A Bayesian Intrusion Detection Framework

Shuai Fu
Faculty of Engineering and Computer Science
Concordia University
Montreal, Canada
f_shuai@encs.concordia.ca

Nizar Bouguila
Concordia Institute for Information Systems Engineering
Concordia University
Montreal, Canada
bouguila@ciise.concordia.ca

***Abstract*—This paper presents our work on a novel intrusion detection classifier based on asymmetric Gaussian mixture (AGM) model and reversible jump Markov chain Monte Carlo (RJMCMC) learning algorithm. Previous efforts reveal the fact that AGM outperforms classic Gaussian mixture model (GMM) by taking asymmetric datasets into consideration which provides more flexibility. Our RJMCMC implementation is based on a hybrid sampling-based approach which takes advantages of both Metropolis-Hastings (MH) and Gibbs sampling methods, therefore, simplifies mathematical complexity and extends adaptability of the model. Moreover, without giving a fixed components number in advance, RJMCMC applies a dynamic data-based strategy to identify the optimal components number throughout iterations which makes the model learning a self-adaptive process. Since the model is nondeterministic, Laplace approximation based marginal likelihood will be calculated for multiple runs as model selection procedure to improve the correctness and fitting accuracy. Both synthetic and challenging intrusion detection datasets are applied to our model to discover its merits.**

***Index Terms*—Asymmetric Gaussian Mixture, Metropolis-Hastings, Gibbs Sampling, RJMCMC, Laplace Approximation, Intrusion Detection**

I. Introduction

Along with the rapid growth of information technologies, personal and commercial behaviors tend to rely on computer network and Internet environments. However, based on the characteristics of networking, exposing sensitive privacy and valuable business secret online is extremely dangerous because accessibility and anonymity make network intrusions hard to be detected and traced, therefore, compromise network security. Cisco 2017 Annual Cybersecurity Report (ACR) [1] pointed out a crucial fact that more than one-third of organizations that experienced a breach in 2016 reported more than 20 percent of customer, opportunity and revenue loss. As a consequence, more than 90 percent of these organizations are improving threat defense technologies and processes by enhancing IT and security functions, increasing security training of employees and implementing risk mitigation techniques. Recently, machine learning-based intrusion detection solutions [2] [3] are drawing more attention because of their efficiency and flexibility.

Earlier intrusion prevention approaches, such as authentication, avoiding programming errors and encryption, were proven as insufficient because along with the increasing of the complexity of network-based software systems, exploitable weaknesses are inevitable due to programming issues. Moreover, authentication and encryption are not always reliable since credentials could be leaked and encryption algorithm could also be compromised by applying powerful hacking techniques to make the attack feasible. In consequence, once intrusion happens, detection will be harder than prevention and sometimes victims could not be even aware of it. Therefore, many supervised data mining solutions were proposed in terms of misuse and anomaly detection systems by establishing known intrusion scenarios, normal usage patterns and the sequential interrelations between user operations to identify intrusion behaviors [4]. However, the disadvantages of supervised intrusion detection systems are significant since predefined patterns and interrelations are inconsistent concerning the system upgrades and newly-founded intrusions which could lead to incessant intrusion detection system adjustment and affect its performance. Furthermore, inductive bias and overfitting problems caused by poor training datasets will also affect the accuracy of the systems. Therefore, researchers are paying more attention to unsupervised solution [5] [6] for seeking flexibility and robustness.

As an improvement of independent methodologies based on single mathematical model, mixture models [7]–[9] can be seen as a superimposition of certain mixture components having dependencies with each other, therefore, provide outstanding suitability and generality especially for adapting multidimensional datasets. Particularly, Gaussian mixture model (GMM) [10] demonstrated effective learning abilities in several regions such as computer vision, pattern recognition and data mining, especially for applications whose datasets are Gaussian-like. However, under a more general circumstances regarding to non-Gaussian or asymmetric datasets, asymmetric Gaussian mixture (AGM) model [11] is a better choice because its two variance parameters related to left and right parts of each component respectively in the mixture model, bringing better accuracy of fitting real applications. Therefore, we choose AGM as our intrusion detection model and its merits will be discussed in detail throughout this paper.

One of the most challenging tasks of applying mixture model is the parameter learning process. It could be achieved by applying maximum-likelihood based expectation maximization (EM) algorithm [12] which is widely deployed and has been proven as an effective estimation approach. However, as a deterministic solution, bad initialization and overfitting problems [13] [14] will significantly affect its efficiency and

usability by causing slow convergence and improper approximation, and eventually, compromise the accuracy of the model. Accordingly, in order to improve the parameter estimation, sampling-based stochastic Bayesian learning algorithms [15], [16], such as Markov Chain Monte Carlo (MCMC) based implementations [17], are proposed to address overfitting problems by introducing prior and posterior distributions for every mixture parameter which decouple the dependency between mixture parameters and mixture components, therefore, allow model adjustment by substituting proposed distributions to adapt to varied application datasets [18], [19]. In this paper, the base learning algorithm is chosen as a hybrid MCMC implementation, which is well known as Metropolis-Hastings within Gibbs sampling [13], based on both Metropolis-Hastings [20] and Gibbs sampling [21] methods because the main difficulty of applying traditional MCMC method is that, under some circumstances, direct sampling is not always straightforward that distributions of mixture parameters are latent and dependencies between parameters are unknown. By taking the advantages of both methods into consideration, proposal mixture parameters will be generated iteratively and decisions will be made by calculating the acceptance rates. Eventually, the optimal parameter values will be identified after convergence. Furthermore, we extend the learning algorithm by introducing reversible jump MCMC (RJMCMC) [14] concept where the mixture components number will be treated as an extra parameter which could be variant throughout iterations by increasing (component birth/death step) and decreasing (component merge/split step) mixture components, therefore, enables model transfer which significantly improves the learning flexibility of the AGM model. On the other hand, because of this stochastic learning process, the iterations could end up with different models by setting different initial components numbers. In order to evaluate the learning performances and identify the best-fit result, the model selection will be performed by calculating marginal likelihood [13].

The next sections are organized as follows. Section II elaborates the Bayesian model and learning algorithms. Section III is concentrating on experimental results analysis and Section IV gives a conclusion of the paper and proposes future research directions.

II. Bayesian model

A. Asymmetric Gaussian Mixture Model

The likelihood function of AGM model [11] with M mixture components can be illustrated as follows:

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{M} p_j p(X_i|\xi_j) \tag{1}$$

where $\mathcal{X} = (X_1, ..., X_N)$ reprensents the dataset with N observations, $\Theta = \{p_1, ..., p_M, \xi_1, ..., \xi_M\}$ defines the mixture parameters set of AGM mixture model including component weight p_j ($0 < p_j \leq 1$ and $\sum_{j=1}^{M} p_j$ = 1) and asymmetric Gaussian distribution (AGD) parameters set ξ_j for mixture component j. Assuming the dataset $\mathcal{X}$ is d-dimensional, for each observation $X_n = (x_{n1}, ..., x_{nd}) \in \mathcal{X}$, the probability density function [11] for j-th component of the model can be defined as follows:

$$p(X_n|\xi_j) \propto \prod_{k=1}^{d} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})} \times \begin{cases} \exp\left[-\frac{(x_{nk}-\mu_{jk})^2}{2(\sigma_{l_{jk}})^2}\right] & if\ x_{nk} < \mu_{jk} \\ \exp\left[-\frac{(x_{nk}-\mu_{jk})^2}{2(\sigma_{r_{jk}})^2}\right] & if\ x_{nk} \geqslant \mu_{jk} \end{cases} \tag{2}$$

parameters set of component j is $\xi_j = (\mu_j, \sigma_{lj}, \sigma_{rj})$ where $\mu_j = (\mu_{j1}, ..., \mu_{jd})$ is the mean, $\sigma_{lj} = (\sigma_{lj1}, ..., \sigma_{ljd})$ and $\sigma_{rj} = (\sigma_{rj1}, ..., \sigma_{rjd})$ represents the left and right standard deviation vectors of AGD .

We bring a M-dimensional membership vector Z to each observation $X_i \in \mathcal{X}, Z_i = (Z_{i1}, ..., Z_{iM})$, indicating which specific component X_i belongs to [22], such that:

$$Z_{ij} = \begin{cases} 1 & \text{if } X_i \text{ belongs to component } j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

that being said, $Z_{ij} = 1$ only when observation X_i has the highest probability of belonging to component j and accordingly, for other components, $Z_{ij} = 0$.

Hence, the complete likelihood function can be obtained by combining Eq. (1) and Eq. (3) as follows:

$$p(\mathcal{X}, Z|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} (p_j p(X_i|\xi_j))^{Z_{ij}} \tag{4}$$

B. Priors and Posteriors

As mentioned before, MH-within-Gibbs based RJMCMC learning algorithm implementation introduces definitions of priors and posteriors for mixture weights and parameters to avoid direct sampling. For a specific iteration t, since mixture weight p_j satisfies $0 < p_j \leq 1$ and $\sum_{j=1}^{M} p_j$ = 1, a natural choice of the prior is Dirichlet distribution [23] as follows:

$$\pi(p_j^{(t)}) \sim \mathcal{D}(\gamma_1, ..., \gamma_M) \tag{5}$$

where $\gamma = \gamma_j = \cdots = \gamma_M$ is a known hyperparameter. By taking the membership vector Z into account as a condition, the posterior probability of mixture weight p_j is defined as follows:

$$p(p_j^{(t)}|Z^{(t)}) \sim \mathcal{D}(\gamma_1 + n_1^{(t)}, ..., \gamma_M + n_M^{(t)}) \tag{6}$$

where n_j represents the number of observations of component j which could be calculated using membership vectors as follows:

$$n_j^{(t)} = \sum_{i=1}^{N} Z_{ij}\ (j = 1, ..., M) \tag{7}$$

The sampling process of mixture parameters employs the same concept. The proposal posterior distribution is $\xi^{(t)} \sim q(\xi|\xi^{(t-1)})$. More specifically, for parameters of AGM model

$\xi^{(t)} = (\mu^{(t)}, \sigma_l^{(t)}, \sigma_r^{(t)})$, we choose d-dimensional Gaussian distributions as posterior distributions respectively:

$$\begin{aligned} \mu_j^{(t)} &\sim \mathcal{N}_d(\mu_j^{(t-1)}, \Sigma) \\ \sigma_{lj}^{(t)} &\sim \mathcal{N}_d(\sigma_{lj}^{(t-1)}, \Sigma) \\ \sigma_{rj}^{(t)} &\sim \mathcal{N}_d(\sigma_{rj}^{(t-1)}, \Sigma) \end{aligned} \tag{8}$$

where Σ is d x d identity matrix which makes the sampling a random walk MCMC process. Correspondingly, the priors are $\mu \sim \mathcal{N}_d(\eta, \Sigma)$ and $\sigma_l, \sigma_r \sim \mathcal{N}_d(\tau, \Sigma)$ given known hyperparameters η and τ.

C. Learning Algorithm

MH-within-Gibbs: MH-within-Gibbs method, a sampling-based learning algorithm, performs random sampling from posteriors of parameters, and then calculates the acceptance ratio r in order to make a decision whether the new samples should be accepted or discarded for next iteration. Due to the usage of membership vector Z, the mixture weight p_j can be derived within Gibbs sampling part. Therefore, it will be excluded from the calculation of the acceptance ratio r which is defined as follows:

$$r = \frac{p(\mathcal{X}|\Theta^{(t)})\pi(\Theta^{(t)})q(\Theta^{(t-1)}|\Theta^{(t)})}{p(\mathcal{X}|\Theta^{(t-1)})\pi(\Theta^{(t-1)})q(\Theta^{(t)}|\Theta^{(t-1)})} \tag{9}$$

Further information about the calculation of acceptance ratio r is explained in Appendix A. Once acceptance ratio r is derived, acceptance probability $\alpha = min[1, r]$ [24] could be computed. Then $u \sim U_{[0,1]}$ is supposed to be generated randomly. If $\alpha < u$, the proposed move should be accepted and parameters should be updated by $p^{(t)}$ and $\xi^{(t)}$ for next iteration. Otherwise, we discard $p^{(t)}$, $\xi^{(t)}$ and set $p^{(t)} = p^{(t-1)}$, $\xi^{(t)} = \xi^{(t-1)}$.

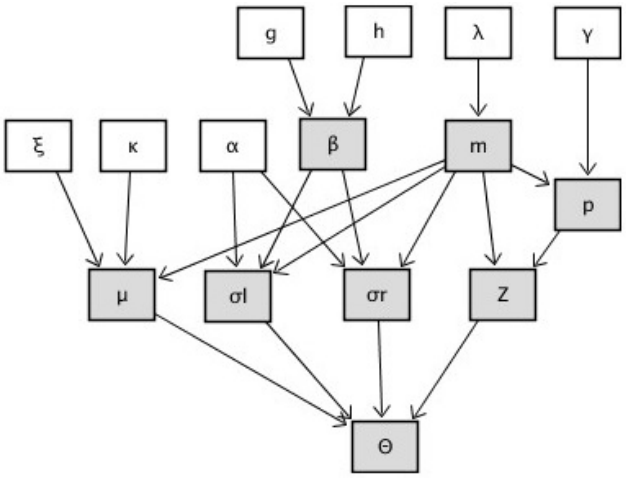

Fig. 1. DAG of RJMCMC parameter learning Bayesian network

RJMCMC moves: Traditional MH-within-Gibbs algorithm assumes that the components number M is given and persistent throughout the learning process. However, because of bad initialization or just information leakage, components number M could be inaccurate or unknown. Under these circumstances, RJMCMC algorithm presents its merits by providing extra four independent steps (birth/death steps and merge/split steps) into learning process which could change components number M, therefore, brings more generalities.

In practice, within every RJMCMC learning iteration, the current components number m is considered as an extra parameter which has a proposed Poisson prior $\mathcal{P}(\lambda)$ with $\lambda = 4$ particularly in our case [10]. Accordingly, let M_{min} and M_{max} denote the minimum and maximum number of components M, and assume the probabilities of performing birth/split and death/merge steps are b_m and $d_m = 1 - b_m$ for $m = M_{min}, \ldots, M_{max}$ respectively. Obviously, $b_{M_{max}} = 0$ and $d_{M_{min}} = 0$. Correspondingly, $d_{M_{max}} = 1 - b_{M_{max}} = 1$ and $b_{M_{min}} = 1 - d_{M_{min}} = 1$. For $m = M_{min}+1, \ldots, M_{max}-1$, for simplification purpose, we choose the same value for both b_m and d_m as $b_m = d_m = 0.5$. Within every iteration, we generate a random value $u' \sim U_{[0,1]}$ respectively for the four RJMCMC steps. If $b_m >= u'$ or $d_m >= u'$, birth/split or death/merge steps should be performed correspondingly [10].

Merge and Split Steps: Randomly choose two components (j_1, j_2) satisfying that $\mu_{j_1} < \mu_{j_2}$ with no other μ_j in the interval $[\mu_{j_1}, \mu_{j_2}]$. The newly merged component j' will contain the observations that previously belonged to both component j_1 and j_2. Meanwhile, reduce current value of components number m to $m-1$, then calculate mixture weight and parameters for j' as follows:

$$\begin{aligned} p_{j'} &= p_{j_1} + p_{j_2} \\ p_{j'}\mu_{j'} &= p_{j_1}\mu_{j_1} + p_{j_2}\mu_{j_2} \\ p_{j'}(\mu_{j'}^2 + \sigma_{j'l}^2) &= p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 l}^2) \\ &\quad + p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 l}^2) \\ p_{j'}(\mu_{j'}^2 + \sigma_{j'r}^2) &= p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 r}^2) \\ &\quad + p_{j_1}(\mu_{j_1}^2 + \sigma_{j_1 r}^2) \end{aligned} \tag{10}$$

As a reverse of merge step, we split component j' into two (j_1 and j_2) with 3 degrees of freedom $(u_1 \sim Beta(2,2), u_2 \sim Beta(2,2), u_3 \sim Beta(1,1))$ and, accordingly, increase m to $m+1$. Therefore, mixture parameters for split components can

be calculated as follows:

$$
\begin{aligned}
p_{j_1} &= p_{j'}u_1, p_{j_2} = p_{j'}u_2 \\
\mu_{j_1} &= \mu_{j'} - \frac{u_2(\sigma_{j'l} + \sigma_{j'r})}{2}\sqrt{\frac{p_{j_2}}{p_{j_1}}} \\
\mu_{j_2} &= \mu_{j'} + \frac{u_2(\sigma_{j'l} + \sigma_{j'r})}{2}\sqrt{\frac{p_{j_1}}{p_{j_2}}} \\
\sigma^2_{j_1 l} &= u_3(1-u_2^2)\sigma^2_{j'l}\frac{p_{j'}}{p_{j_1}} \\
\sigma^2_{j_1 r} &= u_3(1-u_2^2)\sigma^2_{j'r}\frac{p_{j'}}{p_{j_1}} \\
\sigma^2_{j_2 l} &= (1-u_3)(1-u_2^2)\sigma^2_{j'l}\frac{p_{j'}}{p_{j_2}} \\
\sigma^2_{j_2 r} &= (1-u_3)(1-u_2^2)\sigma^2_{j'r}\frac{p_{j'}}{p_{j_2}}
\end{aligned} \tag{11}
$$

In order to decide whether the merge and split steps should be accepted or not, the acceptance probability [10] can be derived as follows:

$$
\begin{aligned}
\mathcal{A} = &\frac{p(\mathcal{X}, Z|\Theta')}{p(\mathcal{X}, Z|\Theta)}\frac{m'\mathcal{P}(m'|\lambda)}{\mathcal{P}(m|\lambda)}\frac{p_{j_1}^{\gamma-1+n_1}p_{j_2}^{\gamma-1+n_2}}{p_{j'}^{\gamma-1+n_1+n_2}Beta(\gamma, m\gamma)} \\
&\times\sqrt{\frac{\kappa}{2\pi}}\exp[-\frac{1}{2}\kappa(\mu_{j_1}-\xi)+(\mu_{j_2}-\xi)+(\mu_{j'}-\xi)] \\
&\times\frac{\beta^\alpha}{\Gamma(\alpha)}(\frac{\sigma^2_{j_1 l}\sigma^2_{j_1 r}\sigma^2_{j_2 l}\sigma^2_{j_2 r}}{\sigma^2_{j'l}\sigma^2_{j'r}})^{-\alpha-1} \\
&\times\exp[-\beta(\sigma^2_{j_1 l}+\sigma^2_{j_1 r}+\sigma^2_{j_2 l}+\sigma^2_{j_2 r}-\sigma^2_{j'l}-\sigma^2_{j'r})] \\
&\times\frac{d_{m'}}{b_m P_{alloc}}[Beta(\mu_1|2,2)Beta(\mu_2|2,2)Beta(\mu_3|1,1)]^{-1} \\
&\times\frac{p_{j'}|\mu_{j_1}-\mu_{j_2}|\sigma^2_{j_1 l}\sigma^2_{j_1 r}\sigma^2_{j_2 l}\sigma^2_{j_2 r}}{\mu_2(1-\mu_2^2)\mu_3(1-\mu_3)\sigma^2_{j'l}\sigma^2_{j'r}}
\end{aligned} \tag{12}
$$

where Θ' and $m' = m+1$ denote the mixture parameters set and the components number respectively before merge or after split steps. κ is a known hyperparameter and ξ is the midpoint of the variation interval of the involved data observations. Besides, P_{alloc} is the probability of which this particular allocation is made. Therefore, the acceptance probability for merge step is $\min(1, \mathcal{A})$ and, correspondingly, for split step is $\min(1, \mathcal{A}^{-1})$.

Birth and Death Steps: Compared to merge and split steps, birth and death steps are relatively straightforward because the newborn and dead components are empty ones which means parameter re-calculation is not needed. Mixture weight p_{new} in birth step can be obtained by sampling from Beat distribution $p_{new} \sim Beta(1, m)$ and mixture parameters can be derived from the priors as follows [25]:

$$
\mu \sim \mathcal{N}(\xi, \kappa^{-1}), \quad \sigma_l^{-2}, \sigma_r^{-2} \sim \Gamma(\alpha, \beta), \quad \beta \sim \Gamma(g, h) \tag{13}
$$

where hyperparameters κ, α, g and h are estimated by data. For death step, an empty component should be randomly selected and deleted among the existing components if there is any. Otherwise, this step will be skipped. After birth and death steps, mixture weights p_j should be re-scaled so that all weights sum to 1. Acceptance probability for birth and death steps is also required as the one for merge and split steps whose definition is as follows:

$$
\begin{aligned}
\mathcal{A}' = &\frac{\mathcal{P}(m'|\lambda)}{\mathcal{P}(m|\lambda)}\frac{1}{Beta(m\gamma, \gamma)}p_{j'}^{\gamma-1}(1-p_{j'})^{N+m\gamma-m}m' \\
&\times\frac{d_{m'}}{(m_0+1)b_m}\frac{1}{Beta(p_{j'}|1, m)}(1-p_{j'})^m
\end{aligned} \tag{14}
$$

where m_0 is the amount of empty components. Thus, the probabilities of occurrence of birth and death steps are $\min(1, \mathcal{A}')$ and $\min(1, \mathcal{A}'^{-1})$ [10].

Finally, Figure 1 describes the dependencies between constants and variables involved in the Bayesian network of RJMCMC mixture parameter learning, and then, a typical learning procedure of AGM can be summarized as follows:

Input: Data observations $\mathcal{X}$ and components number M
Output: AGM mixture parameter set Θ

1) Initialization
2) Step t: For $t = 1, \ldots$

Gibbs sampling part

a) Generate $Z^{(t)}$ from Eq. (3)
b) Compute $n_j^{(t)}$ from Eq. (7)
c) Generate $p_j^{(t)}$ from Eq. (6)

Metropolis-Hastings part

d) Sample $\xi_j^{(t)}$ $(\mu_j^{(t)}, \sigma_{lj}^{(t)}, \sigma_{rj}^{(t)})$ from Eqs. (8)
e) Compute acceptance ratio r from Eq. (9)
f) Generate $\alpha = min[1, r]$ and $u \sim U_{[0,1]}$
g) If $\alpha \geq u$ then $\xi^{(t)} = \xi^{(t-1)}$

RJMCMC part

h) Generate $u' \sim U_{[0,1]}$. If $b_m >= u'$, perform split or birth step, then calculate acceptance probability $\mathcal{A}$. If the step is accepted, set $m = m+1$.

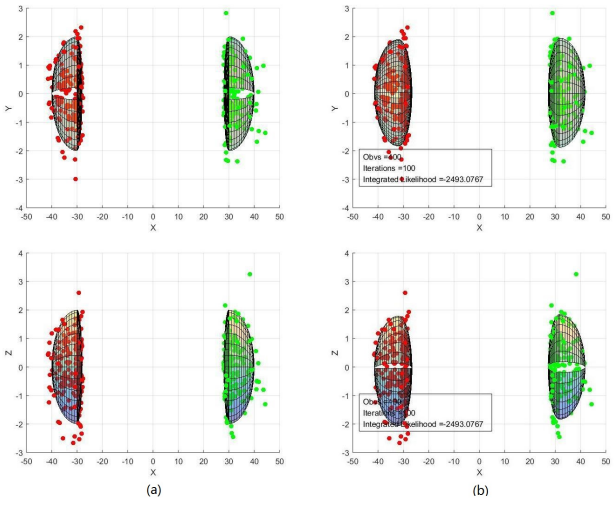

Fig. 2. (a) Original synthetic data grouping; (b) AGM clustering results

i) Generate $u' \sim U_{[0,1]}$. If $d_m >= u'$, perform merge or death step, then calculate acceptance probability $\mathcal{A}'$. If the step is accepted, set $m = m - 1$.

D. Model Selection

Theoretically, RJMCMC learning process should always be able to derive the optimal components number M. However, because of the stochastic sampling, improper proposal distributions or bad initialization parameters, learning result based on a single estimation run is not always satisfactory. In order to establish a robust parameter estimation algorithm, we evaluate the estimation outputs derived from multiple RJMCMC runs with different initial values of components number by calculating their marginal likelihood with the Laplace approximation [13] on the logarithm scale which is defined as follows:

$$\log(p(\mathcal{X}|M)) = \log(p(\mathcal{X}|\hat{\Theta}, M)) + \log(\pi(\hat{\Theta}|M)) + \frac{N_p}{2}\log(2\pi) + \frac{1}{2}\log(|H(\hat{\Theta})|) \quad (15)$$

where $\hat{\Theta}$ denotes the proposed optimal parameter set derived from a specific learning process and $\pi(\hat{\Theta}|M)$ is the prior density of mixture parameters as well as its Hessian matrix $H(\hat{\Theta})$ which is asymptotically equal to the posterior variance matrix.

III. Experimental results

A. Synthetic Data

Before testing the AGM model using real application datasets, a feasibility verification step is achieved against a 3-dimensional synthetic dataset having 400 observations composed of two clusters. Hyperparameters related to the calculations of mixture weights and parameters are set as $\gamma_j = 1$ and both η and τ are considered as d-dimensional zero vectors. Consequently, hyperparameters for priors of mixture parameters are set as follows [26]:

$$\kappa = \frac{1}{\mathcal{R}^2}, \quad \alpha = 3, \quad g = 0.3, \quad h = \frac{100g}{\alpha\mathcal{R}^2} \quad (16)$$

where $\mathcal{R}$ is the interval of variation of observations.

Figure 2 compares original grouping and RJMCMC learning results and Table II represents initial and estimated mixture parameters given components number $M = 2$. Moreover, The Euclidean distances are calculated to evaluate the accuracy of the AGM model. Based on synthetic test results, AGM model demonstrated promising unsupervised clustering by providing satisfactory deviation and grouping accuracy.

B. Intrusion Detection

A challenging NSL-KDD'99 intrusion detection database [27] is selected for our target application which was used for the Third International Knowledge Discovery and Data Mining Tools Competition. In order to simplify the testing

TABLE II
Accuracy Analysis ($M = 2$)

Component number $j = 1$	**Mean** (μ_j)	**Left standard deviation** (σ_{lj})	**Right standard deviation** (σ_{rj})
ξ	-30.00, 0.00, 0.00	5.00, 1.00, 1.00	1.00, 1.00, 1.00
$\hat{\xi}$	-31.9, 0.05, -0.05	4.78, 0.94, 1.01	2.65, 0.92, 0.92
Euclidean Distance	1.882	0.226	1.657
Component number $j = 2$	**Mean** (μ_j)	**Left standard deviation** (σ_{lj})	**Right standard deviation** (σ_{rj})
ξ	30.00, 0.00, 0.00	1.00, 1.00, 1.00	5.00, 1.00, 1.00
$\hat{\xi}$	31.70, 0.02, -0.01	2.24, 0.95, 0.91	5.06, 0.97, 0.93
Euclidean Distance	1.70	1.247	0.099

TABLE III
Translation and Normalization of Internet Protocols (Enumerated Values)

Internet Protocols	***Number of Occurrences***	***Normalized Values***
ICMP	1655	0
UDP	3011	0.071867
TCP	20526	1

TABLE IV
AGM Statistics

Init. Comp. Number	***Estimated Comp. Number***	***Accuracy***	***Marginal Likelihood***
1	2	60.86%	−1.3596e6
2	2	52.34%	−1.9886e6
3	2	53.39%	−2.2218e6

TABLE I
Original NSL-KDD data records

No	***Value***
1	0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00,normal
2	0,udp,other,SF,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0.00,0.00,0.00,0.00,0.08,0.15,0.00,255,1,0.00,0.60,0.88,0.00,0.00,0.00,0.00,0.00,normal
3	0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1.00,1.00,0.00,0.00,0.05,0.07,0.00,255,26,0.10,0.05,0.00,0.00,1.00,1.00,0.00,0.00,neptune
4	0,tcp,http,SF,232,8153,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5,0.20,0.20,0.00,0.00,1.00,0.00,0.00,30,255,1.00,0.00,0.03,0.04,0.03,0.01,0.00,0.01,normal
5	0,tcp,http,SF,199,420,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,30,32,0.00,0.00,0.00,0.00,1.00,0.00,0.09,255,255,1.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,normal
6	0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1,16,1.00,0.00,1.00,1.00,0.00,0.00,0.00,0.00,ipsweep

procedure and improve the fitting accuracy, we choose 20% of the whole database containing 25192 records clustered into two groups with 11743 intrusions and 13449 normal behaviors. Each data instance consists of 41 attributes (38 continuous or discrete numerical and 3 symbolic)(Table I). All the instances are grouped as either normal or attacks including DoS, Prob, U2R and R2L. Before applying our test model to the database, we need to translate non-numerical attributes to numerical and then, normalize records properly to ensure an accurate result. Therefore, enumerated and discrete values are substituted by their number of appearances in the whole database which could reflect their density distribution. Then, we apply feature scaling method to normalize numerical attributes to the range between 0 and 1 as follows:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{17}$$

where x and x' are the attribute values before and after normalization as well as the maximum and minimum values $max(x)$ and $min(x)$. An example of normalizing Internet Protocol attribute values is given in Table III. In this way, we could apply unified proposal distribution to every dimension using universal value of hyperparameter Σ during the random walk MCMC sampling.

We deploy our AGM model to the dataset with initial components number set between 1 and 3 and the learning iteration limit set to 30 loops. To better evaluate the performance and accuracy of our model under different initial number of components, at the end of each estimation, the fitting accuracy and marginal likelihood [13] are recorded into Table IV. Obviously, the best result is the first run with initial components number equals to 1 which has the largest marginal likelihood value $-1.3596\text{e}6$ and accuracy percentage 60.86%. In order to better evaluate the model performance, the compared specifications derived from confusion matrices of AGM and GGM are shown in Table V. In general, compared to GMM, AGM performs in a more accurate way for intrusion detection in terms of higher accuracy and much lower False Negative Rate indicating that AGM causes less false alarms of attack behaviors. However, it has a similar False Positive Rate and a lower precision value which means the intrusion detection abilities of both models are not satisfactory because of the absence of feature selection [28]–[30]. In addition, high-dimensional database will dramatically increase the noise during the clustering and eventually affect the usability of the model for real applications. Therefore, data-oriented model adjustment and dimensionality reduction techniques should be involved in order to mitigate this problem and achieve better detection outcomes.

TABLE V
CONFUSION MATRICES AND STATISTICS OF GMM AND AGM

GMM

	NF [a]	***F*** [b]
NF	4238	7505
F	3397	10052

AGM

	NF	***F***
NF	2456	9278
F	582	12867

	GMM	***AGM***
Accuracy	53.39%	60.86%
Precision	36.09%	20.93%
False Positive Rate	42.75%	41.90%
False Negative Rate	44.49%	19.16%

[a]Non fault-prone, [b]Fault-prone.

IV. CONCLUSION AND FUTURE WORK

A novel Bayesian framework based on asymmetric Gaussian mixture model and reversible jump MCMC is proposed to address intrusion detection problem. RJMCMC reinforced classic MCMC methodology by introducing extra four steps of split, merge, birth and death which enabled the transfer between AGM models with different components numbers and, consequently, different mixture parameters. Therefore, it improves the flexibility and generality of the learning process. A horizontal comparison with Gaussian mixture model is made to show the merits of this model. The results reveal the fact that AGM outperforms GMM from most statistical specifications. However, the performances of both models are not satisfactory especially for high-dimensional datasets which implies that further model adjustments and improvements are necessary. Currently, we are working on the introduction of a feature selection step within this model in order to further improve its capabilities [31]–[35].

APPENDIX A

A. Derivation of Acceptance Ratio r by Eq. (9)

The derivation of acceptance ratio r is based on the assumption that mixture parameters are independent from each other which means that:

$$\begin{aligned}\pi(\Theta) &= \pi(p, \xi) = \pi(\xi) \\ &= \prod_{j=1}^{M} \pi(\mu_j)\pi(\sigma_{lj})\pi(\sigma_{rj}) \\ &= \prod_{j=1}^{M} \mathcal{N}_d(\mu_j|\eta, \Sigma)\mathcal{N}_d(\sigma_{lj}|\tau, \Sigma)\mathcal{N}_d(\sigma_{rj}|\tau, \Sigma)\end{aligned} \tag{18}$$

in Eq. (18), since the mixture weigh p is generated following Gibbs sampling method whose acceptance ratio is always 1, it should be excluded from Metropolis-Hastings estimation step. Accordingly, apply the same rule to the proposal distribution as well:

$$\begin{aligned}&q(\Theta^{(t)}|\Theta^{(t-1)}) = q(\xi^{(t)}|\xi^{(t-1)}) = \\ &\prod_{j=1}^{M} \mathcal{N}_d(\mu_j^{(t)}|\mu_j^{(t-1)}, \Sigma)\mathcal{N}_d(\sigma_{lj}^{(t)}|\sigma_{lj}^{(t-1)}, \Sigma)\mathcal{N}_d(\sigma_{rj}^{(t)}|\sigma_{rj}^{(t-1)}, \Sigma)\end{aligned} \tag{19}$$

by combining Eqs. (2) (4) (8) (18) and (19), equation (9) can be written as follows:

$$
\begin{aligned}
r &= \frac{p(\mathcal{X}|\Theta^{(t)})\pi(\Theta^{(t)})q(\Theta^{(t-1)}|\Theta^{(t)})}{p(\mathcal{X}|\Theta^{(t-1)})\pi(\Theta^{(t-1)})q(\Theta^{(t)}|\Theta^{(t-1)})} \\
&= \prod_{i=i}^{N}\prod_{j=1}^{M}\Big(\frac{p(X_i|\mu_j^{(t)},\sigma_{lj}^{(t)},\sigma_{rj}^{(t)})}{p(X_i|\mu_j^{(t-1)},\sigma_{lj}^{(t-1)},\sigma_{rj}^{(t-1)})} \\
&\quad\times\frac{\mathcal{N}_d(\mu_j^{(t)}|\eta,\Sigma)\mathcal{N}_d(\sigma_{lj}^{(t)}|\tau,\Sigma)\mathcal{N}_d(\sigma_{rj}^{(t)}|\tau,\Sigma)}{\mathcal{N}_d(\mu_j^{(t-1)}|\eta,\Sigma)\mathcal{N}_d(\sigma_{lj}^{(t-1)}|\tau,\Sigma)\mathcal{N}_d(\sigma_{rj}^{(t-1)}|\tau,\Sigma)} \\
&\times\frac{\mathcal{N}_d(\mu_j^{(t-1)}|\mu_j^{(t)},\Sigma)\mathcal{N}_d(\sigma_{lj}^{(t-1)}|\sigma_{lj}^{(t)},\Sigma)\mathcal{N}_d(\sigma_{rj}^{(t-1)}|\sigma_{rj}^{(t)},\Sigma)}{\mathcal{N}_d(\mu_j^{(t)}|\mu_j^{(t-1)},\Sigma)\mathcal{N}_d(\sigma_{lj}^{(t)}|\sigma_{lj}^{(t-1)},\Sigma)\mathcal{N}_d(\sigma_{rj}^{(t)}|\sigma_{rj}^{(t-1)},\Sigma)}
\end{aligned}
\tag{20}
$$

ACKNOWLEDGMENT

The completion of this research work was made possible thanks to Concordia University via a Concordia University Research Chair Tier II.

REFERENCES

[1] Investor.cisco.com. (2018) Cisco 2017 annual cybersecurity report. [Online]. Available: https://investor.cisco.com/investor-relations/news-and-events/news/news-details/2017/Cisco-2017-Annual-Cybersecurity-Report-Chief-Security-Officers-Reveal-True-Cost-of-Breaches-And-The-Actions-That-Organizations-Are-Taking/default.aspx

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1153–1176, Secondquarter 2016.

[3] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484 – 497, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025516302547

[4] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," in *Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, USA, January 26-29, 1998*, A. D. Rubin, Ed. USENIX Association, 1998. [Online]. Available: https://www.usenix.org/conference/7th-usenix-security-symposium/data-mining-approaches-intrusion-detection

[5] N. Bouguila, "Bayesian hybrid generative discriminative learning based on finite liouville mixture models," *Pattern Recognition*, vol. 44, no. 6, pp. 1183–1200, 2011. [Online]. Available: https://doi.org/10.1016/j.patcog.2010.12.010

[6] M. Azam and N. Bouguila, "Unsupervised keyword spotting using bounded generalized gaussian mixture model with ICA," in *2015 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2015, Orlando, FL, USA, December 14-16, 2015*. IEEE, 2015, pp. 1150–1154.

[7] J. Yang, X. Liao, X. Yuan, P. Llull, D. J. Brady, G. Sapiro, and L. Carin, "Compressive sensing by learning a gaussian mixture model from measurements," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 106–119, Jan 2015.

[8] C. K. Wen, S. Jin, K. K. Wong, J. C. Chen, and P. Ting, "Channel estimation for massive mimo using gaussian-mixture bayesian learning," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1356–1368, March 2015.

[9] N. Bouguila, "Count data modeling and classification using finite mixtures of distributions," *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 186–198, 2011.

[10] S. Richardson and P. J. Green, "On bayesian analysis of mixtures with an unknown number of components (with discussion)," *Journal of the Royal Statistical Society: series B (statistical methodology)*, vol. 59, no. 4, pp. 731–792, 1997.

[11] T. Elguebaly and N. Bouguila, "Background subtraction using finite mixtures of asymmetric gaussian distributions and shadow detection," *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1145–1162, 2014.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[13] N. Bouguila, D. Ziou, and R. I. Hammoud, "On bayesian analysis of a finite generalized dirichlet mixture via a metropolis-within-gibbs sampling," *Pattern Anal. Appl.*, vol. 12, no. 2, pp. 151–166, 2009.

[14] N. Bouguila and T. Elguebaly, "A fully bayesian model based on reversible jump MCMC and finite beta mixtures for clustering," *Expert Syst. Appl.*, vol. 39, no. 5, pp. 5946–5959, 2012.

[15] S. Bourouis, M. A. Mashrgy, and N. Bouguila, "Bayesian learning of finite generalized inverted dirichlet mixtures: Application to object classification and forgery detection," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2329–2336, 2014. [Online]. Available: https://doi.org/10.1016/j.eswa.2013.09.030

[16] N. Bouguila, J. H. Wang, and A. B. Hamza, "A bayesian approach for software quality prediction," in *2008 4th International IEEE Conference Intelligent Systems*, vol. 2, Sept 2008, pp. 11–49–11–54.

[17] S. Fu and N. Bouguila, "Bayesian learning of finite asymmetric gaussian mixtures," in *in Proceedings of The 31st International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems Montreal, QC, CA, June 25-28, 2018*, 2018.

[18] N. Bouguila, D. Ziou, and R. I. Hammoud, "A bayesian non-gaussian mixture analysis: Application to eye modeling," in *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007. [Online]. Available: https://doi.org/10.1109/CVPR.2007.383439

[19] S. Fu and N. Bouguila, "Asymmetric gaussian mixtures with reversible jump MCMC," in *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE) (CCECE 2018)*, Quebec City, Canada, May 2018.

[20] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[21] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," in *Readings in Computer Vision*. Elsevier, 1987, pp. 564–584.

[22] N. Bouguila, D. Ziou, and E. Monga, "Practical bayesian estimation of a finite beta mixture through gibbs sampling and its applications," *Statistics and Computing*, vol. 16, no. 2, pp. 215–225, 2006.

[23] T. Elguebaly and N. Bouguila, "Simultaneous bayesian clustering and feature selection using rjmcmc-based learning of finite generalized dirichlet mixture models," *Signal Processing*, vol. 93, no. 6, pp. 1531–1546, 2013.

[24] D. Luengo and L. Martino, "Fully adaptive gaussian mixture metropolis-hastings algorithm," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*. IEEE, 2013, pp. 6148–6152.

[25] G. Casella, C. P. Robert, and M. T. Wells, "Mixture models, latent variables and partitioned importance sampling," *Statistical Methodology*, vol. 1, no. 1-2, pp. 1–18, 2004.

[26] M. Stephens, "Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods," *Annals of statistics*, pp. 40–74, 2000.

[27] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, Ottawa, Canada, July 8-10, 2009*. IEEE, 2009, pp. 1–6.

[28] N. Bouguila and D. Ziou, "A countably infinite mixture model for clustering and feature selection," *Knowl. Inf. Syst.*, vol. 33, no. 2, pp. 351–370, 2012.

[29] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognition*, vol. 64, pp. 141 – 158, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320316303545

[30] S. Boutemedjet, D. Ziou, and N. Bouguila, "Unsupervised feature selection for accurate recommendation of high-dimensional image data," in *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2007, pp. 177–184. [Online]. Available: http://papers.nips.cc/paper/3267-unsupervised-feature-selection-for-accurate-recommendation-of-high-dimensional-image-data

[31] W. Fan, N. Bouguila, and D. Ziou, "Unsupervised anomaly intrusion detection via localized bayesian feature selection," in *11th IEEE In-*

ternational Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011, D. J. Cook, J. Pei, W. Wang, O. R. Zaïane, and X. Wu, Eds. IEEE Computer Society, 2011, pp. 1032–1037.

[32] W. Fan and N. Bouguila, "Variational learning of a dirichlet process of generalized dirichlet distributions for simultaneous clustering and feature selection," *Pattern Recognition*, vol. 46, no. 10, pp. 2754–2769, 2013.

[33] N. Bouguila, "A model-based approach for discrete data clustering and feature weighting using MAP and stochastic complexity," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 12, pp. 1649–1664, 2009.

[34] S. Boutemedjet, D. Ziou, and N. Bouguila, "Model-based subspace clustering of non-gaussian data," *Neurocomputing*, vol. 73, no. 10-12, pp. 1730–1739, 2010.

[35] N. Bouguila, K. Almakadmeh, and S. Boutemedjet, "A finite mixture model for simultaneous high-dimensional clustering, localized feature selection and outlier rejection," *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6641–6656, 2012.