itself and influence the subjective judgement of human to the sample classifications.

In this sense, adversarial samples may only be suitable for the feature value continuous model. For example, if we add a few pixels to a picture of a bird, I think the animal in this picture is still a bird. For the same reason, the main idea of a long article cannot be changed if we modify some irrelevant words. In this manner, slight variations in the feature values cannot change the sample, but that may affect the results of the sample classification.

The distribution of the feature value in the MaxENT-BLP is a typical discrete model. First, the length of each feature item is fixed, and in the preprocessing before model training, there is a filter function to the value of each feature item, and the characters that exceed the fixed length will be filtered. Hence, for generation of the adversarial samples, they can only modify the values of each feature item, and new characters added are invalid.

However, compared to a feature value continuous model such as a picture, the feature value modification is not the same thing. In MaxENT-BLP, any slight changes would change the sample itself. It would be changed to another sample, making it completely not the original one. For instance, for the feature values in Table III, any numerical changes will change the original significances of the subjects, objects, or operations from any subjects to objects. That is, the changes of the feature values would change the sample of access logs to another record. For different input samples, the outputs of the MaxENT-BLP are obviously independent.

The following contents prove and analyse how adversarial samples cannot be generated in MaxENT-BLP:

First, Because the precision of the features is limited, it is not rational for the classifier to respond differently to an input x than to an adversarial input: $\widetilde{x} = x + \eta$, if every element of the perturbation is smaller than the precision of the features. Formally, for problems with well-separated classes, we expect the classifier to assign the same class to x and $\widetilde{x}$ so long as $\|\eta\|_\infty < \varepsilon$, where is small enough to be discarded by the sensor or data storage apparatus associated with our problem. For the dot product of the adversarial samples and weight vector:

$$\omega^T \cdot \widetilde{x} = \omega^T \cdot x + \omega^T \cdot \eta \tag{25}$$

The adversarial perturbation causes the activation to grow by ω^T. We can maximize this increase subject to the max norm constraint on ω by assigning $\eta = sign(\omega)$.

If ω has n dimensions and the average magnitude of an element of the weight vector is m, then the activation will grow by $m \times n$. Because $\| \eta \|_\infty$ does not grow with the dimensionality of the problem but the change in activation caused by perturbation by η can grow linearly with n, for high dimensional problems, we can make many infinitesimal changes to the input that add up to one large change to the output.

In this way, with a disturbance η, it will cause a normal classifier to generate a different output from the expectation for the input x and $\widetilde{x} = x + \eta$.

Let θ be the parameters of a model, x the input to the model, y the targets associated with x (for machine learning tasks that have targets) and $J(\theta, x, y)$ the cost used to train the neural network. We can linearize the cost function around the current value of θ, obtaining an optimal max-norm constrained perturbation of this function.

$$\eta = \epsilon sign(\nabla_x J(\theta, x, y)) \tag{26}$$

In our application scenario, the actual input is a self-defined feature vector $\overrightarrow{x} = (\sigma_1, \gamma, \sigma_2, o_j, x)$ the deep learning model. If the malicious software wants to attack the MaxENT-BLP model, some necessary disturbances η would be added to the feature vector $\overrightarrow{x}$. As the above discussion, the features in this model are sparse, and the feature values of this model are discrete not continuous. If we add a disturbances η to a feature vector $\overrightarrow{x}$, no matter how small, the input sample would be changed to another sample: $\widetilde{x}^* = x + \eta$. In this manner, the case in which the classifier outputs different results is reasonable.

Therefore, we can draw the conclusion that: if the space of the feature value is continuous, and the adversarial samples do exist in the original input data, the parameter training in the MaxENT-BLP model can be attacked by this method. However, the adversarial samples cannot be generated from the input feature vectors of this model, because the value space of this model is discrete.

Hence, to protect the sample data of our model, we can have other secure courses, and the simplest method is integrity verification. For example, the sample data can be validated by computing and verifying its hash code.

VIII. Conclusion

Traditional access control models often lack the ability to perceive unknown threats. By training the historical access logs based on data training methods, MaxENT-BLP breaks the limitation in security state perception and self-optimization. Because most methods of current threat discovery technologies based on machine learning are often just used to detect unrecognizable intrusions, the major contribution of this paper is the proposal of a model training method that adjusts the decisions of the special BLP policies according to the secure annotations of historical data. This model can optimize the special insecure access control polices as the system runs.

The security of the model itself is fully considered in MaxENT-BLP. Because any slight disturbance added to the feature space would change one sample into another, this paper demonstrates that the adversarial sample cannot be generated from the training samples of MaxENT-BLP, and this model is hard to influence because its feature values are discrete.

The experimental results verify the learning ability of MaxENT-BLP, which can help systems to avoid risks and losses.

REFERENCES

[1] R. S. Sandhu and P. Samarati, "Access control: Principle and practice," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 40–48, Sep. 1994.

[2] D. E. Bell, *Secure Computer Systems: Mathematical Foundations*. Bedford, MA, USA: MITRE, 1973.

[3] S. Smalley and P. Loscocco, "Integrating flexible support for security policies into the linux operating system," in *Proc. USENIX Annu. Tech. Conf.*, 2001, pp. 1–15.

[4] M.-A. Jeong, J.-J. Kim, and Y. Won, "A flexible database security system using multiple access control policies," in *Proc. 4th Int. Conf. Parallel Distrib. Comput. Appl. Technol.*, Aug. 2003, pp. 236–240.

[5] P. Watson, "A multi-level security model for partitioningworkflows over federated clouds," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci.*, Nov./Dec. 2011, pp. 180–188.

[6] D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, *Machine Learning, Neural and Statistical Classification*. Upper Saddle River, NJ, USA: Ellis Horwood, 1994.

[7] A. Aztiria, J. C. Augusto, R. Basagoiti, A. Izaguirre, and D. J. Cook, "Learning frequent behaviors of the users in intelligent environments," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 6, pp. 1265–1278, Nov. 2013.

[8] P. Cheeseman and J. C. Stutz, "Bayesian classification (AutoClass): Theory and results," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Menlo Park, CA, USA: AAAI, 1996, pp. 153–180.

[9] T. Denoeux, "A k-nearest neighbor classification rule based on Dempster–Shafer theory," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 5, pp. 804–813, May 1995.

[10] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[11] S. R. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998.

[12] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Comput. Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[13] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 282–289.

[14] I. J. Myung and A. D. Bailey, Jr., "Maximum entropy aggregation of expert predictions," *Manage. Sci.*, vol. 42, no. 10, pp. 1420–1436, 1996.

[15] S. Ferilli, "WoMan: Logic-based workflow learning and management," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 6, pp. 744–756, Jun. 2014.

[16] F. Yamaguchi, F. Lindner, and K. Rieck, "Vulnerability extrapolation: Assisted discovery of vulnerabilities using machine learning," in *Proc. 5th USENIX Conf. Offensive Technol.*, 2011, p. 13.

[17] M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond heuristics: Learning to classify vulnerabilities and predict exploits," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 105–114.

[18] X.-B. Tan, W. P. Wang, and H.-S. Xi, "A hidden Markov model used in intrusion detection," *J. Comput. Res. Develop.*, vol. 40, no. 2, pp. 245–250, 2003.

[19] A. Mosenia, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "CABA: Continuous authentication based on bioaura," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 759–772, May 2017, doi: 10.1109/TC.2016.2622262.

[20] R. S. Puttini, Z. Marrakchi, and L. Mé, "A Bayesian classification model for real-time intrusion detection," in *Proc. AIP Conf.*, 2003, pp. 150–162.

[21] S. Tian, S. Mu, and C. Yin, "Sequence-similarity kernels for SVMs to detect anomalies in system calls," *Neurocomputing*, vol. 70, nos. 4–6, pp. 859–866, 2007.

[22] M. Kuhlmann, D. Shohat, and G. Schimpf, "Role mining—Revealing business roles for security administration using data mining technology," in *Proc. 8th Symp. Access Control Models Technol.*, Como, Italy, 2003, pp. 179–186.

[23] L. Fang and Y. Guo, "A survey of role mining methods in role-based access control system," in *Proc. APWeb Workshops*, vol. 8710. 2014, pp. 291–300.

[24] M. Jafari, A. Chinaei, K. Barker, and M. Fathian, "Role mining in access history logs," *Int. J. Comput. Inf. Syst. Ind. Manage. Appl.*, vol. 1, pp. 258–265, Dec. 2009.

[25] J. Hwang, T. Xie, V. Hu, and M. Altunay, "Mining likely properties of access control policies via association rule mining," in *Proc. IFIP Annu. Conf. Data Appl. Secur. Privacy*, 2010, pp. 193–208.

[26] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel. (2016). "Adversarial perturbations against deep neural networks for Malware classification." [Online]. Available: https://arxiv.org/abs/1606.04435

[27] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. (2016). "Practical black-box attacks against machine learning." [Online]. Available: https://arxiv.org/abs/1602.02697

[28] P. McDaniel, N. Papernot, and Z. B. Celik, "Machine learning in adversarial settings," *IEEE Security Privacy*, vol. 14, no. 3, pp. 68–72, May/Jun. 2016.

[29] D.-Y. Yeung and Y. Ding, "Host-based intrusion detection using dynamic and static behavioral models," *Pattern Recognit.*, vol. 36, no. 1, pp. 229–243, 2003.

[30] Y. Chen and D. Liginlal, "A maximum entropy approach to feature selection in knowledge-based authentication," *Decision Support Syst.*, vol. 46, no. 1, pp. 388–398, 2008.

[31] C.-C. Lo and W.-J. Chen, "A hybrid information security risk assessment procedure considering interdependences between controls," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 247–257, 2012.

[32] D. E. Bell, "Looking back at the Bell-La Padula model," in *Proc. 21st Annu. Comput. Secur. Appl. Conf.*, Tucson, AZ, USA, 2015, pp. 1–15.

[33] K. Nigam, "Using maximum entropy for text classification," in *Proc. Int. Joint Conf. Artif. Intell. Workshop Mach. Learn. Inf. Filtering (IJCAI)*, Stockholm, Sweden, Jul./Aug. 1999, pp. 61–67.

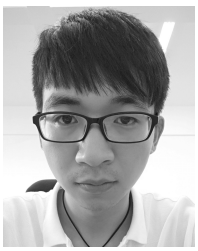
[34] D.-H. Li and M. Fukushima, "On the global convergence of the BFGS method for nonconvex unconstrained optimization problems," *SIAM J. Optim.*, vol. 11, no. 4, pp. 1054–1064, 2001.

[35] hankcs. (2014). *MaxEnt-Master Tools Package, Software*. [Online]. Available: https://github.com/hankcs/MaxEnt

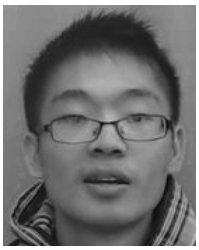
[36] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Mar. 2016, pp. 372–387.

[37] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015.

Zhuo Tang received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, China, in 2008. He is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University, and also the Associate Chair of the Department of Computing Science. His research interests include distributed computing system, cloud computing, and parallel processing for big data, including distributed machine learning, security model, parallel algorithms, and resources scheduling and management in these areas. He is a member of ACM and CCF.



Xiaofei Ding is currently pursuing the master's degree with the College of Information Science and Engineering, Hunan University, China. His research interests include the deep learning and natural language processing, especially the method and model of extracting the knowledge from the text data.



Ying Zhong is currently pursuing the master's degree with the College of Information Science and Engineering, Hunan University, China. In recent years, as one of the main researchers, he developed and implemented the Chinese word segmentation system and other NLP prototype systems based on the spark platform to improve the processing performance. His research interests include machine learning, big data, data analysis, and mining algorithms and their parallel implementation.

Li Yang received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, China, in 2013. She is currently an Assistant Professor of computer and communication engineering with the Changsha University of Science and Technology. She is also the Chief Architect for the IT systems in a large manufacturing corporation: SZZT Electronics, China. Her research interests include distributed information systems, bioinformatics, artificial intelligence, and biomedical text mining.

Keqin Li (F'15) is currently a Distinguished Professor of computer science with the State University of New York at New Paltz. He has published almost 500 journal articles, book chapters, and refereed conference papers. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU–GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things, and cyber-physical systems. He has received several best paper awards. He is currently or has served on the editorial boards of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON SERVICES COMPUTING, and IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING.