 This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

Bayesian Estimation of Inverted Beta Mixture Models with Extended Stochastic Variational Inference for Positive Vectors Classification

Yuping Lai, *Member, IEEE,* Lijuan Luo, Yanhui Guo, Heping Song, Wenbo Guan, and Hongying Meng, *Senior Member, IEEE*

Abstract—Finite inverted Beta mixture model (IBMM) has been proven to be efficient in modeling positive vectors. Under the traditional variational inference framework, the critical challenge in Bayesian estimation of IBMM is that the computational cost of performing inference with large datasets is prohibitively expensive, which often limits usages of Bayesian approaches to small datasets. An efficient alternative provided by the recently proposed stochastic variational inference (SVI) framework allows for efficient inference upon large datasets. Nevertheless, when using the SVI framework to tackle non-Gaussian statistical models, the evidence lower bound (ELBO) cannot be calculated explicitly due to the intractable moment computation. Therefore, the algorithm under the SVI framework cannot directly utilize stochastic optimization to optimize the ELBO and an analytically tractable solution cannot be derived. To address this problem, we propose an extended version of the SVI framework with more flexibilities namely the extended SVI (ESVI). This framework can be employed to many non-Gaussian statistical models. Firstly, some approximation strategies are applied to further lower the ELBO to avoid intractable moment calculation. Then, stochastic optimization with noisy natural gradients is used to optimize the lower bound. The excellent performance and effectiveness of the proposed method is verified in real data evaluation.

Index Terms—Extended stochastic variational inference; Mixture models; Bayesian estimation; Text categorization; Network traffic classification; Misuse intrusion detection.

1 Introduction

Positive vectors [1] emerge naturally in a wide range of real applications, such as text categorization [2], anomaly intrusion detection [3]–[5], object detection [6], software modules classification [1], and human action recognition [7]. Therefore, positive vectors modeling has become a vital research topic and gained increasing attention over the past years. Finite mixture modeling based on several non-Gaussian components has proven to be flexible and useful in modeling positive vectors, which are supposed to be drawn from homogenous populations. Such typical mixture models are finite Watson mixture model (WMM) [8], finite von-Mises Fisher Mixture Model (vFMM) [9], finite Beta mixture model (BMM) [2], finite Dirichlet mixture model (DMM) [10], finite Beta-Liouville mixture model (BLMM) [11], finite inverted Dirichlet mixture (IDMM) [1], finite inverted Beta mixture model (IBMM) [2], finite generalized Gamma mixture model (GGaMM) [12], and finite inverted Beta-Liouville mixture model (IBLMM) [13]. A great number of studies have shown that these mixtures are capable of providing much better modeling capabilities and performances than the commonly used finite Gaussian mixture model (GMM) in processing positive vectors. For instance, DMM, BLMM, and BMM have been proven to be much more efficient in modeling the proportional data [10], [11]. IDMM, IBMM, IBLMM, and GaMM have shown their advantages in modeling positive vectors [1], [2], [6], [13]. WMM and vFMM have demonstrated their advantages in modeling axially symmetric data. The IBMM, among others, is capable of providing high flexibility and easy application for modeling positive vectors [1]. IBMM has been extensively applied for text categorization [2], human action recognition [7], and image categorization [14].

Model learning is a significant issue in finite mixture modeling, which refers to the task of both estimating the model parameters and determining the number of mixture components. A number of approaches have been proposed to deal with this issue and they can be split into two groups namely deterministic and Bayesian methods. Deterministic methods generally adopt the conventional expectation-maximization (EM) algorithm [15] to maximize the data log-likelihood function and to optimize the model parameters. However, deterministic methods have some shortcomings, such as dependency on initialization, over-fitting, etc, and they alone fail to determine the optimal component number. They need to integrate some model selection criteria, such as Akaike information crite-

- *Y. Lai is with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China.*
- *L. Luo is with the School of Business and Management, Shanghai International Studies University, Shanghai, China.*
- *Y. Guo is with the Department of Computer Science, University of Illinois Springfield, Springfield, IL USA.*
- *H. Song is with School of Computer Science and Communications Engineering, Jiangsu University, Zhenjiang, China.*
- *W. Guan is with with the Institute of Information Engineering, CAS, Beijing, China.*
- *H. Meng is with Electronic and Electrical Engineering Department, Brunel University London, U.K.*
- *The corresponding author is Y. Lai. Email: laiyp@bupt.edu.cn*

rion (AIC) [16], Bayesian information criterion (BIC) [17], minimum description length (MDL) [18], etc, to determine the optimal component number. Nevertheless, these approaches demand multiple evaluations of the selection criterion regarding diverse component numbers, which normally yields a heavy computational burden. Moreover, it is worth noting that for the maximization likelihood estimation of most non-Gaussian mixtures, there exists no analytically intractable solution for the model parameter estimation [19], [20]. Hence, iterative numerical computation (*e.g.*, Newton-Raphson methods) is commonly used to deal with this problem, which increases the computational cost further.

Bayesian methods overcome the weaknesses of deterministic methods, which determine the posterior densities through combining the observed data and the prior knowledge of the model's parameters using Bayes' rule. Because the computation of these posteriors often involves high dimensional integrations, it is generally computationally intractable. To address this issue, Bayesian methods [21] have considered the two most prominent strategies: MCMC sampling and variational inference to approximate the true posterior densities. However, a major shortcoming of MCMC methods is that their convergence can be hard to be diagnosed, which often limits their usage to small-scale problems. Variational inference offers an excellent alternative to computationally demanding sampling-based methods, which casts the inference problem as optimization and has been extensively applied in a wide range of applications including finite mixture models learning. Nevertheless, with the conventional variational inference framework, a closed-from solution to the Bayesian estimation of non-Gaussian mixture models cannot be obtained due to the intractable moment calculation. This problem can be solved elegantly through our recently proposed extended variational inference (EVI) framework [2], [4], [22]. However, the aforementioned approximation inference strategies require analyzing the whole dataset in each iteration and thus scale poorly on massive datasets.

Recently, SVI has gained considerable popularity in tackling the aforementioned issue [23], which aims at finding excellent posterior approximations of probabilistic models with massive datasets and has been successfully employed in a wide range of settings, such as topic models [23]–[25], hidden Markov models [26], Bayesian time series models [27], etc. The major idea behind the SVI framework is to utilize noise of the gradient based upon minibatches of data, which avoids an expensive gradient calculation on the entire dataset. Supposing the data are independent and identically distributed and the minibatches are appropriately scaled, the stochastic gradient is a noisy (but unbiased) estimate of the actual gradient. Nevertheless, its application to most of the non-Gaussian statistical models is fairly understudied due to difficulty in carrying out the intractable moment calculation.

Motivated by the powerful and flexible modeling ability of IBMM and the excellent performance achieved by the SVI framework in recent years, this paper focuses on the Bayesian estimation of IBMM with the SVI framework. However, it is infeasible to derive an analytically tractable solution by SVI, due to the fact that the moment in the ELBO that involves functional forms of log-gamma function in their arguments is computationally intractable. To tackle this issue, we propose a novel extended stochastic variational inference (ESVI) framework to develop an analytically tractable alternative for stochastic variational learning of IBMM. This framework is particularly suitable for non-Gaussian statistical models with large-scale datasets. The basic idea behind this approach is that some lower-bound approximations subject to certain constraints are firstly introduced to the original ELBO within the conventional variational inference framework, such that it can be further lower bound in order to avoid the intractable moment evaluation. Then, we can apply stochastic optimization with noisy natural gradients to optimize the lower bound of the ELBO rather than the original ELBO. The effectiveness of the proposed ESVI framework to learn the IBMM is evaluated through real datasets which are generated from real-world challenging applications.

The key contributions of this paper can be summarized in the following three-folds:

- We propose a fairly efficient and attractive ESVI framework to derive an analytically tractable solution to the Bayesian estimation of the non-Gaussian statistical models, that lets us apply these models to analyze large-scale data. More importantly, this framework is able to be generalized to many settings.
- We develop an analytically tractable algorithm for IBMM with the proposed ESVI framework, which can achieve comparable performances compared to the one under the EVI framework for IBMM (EVI-IBMM) [2], and performs much better in terms of computational cost under the setting of the large data modeling.
- We apply the proposed ESVI-based IBMM to three real-life challenging applications, which consists of text categorization, network traffic classification, and misuse intrusion detection. The effectiveness and excellent performance have been validated through extensive comparisons.

The remainder of this paper is structured as follows: In Section 2, we give a brief overview of IBMM. In Section 3, we propose an efficient ESVI framework for non-Gaussian statistical models, and then apply it to the Bayesian estimation of IBMM. In Section 4, we report the experimental results obtained with real data. Conclusions and future works are drawn in the final section.

2 Finite Inverted Beta Mixture Model

This section provides a concise introduction to IBMM [2]. Let $\boldsymbol{\mathcal{X}}=\{\mathbf{x}_1,\cdots,\mathbf{x}_N\}$ denote a set of observations, which are drawn independently from an IBMM with M mixture components with the probability density function (PDF):

$$p(\mathbf{x}_n|\mathbf{U},\mathbf{V},\mathbf{\Pi})=\sum_{m=1}^{M}\pi_m\prod_{d=1}^{D}\mathrm{iBeta}(x_{nd}|u_{md},u_{md}), \quad (1)$$

 This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

where $\mathbf{x}_n = [x_n 1, \cdots, x_n D]^{\mathrm{T}}$, $\mathbf{U}=\{u_{md}\}$ and $\mathbf{V}=\{v_{md}\}$ denote parameter sets, $\mathbf{\Pi} = [\pi_1, \cdots, \pi_M]^{\mathrm{T}}$ denotes the mixing weights, which satisfies the following constraints:

$$0 < \pi_m < 1 \quad \text{and} \quad \sum_{m=1}^{M} \pi_m = 1. \tag{2}$$

Moreover, $\mathrm{iBeta}(x|u, v)$ is the inverted Beta distribution defined by [21]

$$\mathrm{iBeta}(x|u, v) = \frac{\Gamma(u+v)}{\Gamma(u)\Gamma(v)} x^{u-1}(1+x)^{-u-v}, x > 0, \tag{3}$$

where $\Gamma(\cdot)$ represents the Gamma function defined as $\Gamma(a) = \int_0^\infty s^{a-1}e^{-s}ds$. The likelihood of the observations $\boldsymbol{\mathcal{X}}$ is given by

$$p(\boldsymbol{\mathcal{X}}|\mathbf{U}, \mathbf{V}, \mathbf{\Pi}) = \prod_{n=1}^{N} \sum_{m=1}^{M} \pi_m \prod_{d=1}^{D} \mathrm{iBeta}(x_{nd}|u_{md}, v_{md}). \tag{4}$$

The observations $\boldsymbol{\mathcal{X}}$ are regarded as the incomplete data. To provide a proper complete data configuration for the IBMM, we introduce a label indicator vector $\mathbf{z}_n = [z_{n1}, \cdots, z_{nM}]^{\mathrm{T}}$ over each observation $\mathbf{x}_n$, with $z_{nm} \in (0, 1)$ and such that $z_{nm} = 1$ if $\mathbf{x}_n$ is viewed as generated by the mth mixture component, $z_{nm} = 0$ otherwise. Therefore, the latent variable model of the IBMM can be specified as follows:

$$p(\mathbf{Z}|\mathbf{\Pi}) = \prod_{n=1}^{N} \prod_{m=1}^{M} \pi_m^{z_{nm}}, \tag{5}$$

$$p(\boldsymbol{\mathcal{X}}, \mathbf{Z}|\mathbf{U}, \mathbf{V}) = \prod_{n=1}^{N} \prod_{m=1}^{M} \left[\prod_{d=1}^{D} \mathrm{iBeta}(x_{nd}|u_{md}, v_{md}) \right]^{z_{nm}}, \tag{6}$$

where $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_N]$ denotes the set of label indicator vectors. Note that the mixture model in (3) can be recovered by marginalizing (6) over z_{nm}, weighted by the prior (5).

To follow a SVI method for learning the IBMM, we need to impose prior distributions over the parameters $\mathbf{U}$ and $\mathbf{V}$. Since the inverted Beta distribution is a member of the exponential family, it has a formal conjugate prior. However, it cannot be applied in the SVI framework, because it is still defined with an integration expression that makes the closed form of the posterior distribution analytically intractable [2], [28]. We thus specify conjugate Gamma priors upon $\mathbf{U}$ and $\mathbf{V}$, as suggested in [6], by assuming that the Gamma parameters are statistically independent

$$p(\mathbf{U}) = \mathcal{G}(\mathbf{U}|\mathbf{G}, \mathbf{H}) = \prod_{m=1}^{M} \prod_{d=1}^{D} \mathcal{G}(u_{md}|g_{md}, h_{md}), \tag{7}$$

$$p(\mathbf{V}) = \mathcal{G}(\mathbf{V}|\mathbf{P}, \mathbf{Q}) = \prod_{m=1}^{M} \prod_{d=1}^{D} \mathcal{G}(v_{md}|p_{md}, q_{md}), \tag{8}$$

where $\mathbf{G}=\{g_{md}\}$, $\mathbf{H}=\{h_{md}\}$, $\mathbf{P}=\{p_{md}\}$, and $\mathbf{Q}=\{q_{md}\}$ denote the positive hyperparameters. Note that the mixing weights $\mathbf{\Pi}$ are treated as parameters rather than random variables in our case; thus, no prior is imposed on $\mathbf{\Pi}$. The joint distribution of all the random variables is given in (10). Note that in (9) $\mathbf{Z}$ represents local hidden variables (*i.e.*, a variable corresponding to each observation $\mathbf{x}_n$) and $\mathbf{\Theta} = \{\mathbf{U}, \mathbf{V}\}$ denotes the set of global hidden variables (*i.e.*, variables that are coupled to the entire set of observations). Fig. 1 illustrates a directed graphical representation of this model.

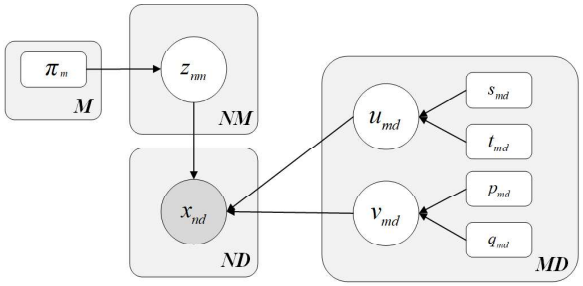


Fig. 1: Graphical representation of the Bayesian mixture model of the inverted Beta distributions. Nodes denote random variables, edges denote possible dependence, and plates indicate replication.

3 Model Learning

In this section, we describe a stochastic variational Bayesian learning approach for IBMM. Compared to batch learning algorithms, such as MCMC, variational inference, expectation propagation [21], [29], etc, stochastic algorithms are more effective in dealing with massive datasets [23], [24], [26], [27].

3.1 Extended Stochastic Variational Inference

The major purpose of Bayesian inference is to compute the posterior distribution of the latent variables. For most of the interesting mixture models, the computation of the posterior is intractable and approximation is required. VI is an optimization-based approach that approximates the intractable posterior $p(\mathbf{Z}, \mathbf{\Theta}|\boldsymbol{\mathcal{X}})$ with a variational distribution $q(\mathbf{Z}, \mathbf{\Theta})$ within a simpler family. Typically, a mean-field approximation is considered:

$$q(\mathbf{Z}, \mathbf{\Theta}) = q(\mathbf{\Theta}|\boldsymbol{\lambda}) \prod_{n=1}^{N} q(\mathbf{z}_n|\boldsymbol{\phi}_n), \tag{10}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}_n$ denote the global and local variational parameters respectively. These parameters are optimized to maximize the ELBO:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi}) = \sum_{n=1}^{N} \{ \mathrm{E}_q [\ln p(\mathbf{x}_n, \mathbf{z}_n|\mathbf{\Pi}, \mathbf{\Theta})] - \mathrm{E}_q [\ln q(\mathbf{z}_n)] \} \\ + \mathrm{E}_q [\ln p(\mathbf{\Theta})] - \mathrm{E}_q [\ln q(\mathbf{\Theta})] . \end{aligned} \tag{11}$$

Maximizing this bound is equivalent to minimizing the KL divergence between $p(\mathbf{Z}, \mathbf{\Theta}|\boldsymbol{\mathcal{X}}, \mathbf{\Pi})$ and $q(\mathbf{Z}, \mathbf{\Theta})$ [21].

Within the VI framework, the local parameters $\boldsymbol{\phi}$ and the global parameters $\boldsymbol{\lambda}$ are updated alternately. Note that the sum over all N datapoints in equation(11) means that the VI algorithm requires an entire pass through the

$$p(\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}|\boldsymbol{\Pi}) = p(\boldsymbol{\mathcal{X}}, \mathbf{Z}|\mathbf{U}, \mathbf{V})p(\mathbf{Z}|\boldsymbol{\Pi})p(\mathbf{U})p(\mathbf{V}) = \prod_{n=1}^{N}\prod_{m=1}^{M}\left[\prod_{d=1}^{D}\frac{\Gamma(u_{md}+v_{md})}{\Gamma(u_{md})\Gamma(v_{md})}x_{nd}^{u_{md}-1}(1+x_{nd})^{-u_{md}-v_{md}}\right]^{z_{nm}},$$
$$\times \prod_{n=1}^{N}\prod_{m=1}^{M}\pi_m^{z_{nm}} \times \prod_{m=1}^{M}\prod_{d=1}^{D}\frac{h_{md}^{g_{md}}}{\Gamma(g_{md})}u_{md}{}^{g_{md}-1}e^{-h_{md}u_{md}} \tag{9}$$
$$\times \prod_{m=1}^{M}\prod_{d=1}^{D}\frac{t_{md}^{p_{md}}}{\Gamma(p_{md})}v_{md}{}^{p_{md}-1}e^{-q_{md}v_{md}}.$$

dataset for each update of the global parameters. This becomes prohibitively demanding computationally when the size of the dataset becomes large.

To deal with the limitation, SVI leverages the stochastic optimization (*a.k.a* Robbins-Monro algorithm) to optimize the ELBO via stochastic gradient ascent. Unfortunately, for most of the non-Gaussian mixture models, such as IDMM, GaMM, IBMM, and IBLMM, $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi})$ is unavailable in a closed form, since it involves evaluation of intractable moments $\mathrm{E}_q\left[\ln p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\Pi}, \boldsymbol{\Theta})\right]$. One remedy for this issue is to further bound the lower bound $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi})$.

With a help function $\tilde{p}(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\Pi}, \boldsymbol{\Theta})$ that satisfies

$$\mathrm{E}_q\left[\ln p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\Pi}, \boldsymbol{\Theta})\right] \geq \mathrm{E}_q\left[\ln \tilde{p}(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\Pi}, \boldsymbol{\Theta})\right], \tag{12}$$

and substituting (12) into (11), we can obtain a lower bound of $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi})$ as follows:

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi}) \geq \tilde{\mathcal{L}}(\boldsymbol{\lambda}, \boldsymbol{\phi}) = \sum_{n=1}^{N}\{\mathrm{E}_q[\ln \tilde{p}(\mathbf{x}_n, \mathbf{z}_n|\Pi, \boldsymbol{\Theta})] - \mathrm{E}_q[\ln q(\mathbf{z}_n)]\} - \mathrm{E}_q[\ln q(\boldsymbol{\Theta})]. \tag{13}$$

The reader is referred to [2], [30]–[32] for the detailed strategy about how to choose a proper $\tilde{p}(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\Pi}, \boldsymbol{\Theta})$. If a single observation index s is sampled uniformly $s \sim \mathrm{Unif}(1, \cdots, N)$, the lower bound corresponding to $(\mathbf{x}_s, \mathbf{z}_s)$ as if it was replicated N times is given by

$$\tilde{\mathcal{L}}_s(\boldsymbol{\lambda}, \boldsymbol{\phi}_s) = N\{\mathrm{E}_q[\ln \tilde{p}(\mathbf{x}_s, \mathbf{z}_s|\boldsymbol{\Pi}, \boldsymbol{\Theta})] - \mathrm{E}_q[\ln q(\mathbf{z}_s)\} + \mathrm{E}_q[\ln p(\boldsymbol{\Theta})] - \mathrm{E}_q[\ln q(\boldsymbol{\Theta})]. \tag{14}$$

Since $\mathrm{E}[\tilde{\mathcal{L}}_s(\boldsymbol{\lambda}, \boldsymbol{\phi}_s)] = \tilde{\mathcal{L}}(\boldsymbol{\lambda}, \boldsymbol{\phi})$, the natural gradient of $\tilde{\mathcal{L}}_s(\boldsymbol{\lambda}, \boldsymbol{\phi}_s)$ regarding each global variational parameter $\boldsymbol{\lambda}$ is an unbiased noisy estimator of the natural gradient of $\tilde{\mathcal{L}}(\boldsymbol{\lambda}, \boldsymbol{\phi})$ [23]. This process-sampling one single datapoint and then calculating the natural gradient of $\tilde{\mathcal{L}}_s(\boldsymbol{\lambda}, \boldsymbol{\phi}_s)$-will provide cheaply computed noisy gradients, which enables stochastic optimization to scale to massive datasets.

Even though $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi})$ cannot be optimized directly through the stochastic optimization, its optimum value can still be reached asymptotically by optimizing the lower bound of $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi})$. We refer to this method as the *extended stochastic variational inference* (ESVI). At each iteration t of the ESVI algorithm, we sample a data point $\mathbf{x}_s$ from the dataset $\boldsymbol{\mathcal{X}}$ and then calculate its local variational parameter $\boldsymbol{\phi}_s^*$ via the current estimate of global variational parameters $\boldsymbol{\lambda}^{*(t)}$, which is specified by the local variational distribution as

$$\ln q_s^*(\mathbf{z}_s|\boldsymbol{\phi}_s^*) = \langle \ln \tilde{p}(\mathbf{x}_s, \mathbf{z}_s|\boldsymbol{\Pi}, \boldsymbol{\Theta})\rangle_{\neq \mathbf{z}_s} + \mathrm{Cst}, \tag{15}$$

where the notation $\langle\cdot\rangle_{\neq \mathbf{z}_s}$ denotes an expectation *w.r.t.* the q distributions over all random variables except for variable $\mathbf{z}_s$ and "Cst" denotes the normalization constant. The intermediate global variational parameters $\tilde{\boldsymbol{\lambda}}^{*(t)}$ for the tth iteration based on N replicates of the sampled datapoint $\mathbf{x}_s$ are specified by the intermediate global variational distribution as

$$\ln \tilde{q}^*(\boldsymbol{\Theta}|\tilde{\boldsymbol{\lambda}}^{*(t)}) = N\langle \ln \tilde{p}(\mathbf{x}_s, \mathbf{z}_s|\boldsymbol{\Pi}, \boldsymbol{\Theta})\rangle_{\neq \boldsymbol{\Theta}} + \langle \ln p(\boldsymbol{\Theta})\rangle_{\neq \boldsymbol{\Theta}} + \mathrm{Cst}. \tag{16}$$

The aforementioned global variational parameters for the tth iteration can then be computed as

$$\boldsymbol{\lambda}^{*(t)} = \boldsymbol{\lambda}^{*(t-1)} + \rho_t(\tilde{\boldsymbol{\lambda}}^{*(t)} - \boldsymbol{\lambda}^{*(t-1)}). \tag{17}$$

Here ρ_t is the step size at iteration t. According to [23], we can set the step-size at iteration t as follows

$$\rho_t = (t+\eta)^{-\kappa}. \tag{18}$$

In the above, the $forgetting\ rate\ \kappa \in (0.5, 1]$ controls how fast old information is forgotten and the $delay\ \eta \geq 0$ downweights early iterations. According to [23], the convergence of the ESVI algorithm is theoretically guaranteed if the step size is subject to the following conditions:

$$\sum_t \rho_t = \infty, \sum_t \rho_t^2 < \infty. \tag{19}$$

In the next section, the proposed ESVI framework will be applied to learn the aforementioned IBMM.

3.2 ESVI for the Optimal Posterior Distributions

The expectation of the joint distribution's logarithm is computed as

$$\begin{aligned}\langle \ln p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\Pi}, \boldsymbol{\Theta})\rangle = &\sum_{m=1}^{M}\langle z_{nm}\rangle\left\{\ln \pi_m + \sum_{d=1}^{D}[\mathcal{R}_{md}\right.\\ &+(\langle u_{md}\rangle - 1)\ln x_{nd} - (\langle u_{md}\rangle + \langle v_{md}\rangle)\ln(1+x_{nd})]\}\\ &+\sum_{m=1}^{M}\sum_{d=1}^{D}[(g_{md}-1)\langle \ln u_{md}\rangle - h_{md}\langle u_{md}\rangle]\\ &+\sum_{m=1}^{M}\sum_{d=1}^{D}[(p_{md}-1)\langle \ln v_{md}\rangle - q_{md}\langle v_{md}\rangle] + \mathrm{Cst},\end{aligned} \tag{20}$$

where $\mathcal{R}_{md}$ is defined by

$$\mathcal{R}_{md} = \left\langle \ln \frac{\Gamma(u_{md}+v_{md})}{\Gamma(u_{md})\Gamma(v_{md})}\right\rangle. \tag{21}$$

 This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

It is noteworthy that (20) is not analytically tractable as it involves intractable moment $\mathcal{R}_{md}$. To use (15) and (16) explicitly compute the optimal local and intermediate global posterior distributions, respectively, we need to introduce a help function $\tilde{\mathcal{R}}_m$, which meets the condition $\mathcal{R}_m \geq \tilde{\mathcal{R}}_m$. Following [2], $\tilde{\mathcal{R}}_m$ can be selected as

$$\begin{aligned}\tilde{\mathcal{R}}_{md} = &\ln \frac{\Gamma(\bar{u}_{md}+\bar{v}_{md})}{\Gamma(\bar{u}_{md})\Gamma(\bar{v}_{md})} + [\Psi(\bar{u}_{md}+\bar{v}_{md}) - \Psi(\bar{u}_{md})] \\ &\times (\langle \ln u_{md}\rangle - \ln \bar{u}_{md})\bar{u}_{md} + [\Psi(\bar{u}_{md}+\bar{v}_{md}) - \Psi(\bar{v}_{md})] \\ &\times (\langle \ln v_{md}\rangle - \ln \bar{v}_{md})\bar{v}_{md},\end{aligned} \tag{22}$$

where $\bar{u}_{md} = \langle u_{md}\rangle$, $\bar{v}_{md} = \langle v_{md}\rangle$, $\Psi(\cdot)$ is the Digamma function defined as $\Psi(a) = \frac{\partial \ln \Gamma(a)}{\partial a}$.

Substituting (22) back into (20), we can further lower the exact lower bound $\langle \ln p(\mathbf{x}_n, \mathbf{z}_n|\mathbf{\Pi}, \mathbf{\Theta})\rangle$ as

$$\begin{aligned}\langle \ln \tilde{p}(\mathbf{x}_n, \mathbf{z}_n|\mathbf{\Pi}, \mathbf{\Theta})\rangle = &\sum_{m=1}^{M} \langle z_{nm}\rangle \left\{ \ln \pi_m + \sum_{d=1}^{D} \Big[\tilde{\mathcal{R}}_{md} \right. \\ &\left. + (\langle u_{md}\rangle - 1)\ln x_{nd} - (\langle u_{md}\rangle + \langle v_{md}\rangle)\ln(1+x_{nd})\Big]\right\} \\ &+ \sum_{m=1}^{M}\sum_{d=1}^{D} \left[(g_{md}-1)\langle \ln u_{md}\rangle - h_{md}\langle u_{md}\rangle\right] \\ &+ \sum_{m=1}^{M}\sum_{d=1}^{D} \left[(p_{md}-1)\langle \ln v_{md}\rangle - q_{md}\langle v_{md}\rangle\right] + \text{Cst}.\end{aligned} \tag{23}$$

Then, we sample a datapoint at each iteration to form noisy estimations of the natural gradient of the ELBO. Specifically, at iteration t, we sample a point uniformly at random with $s \sim \text{Unif}(1, \cdots, N)$. Then, with (23) and the aforementioned ESVI framework, we are able to obtain the analytically tractable solutions for the local, intermediate global and global variational distributions. We now consider each of these in more detail:

1) Solution to the local variational distribution

Considering z_{sm} as the variable and including all terms independent of z_{sm} into a constant term, we can rewrite (15) as

$$\ln q^*(z_{sm}) = \sum_{m=1}^{M} z_{sm} \ln \rho_{sm} + \text{Cst}, \tag{24}$$

where

$$\begin{aligned}\ln \rho_{sm} = &\ln \pi_m^{(t-1)} + \sum_{d=1}^{D} \Big[\tilde{\mathcal{R}}_{md}^{(t-1)} + (\bar{u}_{md}^{(t-1)} - 1)\ln x_{sd} \\ &-(\bar{u}_{md}^{(t-1)} + \bar{v}_{md}^{(t-1)})\ln x_{sd}\Big],\end{aligned} \tag{25}$$

where $\bar{u}_{md}^{(t-1)} = \langle u_{md}^{(t-1)}\rangle, \bar{v}_{md}^{(t-1)} = \langle v_{md}^{(t-1)}\rangle$.

By taking exponential of both sides of (24), $q^*(\mathbf{z}_s)$ is recognized to be a categorical density

$$q^*(\mathbf{z}_s|r_{sm}^*) = \prod_{m=1}^{M} (r_{sm}^*)^{z_{sm}}, r_{sm}^* = \frac{\rho_{sm}}{\sum_{m=1}^{M}\rho_{sm}}, \tag{26}$$

where r_{sm}^* denotes the local variational parameter, r_{sm}^* satisfies: $r_{sm} \in \{0,1\}$ and $\sum_{m=1}^{M} r_{sm} = 1$. From (26), we have $\langle z_{sm}\rangle = r_{sm}^*$.

2) Solutions to the intermediate global distributions

Now considering u_{md} as the global variable, we rewrite (16) by including all terms which do not involve u_{md} into a constant term as

$$\ln \tilde{q}^{*(t)}(u_{md}^{(t)}) = (\tilde{g}_{md}^{*(t)} - 1)\ln u_{md}^{(t)} - \tilde{h}_{md}^{*(t)} u_{md}^{(t)} + \text{Cst}, \tag{27}$$

where

$$\tilde{g}_{md}^{*(t)} = g_{md_0} + N r_{sm}\left[\Psi(\bar{u}_{md}^{(t-1)} + \bar{v}_{md}^{(t-1)}) - \Psi(\bar{u}_{md}^{(t-1)})\right]\bar{u}_{md}^{(t-1)}, \tag{28}$$

$$\tilde{h}_{md}^{*(t)} = h_{md_0} - N r_{sm}\left[\ln x_{nd} - \ln(1+x_{sd})\right], \tag{29}$$

where $\tilde{g}_{md}^{*(t)}$ and $\tilde{h}_{md}^{*(t)}$ represent the intermediate global variational parameters.

Taking exponential of both sides of (27), we recognize $\tilde{q}^{*(t)}(u_{md}^{(t)})$ as a Gamma density

$$\tilde{q}^{*(t)}(\mathbf{U}) = \prod_{m=1}^{M}\prod_{d=1}^{D} \mathcal{G}(u_{md}^{(t)}|\tilde{g}_{md}^{*(t)}, \tilde{h}_{md}^{*(t)}). \tag{30}$$

Considering v_{md} as the global variable and absorbing any terms independent of v_{md} into the additive constant, we can rewrite (16) as

$$\ln \tilde{q}^{*(t)}(v_{md}^{(t)}) = (\tilde{p}_{md}^{*(t)} - 1)\ln v_{md}^{(t)} - \tilde{q}_{md}^{*(t)} v_{md}^{(t)} + \text{Cst}, \tag{31}$$

where $p_{md}^{*(t)}$ and $q_{md}^{*(t)}$ are given by

$$\tilde{p}_{md}^{*(t)} = p_{md_0} + N r_{sm}\left[\Psi(\bar{u}_{md}^{(t-1)} + \bar{v}_{md}^{(t-1)}) - \Psi(\bar{v}_{md}^{(t-1)})\right]\bar{v}_{md}^{(t-1)}, \tag{32}$$

$$q_{md}^{*(t)} = q_{md_0} - N\ln(1+x_{sd}), \tag{33}$$

Taking exponential of both sides of (31), we recognize $\tilde{q}^{*(t)}(v_{md}^{(t)})$ as a Gamma density

$$\tilde{q}^{*(t)}(\mathbf{V}) = \prod_{m=1}^{M}\prod_{d=1}^{D} \mathcal{G}(v_{md}^{(t)}|\tilde{p}_{md}^{*(t)}, \tilde{q}_{md}^{*(t)}). \tag{34}$$

3) Solutions to the global variational distributions

With the above obtained intermediate global variational distributions and by applying (17), we can calculate the global variational distributions as follows:

$$q^{*(t)}(\mathbf{U}) = \prod_{m=1}^{M}\prod_{d=1}^{D} \mathcal{G}(u_{md}^{(t)}|g_{md}^{*(t)}, h_{md}^{*(t)}), \tag{35}$$

$$q^{*(t)}(\mathbf{V}) = \prod_{m=1}^{M}\prod_{d=1}^{D} \mathcal{G}(v_{md}^{(t)}|p_{md}^{*(t)}, q_{md}^{*(t)}), \tag{36}$$

where the global variational parameters are given by

$$g_{md}^{*(t)} = g_{md}^{*(t-1)} + \rho_t(\tilde{g}_{md}^{*(t)} - g_{md}^{*(t-1)}), \tag{37}$$

$$h_{md}^{*(t)} = h_{md}^{*(t-1)} + \rho_t(\tilde{h}_{md}^{*(t)} - h_{md}^{*(t-1)}), \tag{38}$$

$$p_{md}^{*(t)} = p_{md}^{*(t-1)} + \rho_t(\tilde{p}_{md}^{*(t)} - p_{md}^{*(t-1)}), \tag{39}$$

$$q_{md}^{*(t)} = q_{md}^{*(t-1)} + \rho_t(\tilde{q}_{md}^{*(t)} - q_{md}^{*(t-1)}), \tag{40}$$

Finally, the mixing coefficients $\mathbf{\Pi}$ can be updated as follows:

$$\pi_m^{(t)} = (1-\rho_t)\pi_m^{(t-1)} + N\rho_t r_{sm}. \tag{41}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

The expected values in all the above formulas are given as follows:

$$\bar{u}_{md}^{(t)} = g_{md}^{*(t)}/h_{md}^{*(t)}, \langle \ln u_{md}^{(t)} \rangle = \Psi(g_{md}^{*(t)}) - \ln(h_{md}^{*(t)}), \quad (42)$$

$$\bar{v}_{md}^{(t)} = p_{md}^{*(t)}/q_{md}^{*(t)}, \langle \ln v_{md}^{(t)} \rangle = \Psi(p_{md}^{*(t)}) - \ln(q_{md}^{*(t)}). \quad (43)$$

The ESVI of the IBMM is summarized in Algorithm 1.

Algorithm 1 Algorithm for ESVI-based Bayesian IBMM

1: Set the initial number of components M and the initial values for hyperparameters g_{md_0}, h_{md_0}, p_{md_0}, and q_{md_0}.
2: Initiate the values of r_{sm} by K-means algorithm.
3: Set the step-size ρ_t appropriately using (19).
4: **while** TRUE **do**
5: Sample a data point $\mathbf{x}_n$ uniformly from the dataset: $s \sim$ Unif(1,...,N).
6: Optimize the local variational parameter r_{sm}^* using (26).
7: Calculate intermediate global parameters as though $\mathbf{x}_s$ is replicated N times using (28), (29), (32), and (33).
8: Update the current estimate of the global variational parameters using (37)-(40).
9: Update the current solutions for $\mathbf{\Pi}$ using (41).
10: **end while**

4 EXPERIMENTS AND RESULTS

4.1 Experimental Setup

This section presents numerous experimental results to assess the effectiveness and performance of the proposed approach on real positive vectors, generated from three real-life challenging applications namely text categorization, network traffic classification and misuse intrusion detection. The first purpose is to investigate how the mini-batch size S influences the algorithms. The second purpose is to compare ESVI to the traditional batch EVI algorithm. The third purpose is to assess the approaches of these applications by considering the comparable mixture-based approaches. Note that the evaluation of diverse methods is not our main concern and is out of the scope of this paper.

In the initialization stage of all our experiments, the number of components M (with equal mixing weights) is set as $M = 10$. The initial values of hyperparameters g_{md_0}, p_{md_0} of the Gamma priors are set to 1, and h_{md_0}, q_{md_0} are set to 0.5. The parameters η and κ of the learning rate are set to 32 and 0.6, respectively. Moreover, the initial settings for the baseline models in this paper are same as that in their original papers. It is noteworthy that these specific choices were based on our experiments and were found convenient and effective in our case. When the proposed algorithm stops, the posterior means are adopted as parameter estimates in the IBMM. To make a fair comparison, all the simulation experiments are conducted using MATLAB (2018a) on a computer with Inter(R) Core(TM) i7-75600U CPU 2.40 GHz, 16 GB of RAM.

4.2 Performance Metrics

In order to compare classification effectiveness of each approach, we adopt three widely applied performance measures namely *precision*, *recall*, and *F1* score [33] defined as follows:

$$Precision = \frac{\sum_{m=1}^{M} TP_m}{\sum_{m=1}^{M} (TP_m + FP_m)}, \quad (44)$$

$$Recall = \frac{\sum_{m=1}^{M} TP_m}{\sum_{m=1}^{M} (TP_m + FN_m)}, \quad (45)$$

$$F_1 - score = \frac{2Precision \cdot Recall}{Precision + Recall}, \quad (46)$$

where TP_m is the number of true positives, TN_m is the number of true negatives, FP_m is the number of false positives, and FN_m is the number of false negatives. In order to remove the randomness influence upon the results, we conducted **100** rounds of simulations for each experiment and compute the average *precision*, *recall*, and *F1 score* across all classes, based on macro and micro averaging [33], [34]. It is noteworthy mentioning that the macro-average and micro-average scores of these metrics are equal for the balanced datasets. For ease of the representation, we denote macro-averaged recall, precision, and F1 as well as micro-averaged recall, precision, and F1 as *MacroR*, *MacroP*, *MacroF1*, *MicroR*, *MicroP*, and *MicroF1*, respectively. Moreover, the training time for building the classifier is also adopted to measure the efficiency of diverse methods.

4.3 Text Categorization

The amount of text documents in electronic form has grown exponentially over the past few decades, due to the fast growth of the internet and intranets. When carrying out efficient analysis for text in a manual way, it will cost huge manpower to organize and process the text documents. Hence, it is very important to develop efficient techniques to label the textual content with one or more predefined categories automatically, what is known as automatic Text Categorization (TC). A number of machine learning algorithms have been proposed to deal with this challenging task by formulating it as a classification problem [35]–[38]. Among these methods, Bayesian statistical model-based approaches have gained considerable attention in recent years [2], [36], [39], [40].

We report on the results of employing the proposed Bayesian IBMM with ESVI (ESVI-IBMM) to TC upon two publicly available datasets, that have been widely applied in literature for performance evaluation. The first dataset is the WebKB dataset [41], that contains seven categories of Web pages and 8,282 documents in total. However, following the previous works in literature [42] on this dataset, we applied only four categories in our experiments: course (930 documents), faculty (1124 documents), project (504 documents), and student (1641 documents), with a total of 4199 documents; therefore, it is an imbalanced dataset. The second dataset is the 20 Newsgroups (20NG) dataset [43], that is composed of 13,998 documents taken from the Usenet newsgroups. These documents are evenly distributed over 20 classes with 700 articles in each class; therefore, it is a balanced dataset. Each

 This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

categorization of these two datasets was randomly split into two equal parts, one part for training and the other part for testing. Following the previous works in [44], we apply the Porter's Stemming algorithm [45] to reduce the words to their basic forms. Moreover, we remove from documents all words that occur less than 3 times or is shorter than 2 in length, such that each document in the dataset is then represented as a positive feature vector.

In our case, the categorization process is based on two steps involving the training and testing. The training step allows the representation of each class as a statistical model. The testing stage assigns each testing document to the predefined category with the well-known Bayes classification rule. We first investigate how the mini-batch size influences the categorization performance, when ESVI-IBMM, Bayesian IBLMM with ESVI (ESVI-IBLMM), Bayesian GaMM with ESVI (ESVI-GaMM), and Bayesian IDMM with ESVI (ESVI-IDMM) are applied as classifiers. We varied the numbers of minibatch sizes S to be 1, 5, 10, 20, 30, $\cdots$, 100. Figs. 2 and 3 show the mean results of the proposed method in terms of the MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1 (percent) based upon 100 runs, for the WebKB and 20NG datasets, respectively, as a function of the number of minibatch sizes. From these figures, it can be seen that too small batch sizes (*e.g.*, S=1, 5, 10) can influence performance and large batch sizes are preferred. However, the difference in performance is very small once the batch size is set high enough (*e.g.*, S=50, 60).

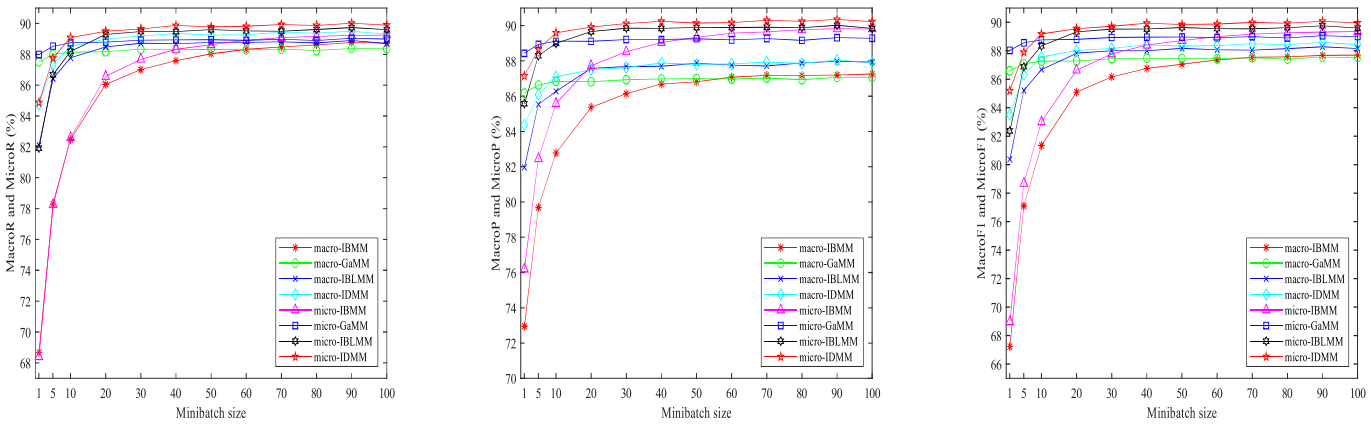
Next, we compare the ESVI framework with the batch EVI framework. To do so, we compare the performance of four ESVI-based methods involving the ESVI-IBMM, ESVI-IBLMM, ESVI-GaMM, and ESVI-IDMM with the performance of other four EVI-based methods involving the EVI-GaMM [6], Bayesian IBLMM with EVI (EVI-IBLMM) [46], Bayesian IBMM with EVI (EVI-IBMM) [2], and Bayesian IDMM with EVI (EVI-IDMM) [6] on the TC task. We fixed the batch size to 60 for both the WebKB and 20NG datasets. Table 1 gives the mean results of the tested methods in terms of computational time as well as MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1 based upon 100 runs. As shown in the table, the algorithms under the ESVI framework are able to achieve comparable performance in terms of MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1, compared to the algorithms under the EVI framework. However, the ESVI-based algorithms are much more computationally efficient with much less computational time. This fact demonstrates the superiorities of applying ESVI over EVI when tackling massive non-Gaussian datasets. Another noticeable observation from Table 1 is that IBLMM and IDMM perform little better than IBMM and GaMM under both the EVI and ESVI frameworks, which can be explained by the fact that the text features obtained above are dependent and both IBLMM and IDMM are more proper than IBMM and GaMM to fit the underlying distributions of the relevant positive feature vectors. The categorization performance distributions on the WebKB and 20NG datasets are shown in Figs. 4 and 5, respectively.

4.4 Network Traffic Classification

Network traffic classification has an extensive variety of applications in network management and security, such as security monitoring, QoS control, and intrusion detection [47], [48]. With the exponential growth of network users and the emergence of new network services, network traffic classification has attracted considerable attention from both industry and academia over the past few decades [49]–[51]. A great number of methods have been proposed and applied to carry out traffic classification tasks. Among these methods, machine learning methods based upon flow statistical features have been the most popular class of methods. Much of their popularity is due to the fact that they can be achieved via applying supervised or unsupervised classification algorithms [51]–[53].

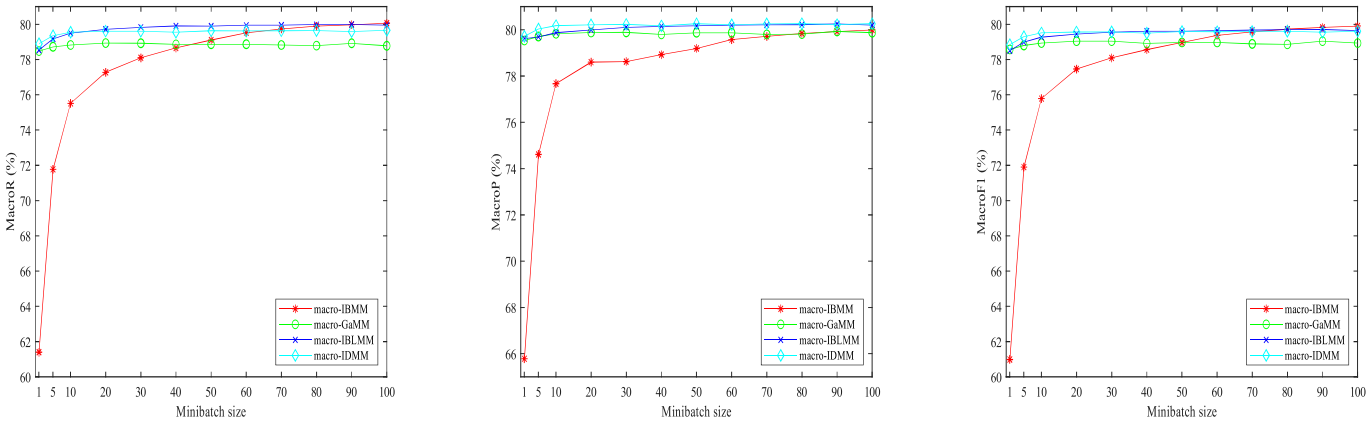
Here, we report the experimental results by applying the proposed ESVI-IBMM as a classifier for the task of network traffic classification upon the publicly available UNIBS Anonymized 2009 Internet Traces [54], that contains 9209 flows distributed imbalance over 4 categories: Web (6173 flows), Mail (653 flows), BitTorrent (215 flows), and EMule (1628 flows). Each categorization is divided into two equal parts, one part for training and the other part for testing. Following the work in [55], four features are extracted from these flows, containing the size of the first packet payload sent from client to server, size of the first packet payload sent from server to client, size of the second packet payload sent from server to client and the port of each packet. All the features are integer values. Note that, since these four features are on quite different scales; therefore, we need to normalize each of these features into the range of [0,1] such that one feature would not dominate the others.

Similar to the TC task, firstly, the influence of minibatch size on classification performance is explored. Fig. 6 shows the mean values of the MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1 on the UNIBS dataset based on 100 runs, as a function of minibatch size. As we can observe from the figure, the classification performance improves with gradually increased minibatch size number and hardly improves with minibatch size increasing once the minibatch size number is large enough. We set the minibatch size $S = 60$. We report the average results and the standard deviations of the tested methods in terms of number of the MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1, and computational time in Table 2 over 100 trials. As shown in the table, the algorithms under the prosed ESVI framework give consistently comparable performance and have much less computational cost, compared to the algorithms under the EVI framework. It also can be seen that, both IBMM and GaMM outperform the IBLMM and IDMM, which can be explained again by the fact that the variables in the feature vectors extracted from the UNIBS dataset are mutually independent and IBMM and GaMM are more appropriate to model the independent positive feature vectors, compared to both IBLMM and IDMM. The classification performance distributions are shown in Fig. 7.



(a) (b) (c)

Fig. 2: Categorization performance of different ESVI-based algorithms versus minibatch size for the WebKB dataset. (a) MacroR and MicroR. (b) MacroP and MicroP. (c) MacroF1 and MicroF1.



(a) (b) (c)

Fig. 3: Categorization performance of different ESVI-based algorithms versus minibatch size for the 20NG dataset. (a) MacroR. (b) MacroP. (c) MacroF1.

TABLE 1: The mean categorization results and the standard deviations on the WebKB and 20NG datasets in terms of MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1 (in %) and computational time (in s) over 100 runs

Dataset	Method	ESVI-IBMM	EVI-IBMM	ESVI-GaMM	EVI-GaMM	ESVI-IBLMM	EVI-IBLMM	ESVI-IDMM	EVI-IDMM
Webkb	MacroR	88.33±0.98	89.21±0.52	88.30±0.65	88.68±0.65	88.78±0.78	88.96±0.59	89.27±0.63	89.31±0.55
	MacroP	87.08±1.12	87.65±0.59	86.97±0.65	87.07±0.62	87.76±1.03	89.07±0.53	87.83±0.69	87.89±0.59
	MacroF1	87.35±1.12	88.26±0.53	87.46±0.63	87.68±0.60	88.07±0.91	88.98±0.52	88.36±0.66	88.43±0.56
	MicroR	88.91±0.96	89.73±0.47	88.91±0.59	89.20±0.53	89.52±0.82	90.29±0.42	89.82±0.56	89.87±0.50
	MicroP	89.57±0.69	90.08±0.44	89.19±0.58	89.57±0.50	89.89±0.63	90.28±0.43	90.17±0.49	90.19±0.47
	MicroF1	89.00±0.91	89.79±0.47	88.92±0.58	89.27±0.52	89.55±0.80	90.26±0.43	89.88±0.55	89.92±0.50
	Runtime	0.17±0.02	0.50±0.03	0.11±0.02	0.35±0.04	0.15±0.02	0.49±0.02	0.13±0.02	0.47±0.02
20NG	MacroR	79.52±0.42	80.94±0.36	78.87±0.45	79.00±0.39	79.94±0.37	81.12±0.33	79.63±0.36	80.22±0.41
	MacroP	79.57±0.44	80.86±0.37	79.87±0.36	79.10±0.39	80.20±0.38	81.15±0.32	80.22±0.33	80.20±0.39
	MacroF1	79.37±0.43	80.81±0.37	78.96±0.52	79.00±0.39	79.64±0.39	80.83±0.33	79.58±0.36	80.02±0.41
	Runtime	2.42±0.18	5.05±0.10	1.05±0.06	2.15±0.12	1.66±0.11	3.96±0.09	1.51±0.06	3.07±0.06

4.5 Misuse Intrusion Detection

With the explosive growth of network-based services and sensitive information upon networks, network security issues have become more and more prominent. Intrusion detection is one of the most important technologies to ensure network security, whose purpose is to automate the process of detecting when intrusions are occurring in a network [56]–[59]. Current methods detecting intrusions can be grouped into two main classes: misuse and anomaly detection [60], [61]. The goal of misuse detection is to identify known attacks based upon pre-defined attack patterns and signatures. Anomaly detection discovers unknown attacks based upon deviations from normal activities. Misuse detection algorithms have gained a lot

 This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

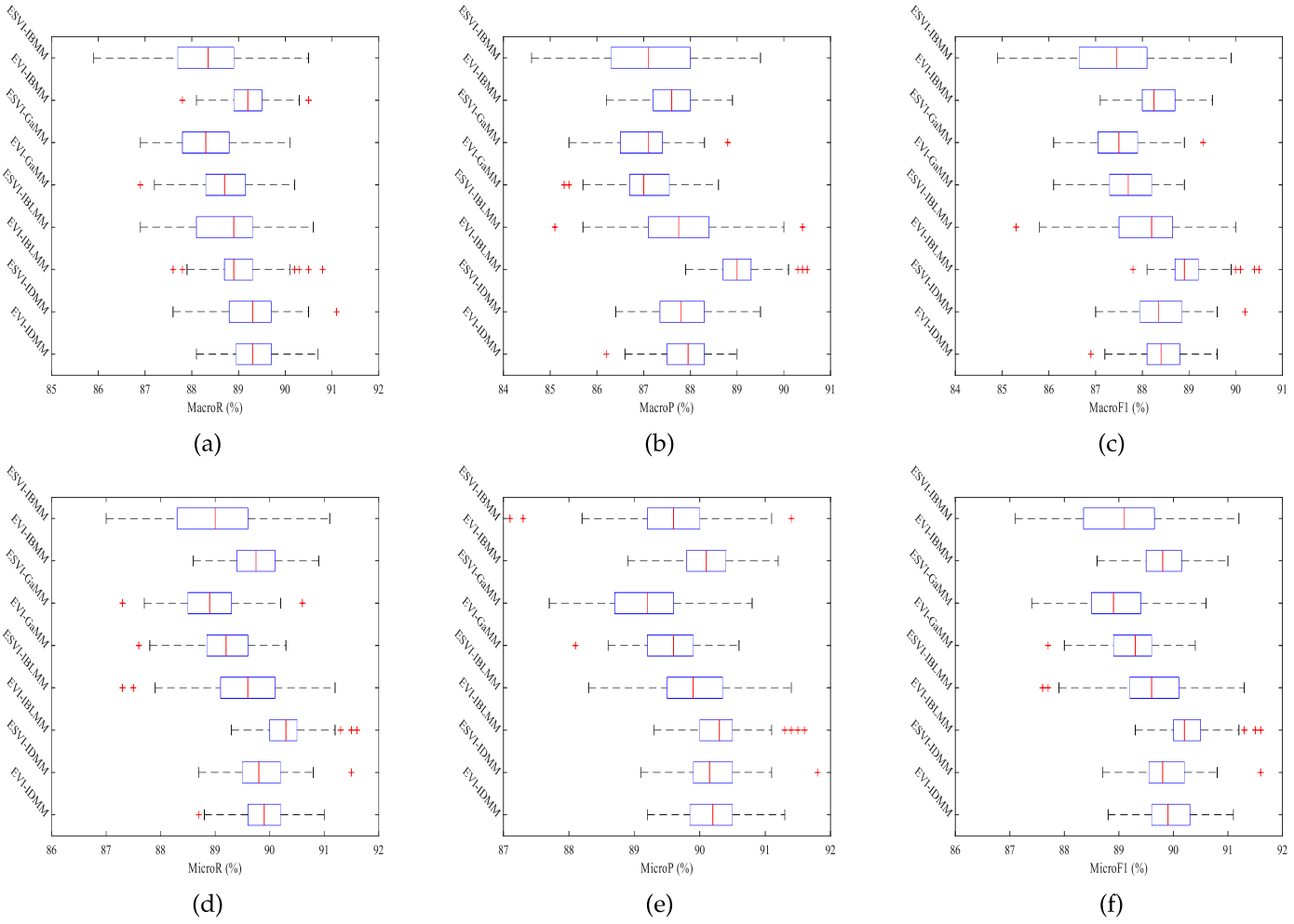


Fig. 4: Boxplots for comparisons of the TC performance's distributions for the WebKB dataset. The central mark is the median; the edges of the box are the 25th and 75th percentiles. The outliers are marked individually. (a) MacroR. (b) MacroP. (c) MacroF1. (d) MicroR. (e) MicroP. (f) MicroF1.

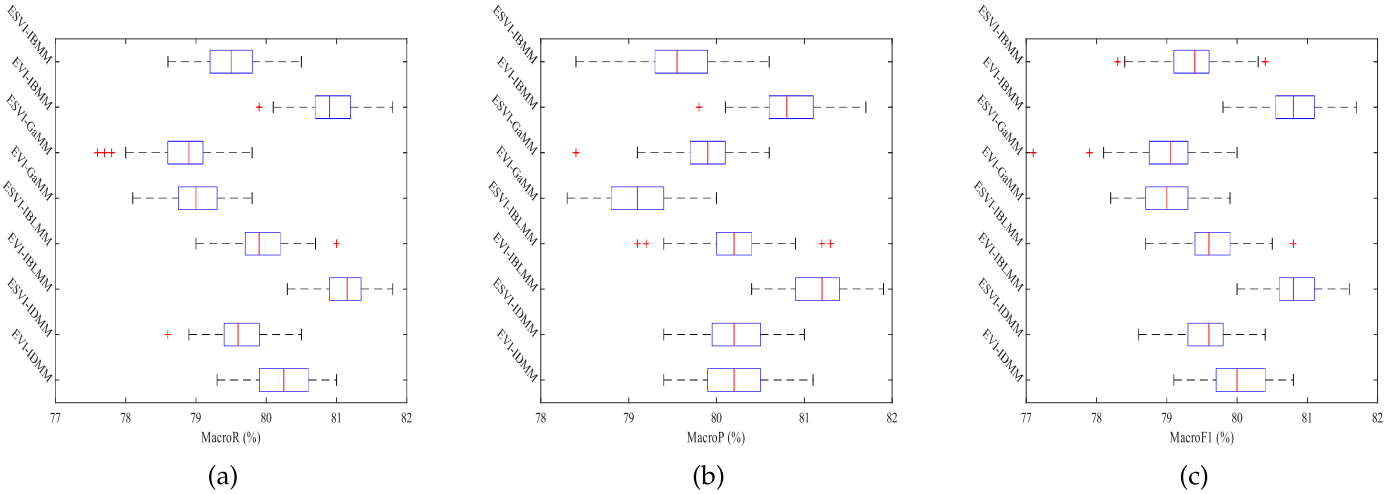
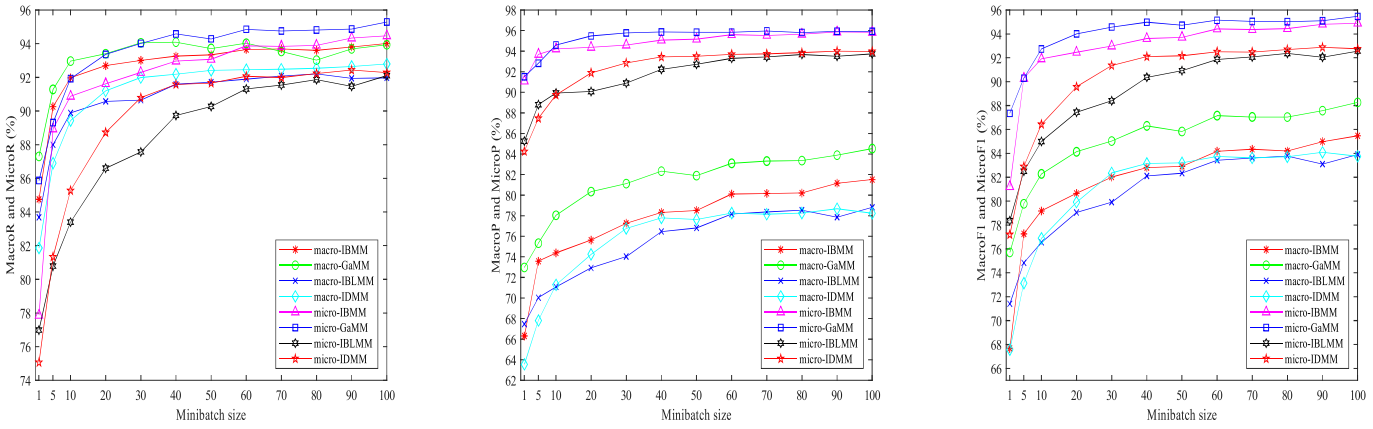


Fig. 5: Boxplots for comparisons of the TC performance's distributions for the 20NG dataset. The central mark is the median; the edges of the box are the 25th and 75th percentiles. The outliers are marked individually. (a) MacroR. (b) MacroP. (c) MacroF1.

of attention over the past few decades since they have the advantage that they are fast and have a low false positive rate. In this work, the proposed ESVI-IBMM is applied as a classifier to investigate its performance in misuse intrusion

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems



(a) (b) (c)

Fig. 6: Categorization performance of different ESVI-based algorithms versus minibatch size for the UNIBS dataset. (a) MacroR and MicroR. (b) MacroP and MicroP. (c) MacroF1 and MicroF1.

TABLE 2: The average classification results and the standard deviations on the UNIBS dataset in terms of MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1 (in %) and computational time (in s) over 100 runs

Method	ESVI-IBMM	EVI-IBMM	ESVI-GaMM	EVI-GaMM	ESVI-IBLMM	EVI-IBLMM	ESVI-IDMM	EVI-IDMM
MacroR	93.66±0.69	94.43±0.87	94.03±3.47	95.12±0.64	91.90±1.13	92.35±0.68	92.45±0.91	93.66±0.48
MacroP	80.11±2.57	83.32±1.84	83.10±4.17	85.46±1.74	78.17±2.19	78.95±1.76	78.26±1.86	78.82±1.57
MacroF1	84.18±2.16	86.89±1.69	87.16±3.90	88.98±1.32	83.42±1.83	83.99±1.58	83.75±1.60	84.44±1.28
MicroR	93.87±1.23	95.05±0.70	94.86±1.35	95.86±0.58	91.31±1.63	91.55±1.22	92.06±1.07	92.25±0.93
MicroP	95.61±0.69	96.18±0.45	95.85±0.88	96.56±0.38	93.31±0.94	93.24±0.52	93.69±0.71	94.01±0.45
MicroF1	94.42±1.07	95.39±0.60	95.15±1.22	96.06±0.52	91.87±1.43	92.01±1.00	92.51±0.97	92.73±0.79
Runtime	0.17±0.02	0.91±0.03	0.12±0.02	0.54±0.11	0.19±0.03	0.99±0.05	0.15±0.02	0.82±0.08

detection.

We employ a commonly-used public dataset, namely the well-known KDD Cup 1999 dataset [1], to evaluate the effectiveness of the proposed method. Each record in this dataset denotes a network connection with 41 features in which 34 are numeric and 7 are symbolic. These records were obtained from the simulated intrusions and can be classified as normal or one of four attack categories: DoS (denial-of-service), R2L (remote-to-local), U2R (user-to-root), and Prb (probing). Each class in this dataset was randomly divided into two separate halves, one for training and the other for test during evaluations. In the feature extraction phase, following the work in [62], we keep 493,965 records containing four categories: Normal (972,78 records), DoS (391,454 records), Prb (4107 records), and R2L (1126 records) to design misuse intrusion detection analysis. Additionally, the symbolic features such as 'protocol', 'TCP Status flag' and 'service type', are mapped into binary numeric features, each record is represented as a 52-D positive feature vector. Then, an ESVI-IBMM was trained for each category. Finally, a testing record is categorized to the corresponding category that yields the highest likelihood score.

Similar to the aforementioned two tasks, we firstly want to explore the influence of minibatch sizes upon the detection performance of the ESVI-IBMM classifier in terms of MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1. The detection performances, for the KDD CUP 1999 dataset, when employing ESVI-IBMM, as a function of the number of minibatch sizes, are shown in Fig. 8. As shown in this figure, too small minibatch sizes influence the detection performance that will be improved when more records are subsampled in the training dataset. Nevertheless, the detection performances of these approaches keep almost unchangeable while the minibatch size is set high enough. We fix the batch size to 90 records. Table 3 lists the average values and the standard deviations of the MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1 as well as running time of the KDD CUP 1999 dataset, obtained by different approaches based on 100 runs. For MacroP, MacroR, and MacroF1, the ESVI-based algorithms perform a little worse than the EVI-based algorithms. But for MicroP, MicroR, and MicroF1, the ESVI-based algorithms achieve almost identical performance, compared to the EVI-based algorithms. It can be also seen that the ESVI-based algorithms run much faster than the EVI-based algorithms, that further verifies the superiorities of employing ESVI over EVI when addressing the problem of massive non-Gaussian data modeling. Moreover, it can be seen that IBMM and GaMM perform better than IBLMM and IDMM under the corresponding EVI and ESVI frameworks, that can be explained by the fact that the extracted features from the KDD CUP 1999 are independent. Moreover, IBMM and GaMM are more efficient than IBLMM and IDMM in fitting the underlying distributions of the irrelevant positive feature vectors. The detection performance distributions are shown in Fig. 9.

1. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

 This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

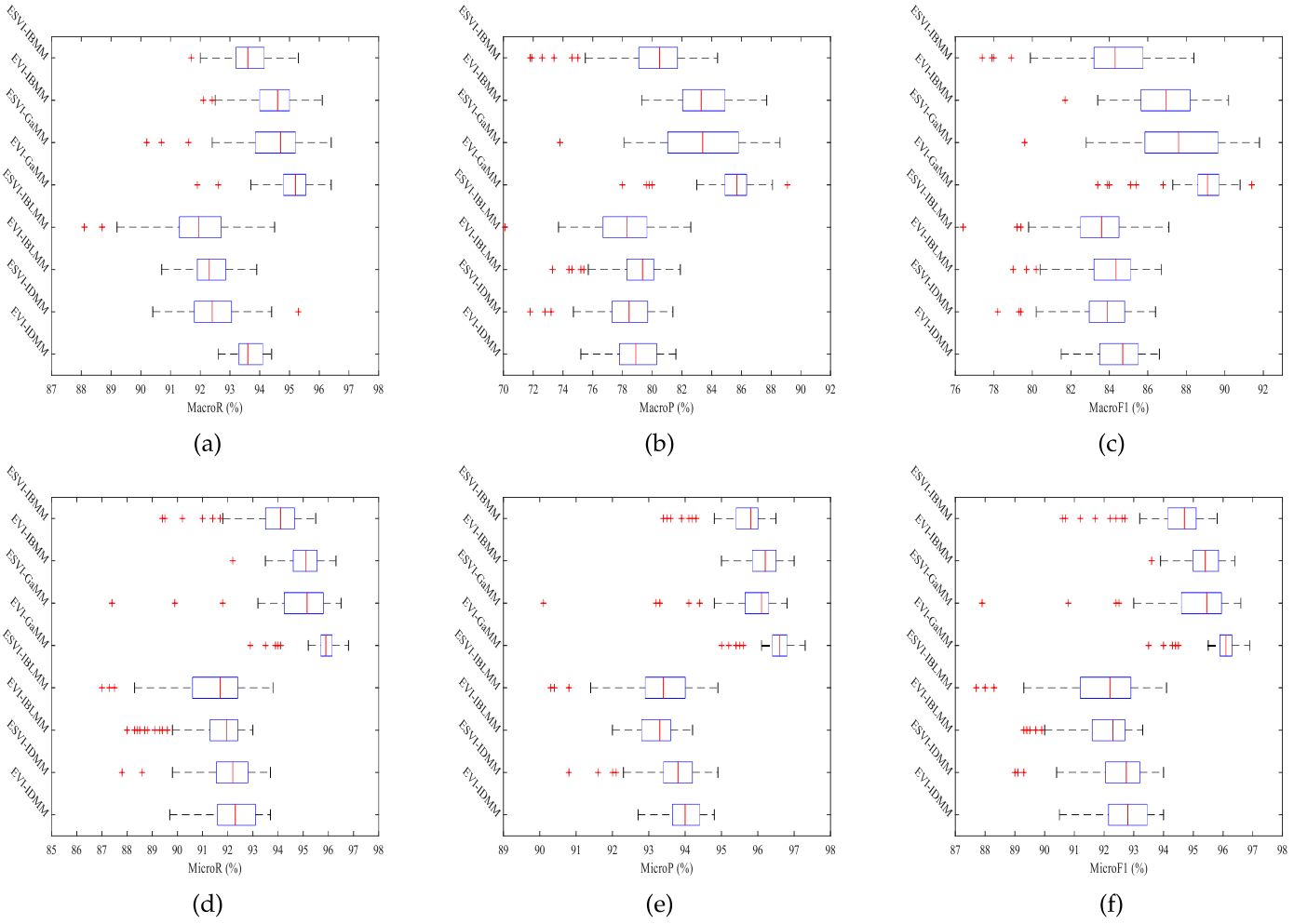


Fig. 7: Boxplots for comparisons of the image classification accuracies distributions for the UNIBS dataset. The central mark is the median; the edges of the box are the 25th and 75th percentiles. The outliers are marked individually. (a) MacroR. (b) MacroP. (c) MacroF1. (d) MicroR. (e) MicroP. (f) MicroF1.

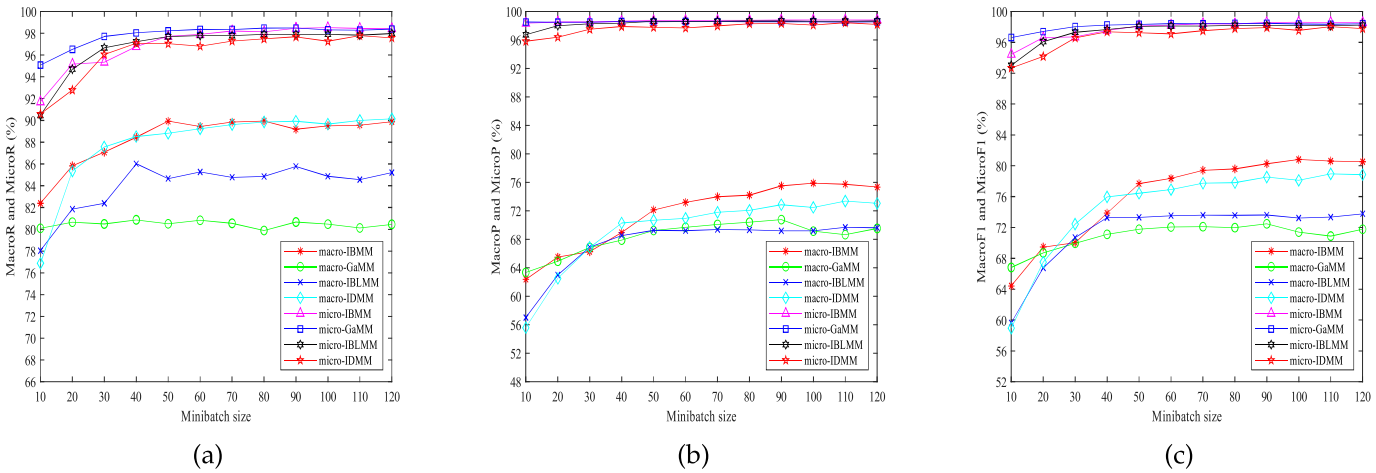


Fig. 8: Detection performance of different ESVI-based algorithms versus minibatch size for the KDD CUP 1999 dataset. (a) MacroR and MicroR. (b) MacroP and MicroP. (c) MacroF1 and MicroF1.

4.6 Comparison with Deep Learning-Based Methods

To explore more insights for the ESVI framework, the ESVI-IBMM algorithm is further compared to deep neural networks (DNNs) [63] on the tasks of both the text categorization and network traffic classification. We apply the fully connected (FC) neural networks with diverse

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

TABLE 3: The average detection results and the standard deviations on the KDD CUP 1999 dataset in terms of MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1 (in %) and computational time (in s) over 100 trails

Method	ESVI-IBMM	EVI-IBMM	ESVI-GaMM	EVI-GaMM	ESVI-IBLMM	EVI-IBLMM	ESVI-IDMM	EVI-IDMM
MacroR	89.19±3.31	85.97±2.68	80.67±2.74	84.32±2.53	85.77±5.13	88.58±2.16	89.92±1.22	89.24±1.24
MacroP	75.51±2.32	82.13±2.70	70.77±3.68	75.50±4.06	69.19±4.12	70.57±4.80	72.86±4.49	69.64±5.07
MacroF1	80.26±2.10	83.25±2.36	72.47±3.07	78.66±3.00	73.61±4.33	75.78±5.46	78.54±4.35	75.08±5.71
MicroR	98.45±0.25	98.94±0.28	98.48±0.27	98.78±0.14	97.90±0.85	97.16±2.05	97.68±2.01	96.43±2.32
MicroP	98.79±0.06	99.06±0.10	98.57±0.19	98.98±0.09	98.66±0.11	98.44±0.60	98.33±0.99	98.06±0.17
MicroF1	98.58±0.17	98.98±0.21	98.47±0.22	98.86±0.10	98.21±0.52	97.70±1.32	97.92±1.58	97.12±1.34
Runtime	12.59±0.91	227.96±18.19	7.85±0.93	59.56±12.99	16.46±2.01	167.76±6.45	11.06±0.84	137.07±1.45

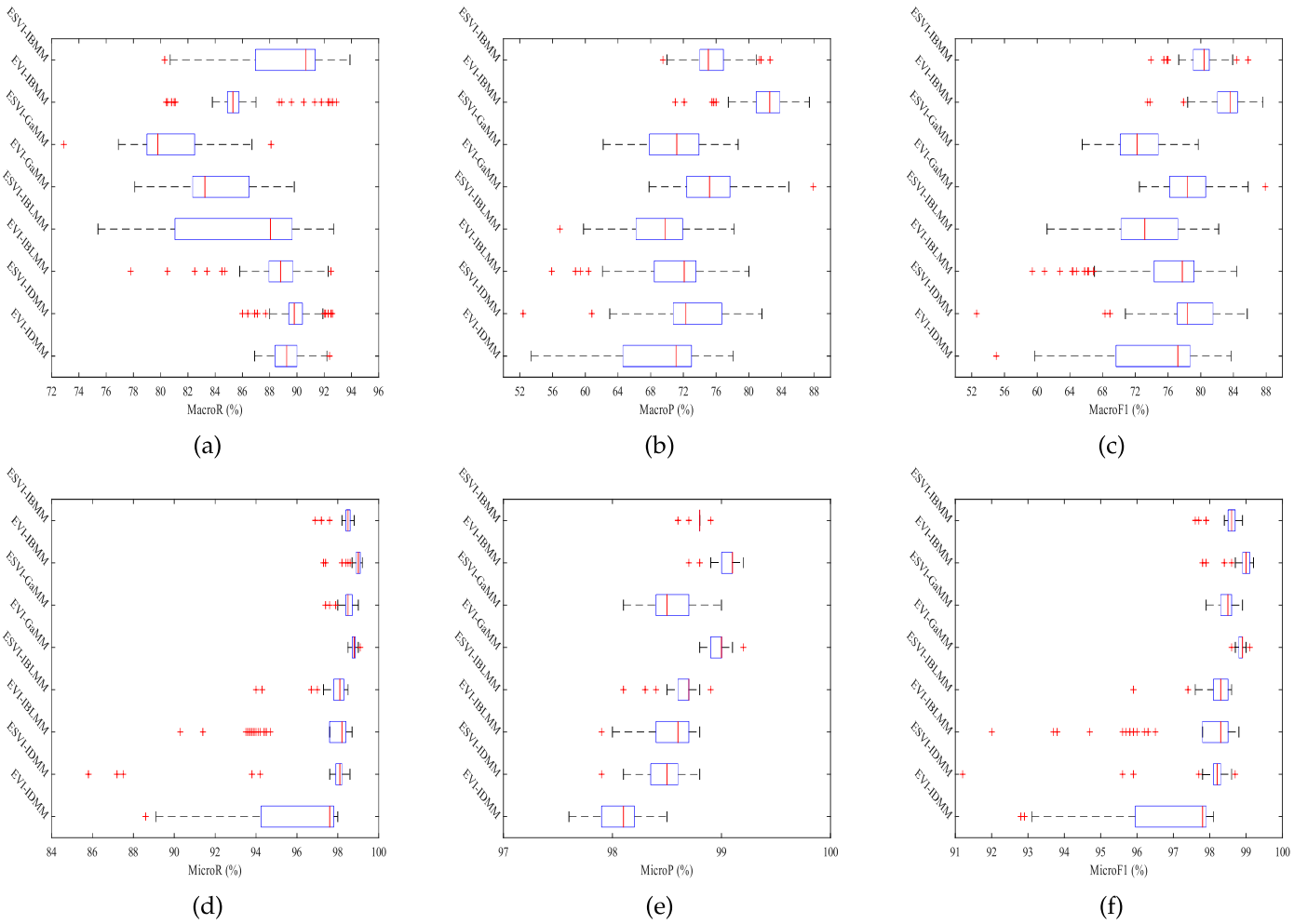


Fig. 9: Boxplots for comparisons of the intrusion detection performance's distributions for the KDD CUP 1999 dataset. The central mark is the median; the edges of the box are the 25th and 75th percentiles. The outliers are marked individually. (a) MacroR. (b) MacroP. (c) MacroF1. (d) MicroR. (e) MicroP. (f) MicroF1.

numbers (*i.e.*, l) of hidden layers. The previously extracted 4-D features for the Webkb dataset, 20-D features for the 20NG dataset, 4-D features for the UNIBS dataset, and 52-D features for the KDD CUP 1999 dataset are applied as inputs. The l is set as 1, 2, and 4, respectively, and the number of nodes in each hidden layer is the same as the dimension of the features for these four datasets. The minibatch size S is set to 32.

The average values and the standard deviations of the MacroP, MacroR, MacroF1, MicroP, MicroR, and MicroF1 of the different datasets got by diverse methods over 100 runs are shown in Table 4. On the WebKB dataset, our presented method can provide comparably good or better performance, compared to the shallower FC neural networks (*i.e.*, l=1, 2), and performs better than the deeper FC neural networks (*i.e.*, l=4) with large gaps. On the 20NG dataset, our presented approach performs a little bit worse than different FC neural networks. On the UNIBS dataset, our presented method performs better than different FC neural networks with large gaps. On the KDD CUP 1999 dataset, for MacroP, MacroR, and MacroF1, our presented method performs worse than the shallower FC neural networks (*i.e.*, l=1, 2) with large gaps; however, for MicroP, MicroR, and MicroF1, our presented approach performs better than the shallower FC neural network (*i.e.*, l=1, 2) with large gaps. On the KDD CUP

 This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

1999 dataset, our presented approach performs better than the deeper FC neural network (*i.e.*, l=4) with large gaps. Another noticeable observation from Table 4 is that, on the unbalanced datasets (*i.e.*, WebKB, UNIBS, and KDD CUP 1999), as the number of hidden layer of the FC neural networks increases, the performance of the FC neural networks become worse; however, for the balanced dataset (*i.e.*, 20NG), the layer of the FC neural networks has no affect on their performance.

TABLE 4: Comparison of performance (in %) and computational time (in s) between the FC Neural Networks and the proposed ESVI-IBMM model on the tasks of TC, network traffic classification, and misuse intrusion detection. Notice that l represents the number of hidden layer of the FC neural networks.

Dataset	Method	FC ($l=1$)	FC ($l=2$)	FC ($l=4$)	ESVI-IBMM
Webkb	MacroR	88.53±1.82	87.78±1.53	73.63±12.75	88.33±0.98
	MacroP	86.45±2.61	85.93±2.13	75.53±8.75	87.08±1.12
	MacroF1	87.29±2.41	86.67±1.93	74.08±10.81	87.35±1.12
	MicroR	89.15±1.60	88.71±1.14	79.70±8.40	88.91±0.96
	MicroP	89.25±1.54	88.83±1.03	83.12±4.93	89.57±0.69
	MicroF1	89.08±1.69	88.67±1.16	81.00±6.82	89.00±0.91
20NG	MacroR	81.34±0.37	81.31±0.36	81.09±0.34	79.52±0.42
	MacroP	81.33±0.36	81.27±0.37	81.06±0.35	79.57±0.44
	MacroF1	81.28±0.36	81.23±0.37	81.01±0.35	79.37±0.43
UNIBS	MacroR	69.65±0.63	65.46±13.95	69.94±19.96	93.66±3.69
	MacroP	70.62±2.80	68.17±12.76	64.55±18.50	80.11±2.57
	MacroF1	69.64±2.14	66.51±13.41	62.44±19.32	84.18±2.16
	MicroR	93.23±0.78	90.32±10.98	86.62±15.22	93.87±1.23
	MicroP	94.81±0.80	93.35±6.04	91.28±8.36	95.61±0.69
	MicroF1	93.78±0.95	91.54±8.87	88.54±12.29	94.42±1.07
KDD	MacroR	92.65±0.29	90.43±10.74	84.98±13.61	89.19±3.31
	MacroP	80.33±0.71	79.11±9.59	76.99±11.43	75.51±2.32
	MacroF1	84.72±0.43	83.59±10.20	79.97±12.40	80.26±2.10
	MicroR	94.53±0.17	94.79±0.41	94.91±0.48	98.45±0.25
	MicroP	94.47±0.08	94.69±0.25	94.78±0.26	98.79±0.06
	MicroF1	94.17±0.11	94.39±0.29	94.46±0.34	98.58±0.17

TABLE 5: Comparison of the average computational time (in s) of different methods based on 100 runs. Note that l represents the number of hidden layer of the FC neural networks.

Dataset & Method	FC ($l=1$)	FC ($l=2$)	FC ($l=4$)	ESVI-IBMM
Webkb	4.39±0.56	5.04±0.09	6.47±0.08	0.17±0.02
20NG	15.42±0.82	18.57±1.35	23.79±1.05	2.42±0.18
UNIBS	9.40±0.07	11.08±0.31	13.98±0.21	0.17±0.02
KDD	554.06±1.62	703.53±9.69	896.13±18.45	12.59±0.91

5 Conclusions

In this paper, we have proposed a fairly efficient ESVI framework to perform Bayesian inference of IBMM with closed-form solutions. It is worth mentioning that this framework is particularly scalable and suitable to attain an analytically tractable solution for Bayesian estimation of non-Gaussian statistical models for massive datasets. The effectiveness of the proposed method is verified through extensive realistic data evaluations. The research outcomes show that the proposed approach can not only achieve comparable performance for three real-world applications namely text categorization, misuse intrusion detection, and network traffic classification, but also is more computationally efficient, compared to the existing batch variational learning methods and the FC neural networks. Future work can be devoted to simultaneously dealing with the problem of parameter estimation, model selection and feature selection through using the ESVI framework in order to improve the performance of modeling high-dimensional positive vectors. Another ongoing work may focus on the extension of the finite IBMM to the infinite case.

Acknowledgment

This work was supported in part by the Fundamental Research Funds for the Central Universities; in part by the General Project of Science and Technology Plan of Beijing Municipal Commission of Education under Grant KM201910009014; in part by the National Natural Science Foundation of China (NSFC) under Grant 72101157, Grant 62172193, and Grant 61802094, and in part by the National Natural Science Foundation of Zhejiang Province under Grant LY20F020012.

References

[1] T. Bdiri and N. Bouguila, "Positive vectors clustering using inverted dirichlet finite mixture models," *Expert Systems with Applications*, vol. 39, no. 2, pp. 1869–1882, 2012.

[2] Z. Ma, J. Xie, Y. Lai, J. Taghia, J. Xue, and J. Guo, "Insights into multiple/single lower bound approximation for extended variational inference in non-gaussian structured data modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2240–2254, 2020.

[3] N. Moustafa, K.-K. R. Choo, I. Radwan, and S. Camtepe, "Outlier dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 1975–1987, 2019.

[4] W. Fan and N. Bouguila, "Modeling and clustering positive vectors via nonparametric mixture models of liouville distributions," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3193–3203, 2020.

[5] N. Moustafa, G. Misra, and J. Slay, "Generalized outlier gaussian mixture technique based on automated association features for simulating and detecting web application attacks," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 2, pp. 245–256, 2021.

[6] Y. Lai, H. Cao, L. Luo, Y. Zhang, F. Bi, X. Gui, and Y. Ping, "Extended variational inference for gamma mixture model in positive vectors modeling," *Neurocomputing*, vol. 432, pp. 145–158, 2021.

[7] W. Fan, R. Wang, and N. Bouguila, "Simultaneous positive sequential vectors modeling and unsupervised feature selection via continuous hidden markov models," *Pattern Recognition*, vol. 119, 2021.

[8] J. Taghia and A. Leijon, "Variational inference for watson mixture model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1886–1900, 2016.

[9] J. Taghia, Z. Ma, and A. Leijon, "Bayesian estimation of the von-mises fisher mixture model with variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 9, pp. 1701–1715, 2014.

[10] N. Bouguila, D. Ziou, and J. Vaillancourt, "Unsupervised learning of a finite mixture model based on the dirichlet distribution and its application," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1533–1543, 2004.

[11] W. Fan and N. Bouguila, "Learning finite beta-liouville mixture models via variational bayes for proportional data clustering," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2012.

[12] H.-C. Li, V. A. Krylov, P.-Z. Fan, J. Zerubia, and W. J. Emery, "Unsupervised learning of generalized gamma mixture model with application in statistical modeling of high-resolution sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 4, pp. 2153–2170, 2016.

[13] C. Hu, W. Fan, J. X. Du, and N. Bouguila, "A novel statistical approach for clustering positive data based on finite inverted beta-liouville mixture models," *Neurocomputing*, vol. 333, pp. 110–123, 2019.

[14] Z. Ma, Y. Lai, J. Xie, D. Meng, W. B. Kleijn, J. Guo, and J. Yu, "Dirichlet process mixture of generalized inverted dirichlet distributions for positive vector data with extended variational inference," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021.

[15] A. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[16] A. Saumard and F. Navarro, "Finite sample improvement of akaikes information criterion," *IEEE Transactions on Information Theory*, vol. 67, no. 10, pp. 6328–6343, 2021.

[17] X. Ke, Y. Zhao, and L. Huang, "On accurate source enumeration: A new bayesian information criterion," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1012–1027, 2021.

[18] F. Ridder, R. Pintelon, J. Schoukens, and D. Gillikin, "Modified aic and mdl model selection criteria for short data records," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 1, pp. 144–150, 2005.

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

[20] H. Li, V. A. Krylov, P. Fan, J. Zerubia, and W. J. Emery, "Unsupervised learning of generalized gamma mixture model with application in statistical modeling of high-resolution sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 4, pp. 2153–2170, 2016.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer-Verlag, 2006.

[22] Z. Ma, Y. Lai, W. B. Kleijn, Y.-Z. Song, L. Wang, and J. Guo, "Variational bayesian learning for dirichlet process mixture of inverted dirichlet distributions in non-gaussian image feature modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 449–463, 2019.

[23] M. Hoffman, D. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[24] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent dirichlet allocation," in *In advances in Neural Information Processing Systems (NIPS)*, 2010.

[25] F. Tomasi, P. Chandar, G. Levy-Fix, M. Lalmas-Roelleke, and Z. Dai, "Stochastic variational inference for dynamic correlated topic models," in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

[26] N. Foti, J. Xu, D. Laird, and E. Fox, "Stochastic variational inference for hidden markov models," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[27] J. Matthew and W. Alan, "Stochastic variational inference for bayesian time series models," in *International Conference on Machine Learning (ICML)*, 2014.

[28] J. Taghia and A. Leijon, "Variational inference for watson mixture model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1886–1900, 2016.

[29] T. Minka, "Expectation propagation for approximate bayesian inference," Ph.D. dissertation, April 2001.

[30] D. Blei and J. Lafferty, "Correlated topic models," in *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[31] ——, "A correlated topic model of science," *The Annals of Applied Statistics*, vol. 1, no. 1, pp. 17–35, 2007.

[32] M. Hoffman, D. Blei, and P. Cook, "Bayesian nonparametric matrix factorization for recorded music," in *Proceedings of International Conference on Machine Learning (ICML)*, 2010.

[33] R. Madsen, D. Kauchak, and C. Elkan, "Modeling word burstiness using the dirichlet distribution,," in *International Conference of Machine Learning (ICML)*, 2005.

[34] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[35] X. Xue and Z. Zhou, "Distributional features for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 3, pp. 428–442, 2009.

[36] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A bayesian classification approach using class-specific features for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1602–1606, 2016.

[37] J. Jiang, R. Liou, and S. Lee, "A fuzzy self-constructing feature clustering algorithm for text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 3, pp. 335–349, 2011.

[38] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 721–735, 2009.

[39] N. Bouguila and D. Ziou, "A dirichlet process mixture of generalized dirichlet distributions for proportional data modeling," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 107–122, 2010.

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

[41] M. Craven, F. D. DiPasquo, D., M. T. McCallum, A.K., K. Nigam, and S. Slattery, "Learning to extract symbolic knowledge from the world wide web," in *Proceedings of the National Conference on Artificial Intelligence (NCAI)*, 1998.

[42] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Learning to classify text from labeled and unlabeled documents," in *Proceedings of the National Conference on Artificial Intelligence (NCAI)*, 1998.

[43] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.

[44] Y. Ping, Y. Zhou, C. Xue, and Y. Yang, "Efficient representation of text with multiple perspectives," *The Journal of China Universities of Posts and Telecommunications*, vol. 19, no. 1, pp. 101–111, 2012.

[45] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[46] Y. Ling, W. Guan, Q. Ruan, H. Song, and Y. Lai, "Variational learning for the inverted beta-liouville mixture model and its application to text categorization," *International Journal of Interactive Multimedia and Artificial Intelligence*, https://arxiv.org/abs/2112.14375.

[47] Y. Xiang, W. Zhou, and M. Guo, "Flexible deterministic packet marking: An ip traceback system to find the real source of attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 4, pp. 567–580, 2009.

[48] J. Luo, J. Li, L. Jiao, and J. Cai, "On the effective parallelization and near-optimal deployment of service function chains," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1238–1255, 2021.

[49] T. Auld, A. W. Moore, and S. F. Gull, "Bayesian neural networks for internet traffic classification," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 223–239, 2007.

[50] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and A. V. Vasilakos, "An effective network traffic classification method with unknown flow detection," *IEEE Transactions on Network and Service Management*, vol. 10, no. 2, pp. 133–147, 2013.

[51] A. M. Sadeghzadeh, S. Shiravi, and R. Jalili, "Adversarial network traffic: Towards evaluating the robustness of deep-learning-based network traffic classification," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1962–1976, 2021.

[52] T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[53] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescap, "A big data-enabled hierarchical framework for traffic classification," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2608–2619, 2020.

[54] The UNIBS Anonymized 2009 Internet Traces [EB/OL]. 18 Mar. 2010 http://www.ing.unibs.it/ ntw/tools/traces.

[55] Y. Ping, Y. Chang, Y. Zhou, Y. Tian, Y. Yang, and Z. Zhang, "Fast and scalable support vector clustering for large-scale data analysis," *Knowledge and Information Systems*, vol. 43, pp. 281–310, 2008.

[56] W. Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 577–583, 2008.

[57] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008.

[58] Y. Xie, D. Feng, Y. Hu, Y. Li, S. Sample, and D. Long, "Pagoda: A hybrid approach to enable efficient real-time provenance based intrusion detection in big data environments," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1283–1296, 2020.

[59] J. Cai, Z. Huang, L. Liao, J. Luo, and W. Liu, "Appm: Adaptive parallel processing mechanism for service function chains," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1540–1555, 2021.

 This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tnnls.2022.3213518, IEEE Transactions on Neural Networks and Learning Systems

[60] P. Marteau, "Random partitioning forest for point-wise and collective anomaly detection–application to network intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2157–2172, 2021.
[61] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 1, pp. 66–79, 2021.
[62] C. Guo, Y. Zhou, Y. Ping, S. Luo, Y. Lai, and Z. Zhang, "Efficient intrusion detection using representative instances," *Computers & Security*, vol. 39, pp. 255–267, 2013.
[63] G. Deshpande, P. Wang, D. Rangaprakash, and B. Wilamowski, "Fully connected cascade artificial neural network architecture for attention deficit hyperactivity disorder classification from functional magnetic resonance imaging data," *IEEE Transactions on Cybernetics*, vol. 45, no. 12, pp. 2668–2679, 2015.

Yuping Lai received his Ph.D. degree in Information Security from Beijing University of Posts and Telecommunications, Beijing, China, in 2014. He is currently an associate professor in the School of Cyberspace Security at Beijing University of Posts and Telecommunications, Beijing, China. His research interests include information security, machine learning, and data mining.

Lijuan Luo is currently an associate professor in Shanghai International Studies University, China, since 2015. He received her Ph.D. degree in Management Science and Engineering from Beijing University of Posts and Telecommunications, China, in 2015. Her research interests include information management, machine learning, big data and data decision.

Yanhui Guo received his B. S. degree from Zhengzhou University, China, M.S. degree from Harbin Institute of Technology, China, and a Ph.D. degree in the Department of Computer Science, Utah State University, USA. He was a research fellow in the Department of Radiology at the University of Michigan and an assistant professor at St. Thomas University. Dr. Guo is currently an associate professor in the Department of Computer Science at the University of Illinois Springfield. Dr. Guo has published more than 100 journal papers and 30 conference papers, completed more than 10 grant-funded research projects, and worked as an associate editor of different international journals, reviewers for top journals and conferences. His research area includes computer vision, machine learning, big data analytics, and computer-aided detection/diagnosis.

Heping Song received the Ph.D. degree in computer application technology from Sun Yat-sen University, Guangzhou, China, in 2011. He is currently a Associate Professor with the Department of Software Engineering, School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China. His current research interests include low-level vision, inverse problems, computer vision and deep learning.

Wenbo Guan received the bachelor's degree from the Beijing International Studies University and the master's degree from the North China University of Technology, now he is currently pursuing the Ph.D. degree in the Institute of Information Engineering, CAS. His research interests include pattern recognition, machine learning and cyberspace security.

Hongying Meng received the Ph.D. degree in communication and electronic systems from Xian Jiaotong University. He was a Lecturer with the Electronic Engineering Department, Tsinghua University, China. He is currently a Reader with the Electronic and Electrical Engineering Department, Brunel University London, U.K. He has wide research interests, including digital signal processing, machine learning, human-computer interaction, computer vision, image processing, and embedded systems.