Temporal Multi-View Inconsistency Detection for Network Traffic Analysis

Houping Xiao[1], Jing Gao[1], Deepak S. Turaga[2], Long H. Vu[2], and Alain Biem[2]
[1]SUNY Buffalo, Buffalo, NY USA
[2]IBM T.J. Watson Research Center, NY USA
{houpingx,jing}@buffalo.edu, {turaga,lhvu,biem}@us.ibm.com

ABSTRACT

In this paper, we investigate the problem of identifying inconsistent hosts in large-scale enterprise networks by mining multiple views of temporal data collected from the networks. The time-varying behavior of hosts is typically consistent across multiple views, and thus hosts that exhibit inconsistent behavior are possible anomalous points to be further investigated. To achieve this goal, we develop an effective approach that extracts common patterns hidden in multiple views and detects inconsistency by measuring the deviation from these common patterns. Specifically, we first apply various anomaly detectors on the raw data and form a three-way tensor (host, time, detector) for each view. We then develop a joint probabilistic tensor factorization method to derive the latent tensor subspace, which captures common time-varying behavior across views. Based on the extracted tensor subspace, an inconsistency score is calculated for each host that measures the deviation from common behavior. We demonstrate the effectiveness of the proposed approach on two enterprise-wide network-based anomaly detection tasks. An enterprise network consists of multiple hosts (servers, desktops, laptops) and each host sends/receives a time-varying number of bytes across network protocols (e.g.,TCP, UDP, ICMP) or send URL requests to DNS under various categories. The inconsistent behavior of a host is often a leading indicator of potential issues (e.g., instability, malicious behavior, or hardware malfunction). We perform experiments on real-world data collected from IBM enterprise networks, and demonstrate that the proposed method can find hosts with inconsistent behavior that are important to cybersecurity applications.

Categories and Subject Descriptors

F.6.0 [**Theory and Algorithms for Application Domains**]: Machine Learning Theory–Unsupervised Learning; G.0.3 [**Probability and Statistics**]: Probabilistic Algorithms

WWW'2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2745399.

Keywords

Network Traffic Analysis, Temporal Multi–View Learning, Tensor Factorization, Inconsistency Detection

1. INTRODUCTION

An immense amount of data is collected each day from enterprise networks, such as network packet flow, URL requests, and other network activities. It is important to monitor network traffic and identify hosts with suspicious behavior in a timely manner to prevent large-scale damage to the whole network. This is a challenging task due to the fact that network traffic data are heterogeneous, noisy and gigantic in which meaningful patterns are deeply hidden.

Some existing work tackles this challenge by developing anomaly detection techniques that identify hosts whose data are significantly different from those of the majority of the hosts [12, 20, 23]. However, these techniques suffer from the limitations that they ignore the intrinsic relationships among attributes. In fact, network traffic data typically involve multiple views of attributes, each of which captures network traffic from a particular perspective. For example, network traffic data can be collected through different protocols, such as TCP, UDP and ICMP. Although absolute traffic volume may differ among protocols, there exists some consistency among these views. For example, when the network undergoes heavy loads, we can expect increase in the traffic volumes in all the views.

In this paper, we propose a novel approach to detect hosts with unusual behavior by mining enterprise network traffic data that are collected from multiple views. Different from existing single-view anomaly detection approaches, the proposed method considers correlations among different views and detects hosts that have inconsistent behavior across these views. There are two major challenges with this problem. We describe these challenges and the corresponding solutions as follows.

Inconsistency Detection
The key idea of this task is to model commonalities across views and flag those hosts that violate such common patterns as the inconsistent hosts we try to detect. However, network data may be quite noisy, so it is hard to find common patterns at the surface. Moreover, data from different views may have different value ranges, and this makes the comparison across views difficult. Therefore, we propose to first preprocess each view's data by applying existing anomaly detection algorithms, such as Local Outlier Factor (LOF) [4,6]. Since we are interested in hosts with anomalous behavior, this step converts data from different views into

comparable features and discards noisy information while keeping useful information that helps with the task.

However, even after the application of anomaly detectors on each view, it is still challenging for researchers to compare anomaly detector outputs from different views. Figure 1 and 2 show the anomaly detector outputs for 50 hosts on a network traffic dataset and a DNS dataset respectively. Each dataset consists of 4 views (four protocols or four categories of URL). More details about these two datasets can be found in experiments. Each plot in each figure illustrates one view, in which x-axis denotes host ID and y-axis denotes anomaly detector outputs. As can be seen, there exist some commonality across views, but it is difficult to extract such common patterns by a simple comparison. Correspondingly, we may not be able to distinguish inconsistent and consistent hosts easily. This observation motivates us to project multi-view data into a new space with the hope that the inconsistent and consistent hosts can be well separated in this new space.

Temporal Behavior

Another challenge is that the behavior of hosts may evolve over time, and thus approaches that model statical behavior of hosts cannot work. The temporal patterns in hosts' behavior must be taken into consideration when detecting inconsistent hosts. For example, a host with a very high volume of network traffic should not be considered as anomalous if the high volume occurs on weekdays, but it may indicate suspicious behavior if it occurs on weekends. Therefore, anomaly detector scores from weekdays and weekends should be considered as two separate groups and we model consistency on each group separately.

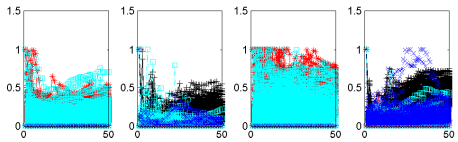


Figure 1: Raw detector scores from netflow data

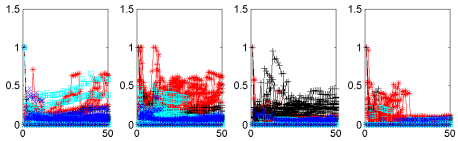


Figure 2: Raw detector scores from DNS data

Summary of Proposed Framework

To tackle the aforementioned challenges, we develop an effective approach to automatically extract common patterns across multiple views and detect hosts which behave inconsistently from time-evolving network data. We name the approach Temporal Multi-View Inconsistency Detection (TMVID), which consists of the following steps: 1) We apply various anomaly detectors on different attributes of network data, such as the number of bytes, flows, and packets to obtain anomaly detection output for each time snapshot and each view. For each view, the scores can be summarized as a tensor with three dimensions: host, time and detector. 2) We then conduct joint probabilistic factorization on multiple tensors simultaneously with the constraint that the projection matrices across multiple views are similar. In this way, the commonality across multiple views is captured by the mean latent tensor and this projection takes temporal context into consideration. 3) After tensor factorization, we calculate inconsistency score for each host as the variance of the similarity between each view's latent tensor and the mean tensor. This score measures the deviation of each host's pattern in one view from the average behavior considered across views.

Applications and Experiments

We implement the proposed approach and apply it to two network datasets collected from IBM enterprise networks, Network flow traffic and domain name system data sets. The task is to detect inconsistent hosts based on information collected from multiple internet protocols and URL request categories, respectively. Besides network traffic applications, the proposed method is applicable to other scenarios that involve multiple views of time-varying data to detect inconsistent hosts, events, or actions. The detected inconsistency can further contribute to the improvement of safety and security in the cyber or physical world. The advantage of the proposed approach is its ability to correlate and compare information from multiple views by joint tensor factorization to extract information inconsistency. Experimental results show that the proposed mechanism has the capacity of detecting inconsistent behavior. Especially, the inconsistent hosts can be distinguished from the consistent hosts in the final projection results. By comparing with the static matrix factorization approach, we demonstrate the importance of modeling temporal behavior of hosts. Running time experiments show that TMVID is efficient and scalable, which makes it applicable in the real-world big data environment.

To summarize, we make the following contributions:

- We propose the important problem of inconsistency detection on temporal data with multiple views. Different from traditional approaches that work on single views and static data, the proposed approach can output more meaningful alerts from heterogeneous, dynamic and noisy data for the purpose of cyber security.

- We propose an effective approach based on Joint Probabilistic Tensor Factorization (JPTF) to capture the latent common behavior across multiple views. For each view, we form a tensor by applying various anomaly detectors on the raw data and record the anomalous scores for each time snapshot. After joint tensor factorization on multiple tensors, we calculate each host's inconsistency score by comparing the latent tensor of each view with the average lathent tensor.

- As the major component of the proposed method, joint probabilistic tensor factorization is modeled as an optimization problem and we propose an algorithm to solve this problem by iteratively updating the projection matrices and latent tensors.

- We validate the proposed algorithm on both synthetic and real-world data sets, and the results demonstrate the advantages of the proposed approach in detecting inconsistent behavior. Due to its ability of modeling temporal behavior and extracting common patterns, the proposed approach outperforms existing baselines that only model static behavior and conduct simple across-view comparison.

2. PRELIMINARY

In this section, we introduce the notations and the key tensor operations used throughout this paper.

DEFINITION 2.1 (TENSOR). *A tensor is a mathematical representation of a multi-way array. The order of a tensor is the number of modes (or ways). The dimensionality of a mode is the number of elements in that mode.*

Specifically, a first-order tensor is a vector denoted by a lowercase letter, x; a second-order tensor is a matrix denoted by a capital letter X; and a higher-order tensor has three or more modes denoted by a Euler script letter $\mathcal{X}$. For example, a tensor $\mathcal{X} \in \mathbb{R}^{4\times5\times6}$ has 3 modes with dimensionality of 4, 5, and 6 respectively. Denote the i-th entry of a vector by x_i, the (i,j)-th element of a matrix X by X_{ij}, and the (i,j,k)-th element of a third-order tensor $\mathcal{X}$ by $\mathcal{X}_{ijk}$. Indices range from l to their capital version, e.g. $l = 1, \cdots, L$.

EXAMPLE 1. *We have* 50 *detectors,* 500 *hosts, and* 500 *days in a cyber network application. The detector scores form a tensor* $\mathcal{X} \in \mathbb{R}^{50\times500\times500}$. $\mathcal{X}_{ijk}$ *represents the detector score obtained by applying* i*-th detector on the number of bytes of* j*-th host on day* k.

DEFINITION 2.2 (MATRICIZATION). *Matricization, also known as unfolding or flattening, is the process of reordering the elements of an* N*-mode array into a matrix.*

Let $X_{(d)}$ denote the mode-d matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N}$, where columns of the matrix $X_{(d)}$ corresponds to the mode-n fibers. The $(i_1, i_2, \cdots, i_N)$ element of tensor maps to the (i_d, j) elements of the matrix $X_{(d)}$ in which $j = 1 + \sum_{k=1,k\neq d}^{N}(i_k - 1)J_k$, where $J_K = \Pi_{m=1,m\neq d}^{k-1} I_m$.

For example, mode-1 matricization of $\mathcal{X}$ in cyber network scenario, we can obtain $X_{(1)} \in \mathbb{R}^{50\times(500\cdot500)}$; mode-2 matricization results in $X_{(2)} \in \mathbb{R}^{500\times(50\cdot500)}$; and mode-3 matricization $X_{(3)} \in \mathbb{R}^{500\times(50\cdot500)}$. The concept is easier to understand using an example in [17].

EXAMPLE 2. *Assume* $\mathcal{Y} \in \mathbb{R}^{3\times4\times2}$ *and*

$$\mathbf{Y_1} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix};$$

$$\mathbf{Y_2} = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}.$$

Therefore, we have:

$$\mathbf{Y_{(1)}} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix};$$

$$\mathbf{Y_{(2)}} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix};$$

and

$$\mathbf{Y_{(3)}} = \begin{bmatrix} 1 & 2 & 3 & \dots & 10 & 11 & 12 \\ 13 & 14 & 15 & \dots & 22 & 23 & 24 \end{bmatrix}.$$

DEFINITION 2.3 (VECTORIZATION). *Vectorization,* $vec(\cdot)$*, is the process of reordering the elements of an* N*-mode array into a vector.*

EXAMPLE 3. *Assume* $\mathcal{Y}$ *are defined in Example 2. Thus,* $vec(\mathcal{Y}) = (1, 2, \cdots, 24)^T$.

DEFINITION 2.4 (MODE-d PRODUCTION). *The mode-*d *matrix product of a tensor* $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots I_d\cdots\times I_N}$ *with a matrix* $U \in \mathbb{R}^{J\times I_d}$ *element-wise is defined as:*

$$(\mathcal{X} \times_d U)_{i_1\cdots i_{d-1} j i_{d+1}\cdots i_N} = \sum_{i_d=1}^{I_d} X_{i_1\cdots i_d\cdots i_N} U_{j i_d}, \qquad (1)$$

where $\mathcal{X}\times_d U$ *has a size of* $I_1\times\cdots\times I_{d-1}\times J\times I_{d+1}\times\cdots\times I_N$.

Semantically, the mode-d production transforms $\mathcal{X}$ to a new tensor $\mathcal{X} \times_d U$ by applying the linear transformation described by the matrix U to each of the mode-d fibers of $\mathcal{X}$. We introduce the following simple notation for multiplication in each mode:

$$\mathcal{G}\Pi_{\times d} U^d \triangleq \mathcal{G} \times_1 U^1 \times_2 U^2 \times \cdots \times_N U^N. \qquad (2)$$

DEFINITION 2.5 (TENSOR NORM). *The Frobenius norm of an* N*-mode tensor* $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ *is:*

$$\|\mathcal{X}\|_F^2 = \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_n=1}^{I_N} x_{i_1 i_2\cdots i_N}^2. \qquad (3)$$

Especially, we have the following useful property about the Frobenius norm of N-mode tensor [17].

PROPERTY 2.6. *Assume that* $\mathcal{X}$ *and* $\mathcal{G}\Pi_{\times d}U^d$ *are three-mode tensors, the Frobenius norm of* $\mathcal{X} - \mathcal{G}\Pi_{\times d}U^d$ *has the following properties:*

$$\|\mathcal{X} - \mathcal{G}\Pi_{\times d}U^d\|_F^2 = \|X_{(d)} - U^d G_{(d)}(\bigotimes_{l=1,l\neq d}^{N} U^d)^T\|_F^2 \qquad (4)$$

$$= \|vec(\mathcal{X}) - \bigotimes_{d=1}^{N} U^d vec(\mathcal{G})\|_F^2, \qquad (5)$$

where $\bigotimes_{l=1,l\neq d}^{N} U^l = U^1 \otimes \cdots \times U^{d-1} \otimes U^{d+1} \otimes \cdots \otimes U^N$, *and* $\bigotimes_{d=1}^{N} U^d = U^1 \otimes U^2 \times \cdots \times U^N$.

3. METHODOLOGY

We first formally formulate the problem and pose the joint probabilistic tensor factorization based framework, Temporal Multi-View Inconsistency Detection (TMVID) in Section 3.1. The details of the proposed framework, TMVID, are discussed in Section 3 and the complete algorithm of probabilistic tensor factorization is described in Section 3.3.

3.1 Problem Formulation

First we introduce some notations and discuss the problem setting based on the network flow traffic scenario. Assume there are M views to describe the behavior of K hosts over T days. In the network flow traffic scenario, we have M protocols, such as TCP incoming/outgoing traffic, UDP incoming/outcoming traffic and so on. Each view consists of many attributes: the number of bytes, flows, and packets. We aim to apply inconsistency analysis to identify unusual hosts which behave inconsistently across M views. In order to achieve this goal, we propose a three step framework. First, we conduct N detectors on these attributes on each day to convert the raw data into comparable level and then form the detector scores into M tensors. We denote the detector scores tensor from the s-th view as $\mathcal{X}^s \in \mathbb{R}^{N\times K\times T}$.

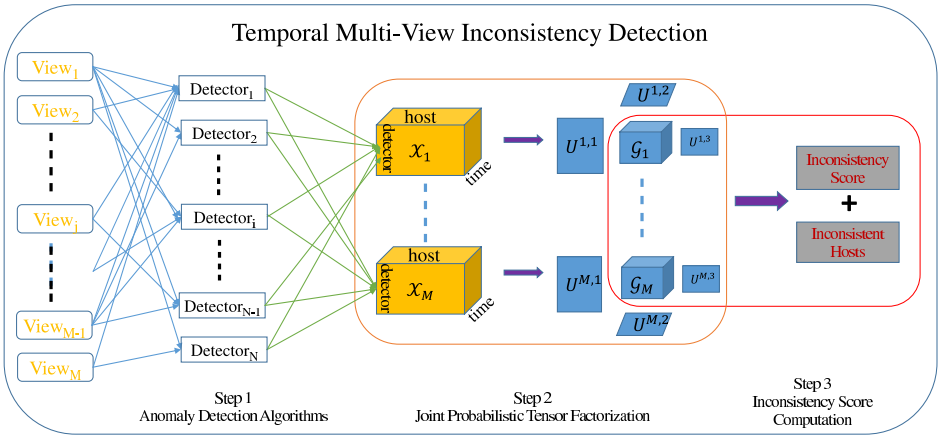


Figure 3: Complete flow of the proposed mechanism: Temporal Multi-View Inconsistency Detection

Namely, $\mathcal{X}^s_{ijk}$ means the j-th host's detector score obtained by conducting the i-th detector on the s-th view in day k . Second, we apply joint probabilistic tensor factorization to project the detector scores into the latent subspaces. Third, we calculate the inconsistency score of each host based on similarity between its latent tensors and the mean latent tensor.

3.2 Proposed Framework

In this section, we are going to introduce the proposed three-step framework as shown in Figure 3. First, we apply several anomaly detectors across multiple views. Second, we conduct probabilistic tensor factorization to capture the subspace shared by multiple views, which is the core component of TMVID and the details will be given in following. Finally, the inconsistency score of each entity will be calculated based on the dissimilarity from the common subspace.

Probabilistic Tensor Factorization

Probabilistic tensor factorization multilinearly project the observed tensor $\mathcal{X}^s$ for $s = 1, 2, \cdots, M$ in the high-dimensional space $\mathbb{R}^{N\times K\times T}$ to the corresponding latent tensors $\mathcal{G}^s$ in the low-dimensional space $\mathbb{R}^{C_N\times K\times C_T}$, i.e.

$$\mathcal{X}^s = \mathcal{G}^s \Pi_{\times d} U^{s,d} + \mathcal{E}^s, \tag{6}$$

where

- $\mathcal{G}^s \in \mathbb{R}^{C_N\times K\times C_T}$ is the latent tensor. Each entry $\mathcal{G}_{uvw}$ of $\mathcal{G}^s$ stands for the detector score at the u-th detector and w-th timestamp cluster for v-th host.
- $U^{s,d}$ is the d-th projection matrix, which constructs the multilinear mapping between the observed detector scores and the latent tensors.
- $\mathcal{E}^s \in \mathbb{R}^{M\times K\times T}$ is the residue tensor. Each entry of $\mathcal{E}^s$ is assumed to follow a Gaussian distribution $N(0, \sigma^2)$.

Based on these observations, we therefore introduce a probabilistic tensor factorization model to describe the distribution of the entry of residue tensor

$$Pr(\mathcal{E}^s|\mathcal{X}^s, \mathcal{G}^s, U^{s,d}) \propto exp(-\|\mathcal{X}^s - \mathcal{G}^s \Pi_{\times d} U^{s,d}\|^2_F). \tag{7}$$

Let $\Theta = \{\mathcal{G}^s, U^{s,d}|s = 1, 2, \cdots, M, d = 1, 2, 3\}$ denote the parameters set. All parameters are estimated from the observed tensor data. Next, we mathematically formulate the task as an optimization problem.

Regarding a three-dimension network flow traffic scenario, we assume detector scores from M views have been collected. Note that although we only consider a three-dimension tensor for ease of presentation, the extension to high dimension setting is straightforward. The log-likelihood of parameter set Θ given M observed tensors is expressed as:

$$\begin{aligned}\mathcal{L}(\Theta) &\propto \frac{1}{M}\log \Pi_{s=1}^{M} Pr(\mathcal{E}^s|\mathcal{X}^s, \Theta) \\ &\propto -\frac{1}{M}\sum_{s=1}^{M}\|\mathcal{X}^s - \mathcal{G}^s\Pi_{\times d}U^{s,d}\|^2_F. \end{aligned}\tag{8}$$

Consistent hosts are those whose behavior is consistent across different views. Thus, we assume the behavior of anomaly detectors shall be similar across different attributes. Based on this observation, we estimate the parameters by minimizing the penalized log-likelihood function, which is defined as:

$$\mathcal{L}_\Lambda(\Theta) \propto -\frac{1}{2}\mathcal{L}(\Theta) + \sum_{l=1}^{3}\sum_{s=1}^{M}\left(\frac{\lambda_l}{2}\|U^{s,l} - U^{*,l}\|^2_F\right), \tag{9}$$

where $U^{*,l} = \frac{1}{M}\sum_{s=1}^{M} U^{s,l}, l = 1, 2, 3$, and $\Lambda = [\lambda_1, \lambda_2, \lambda_3]$ is a regularizer parameter vector. The first term represents the negative log-likelihood, while the second term is a regularizer which has two-fold meaning: (1) the behavior of the detectors and the pattern of days shall be similar, and (2) it is adopted to prevent overfitting. More specifically, denote $\mathcal{L}_\Lambda(U^{s,l}|\Theta)$ as the objective function with respect to $U^{s,l}$. $\mathcal{L}_\Lambda(\mathcal{G}^s|\Theta)$ is the objective functions in terms of $\mathcal{G}^s$.

Next, we propose an algorithm which iteratively optimizes $\mathcal{L}_\Lambda(U^{s,l}|\Theta)$ and $\mathcal{L}_\Lambda(\mathcal{G}^s|\Theta)$ by constructing the corresponding surrogate functions to decouple the parameters.

Model Inference

Denote by $\Theta_n = \{\mathcal{G}^s_n, U^{s,l}_n|1 \le s \le M, 1 \le l \le 3\}$ the parameters set on n-th iteration. We construct surrogate functions $Q_1(U^{s,l}|\Theta; \Theta_{(n)})$ and $Q_2(\mathcal{G}^s|\Theta; \Theta_{(n)})$, and will show they are tight upper bounds of $\mathcal{L}_\Lambda(U^{s,l}|\Theta)$ and $\mathcal{L}_\Lambda(\mathcal{G}^s|\Theta)$ with

respect to $U^{s,l}$ and $\mathcal{G}^s$ separately,

$$Q_1(U^{s,l}|\Theta;\Theta_n) = \sum_{s=1}^{M}\Bigg[\sum_{i,j}\frac{\left[U_n^{s,l}(A_l^s(A_l^s)^T+\lambda_l I_l)\right]_{ij}(U_{ij}^{s,l})^2}{2{U_n^{s,l}}_{ij}} - 2\sum_{i,j}{U_n^{s,l}}_{ij}\left[X_{(l)}^s(A_l^s)^T\right]_{ij}(1+\log\frac{U_{ij}^{s,l}}{{U_n^{s,l}}_{ij}}) - 2\lambda_l\sum_{i,j}{U_n^{s,l}}_{ij}U_{ij}^{s,l}(1+\log\frac{U_{ij}^{s,l}}{{U_n^{s,l}}_{ij}})\Bigg], \quad (10)$$

$$Q_2(\mathcal{G}^s|\Theta;\Theta_n) = \sum_{s=1}^{M}\Bigg[\sum_{l}\frac{\left[vec(\mathcal{G}_n^s)U^s(U^s)^T\right]_l vec(\mathcal{G}^s)_l^2}{2vec(\mathcal{G}^s)} - 2\sum_l vec(\mathcal{G}_n^s)_l vec(\mathcal{X}^s)_l(U^s)^T\left(1+\log\frac{vec(\mathcal{G}^s)_l}{vec(\mathcal{G}_n^s)_l}\right)\Bigg]; \quad (11)$$

where the terms $X_{(l)}^s$ and $G_{(l)}^s$ are matrices unfolding $\mathcal{X}^s$ and $\mathcal{G}^s$ on l-th mode, $vec(\cdot)$ is vectorization operation of tensor as defined above, $A_l^s = G_{(l)}^s(U^{s,m}\otimes U^{s,n})^T$ in which $m,n\neq l$ and $m>n$, and $U^s = U^{s,3}\otimes U^{s,2}\otimes U^{s,1}$.

Note that $Q_1(U^{s,l}|\Theta;\Theta_n)$ and $Q_2(\mathcal{G}^s|\Theta;\Theta_n)$ enjoy the following desired properties:

$$\begin{cases} Q_1(U^{s,l}|\Theta;\Theta_n)\geq\mathcal{L}_\Lambda(U^{s,l}|\Theta), & \forall\Theta,\Theta_n;\\ Q_1(U^{s,l}|\Theta;\Theta_n)=\mathcal{L}_\Lambda(U^{s,l}|\Theta_n), & \forall\Theta_n. \end{cases}$$

and

$$\begin{cases} Q_2(\mathcal{G}^s|\Theta;\Theta_n)\geq\mathcal{L}_\Lambda(\mathcal{G}^s|\Theta), & \forall\Theta,\Theta_n;\\ Q_2(\mathcal{G}^s|\Theta;\Theta_n)=\mathcal{L}_\Lambda(\mathcal{G}^s|\Theta_n), & \forall\Theta_n. \end{cases}$$

The proof of these desired properties of the surrogate functions can be found in the Appendix. Assume that we have obtained the solutions, $U_{n+1}^{s,l}$ and $\mathcal{G}_{n+1}^s$, of the optimization problems $\min_{U^{s,l}\in\Theta}Q_1(U^{s,l}|\Theta;\Theta_n)$ and $\min_{\mathcal{G}^s\in\Theta}Q_2(\mathcal{G}^s|\Theta;\Theta_n)$. Following the above properties, it is easy to deduce that $\mathcal{L}_\Lambda(U^{s,l}|\Theta_n)\geq\mathcal{L}_\Lambda(U^{s,l}|\Theta_{n+1})$ and $\mathcal{L}_\Lambda(\mathcal{G}^s|\Theta_n)\geq\mathcal{L}_\Lambda(\mathcal{G}^s|\Theta_{n+1})$, which means that minimizing $Q_1(U^{s,l}|\Theta;\Theta_n)$ and $Q_2(\mathcal{G}^s|\Theta;\Theta_n)$ at each iteration guarantees that $\mathcal{L}_\Lambda(U^{s,l}|\Theta_n)$ and $\mathcal{L}_\Lambda(\mathcal{G}^s|\Theta_n)$ will monotonically decrease w.r.t. $U^{s,l}$ and $\mathcal{G}^s$ respectively.

Updating Parameters

Owing to the desired property of surrogate functions built above, we can derive the closed form solution of $U^{s,l}$ and $\mathcal{G}^s$ by solving the optimization problems $\min_{U^{s,l}\in\Theta}Q_1(U^{s,l}|\Theta;\Theta_n)$ and $\min_{\mathcal{G}^s\in\Theta}Q_2(\mathcal{G}^s|\Theta;\Theta_n)$, respectively. By deriving the derivatives of $Q_1(U^{s,l}|\Theta;\Theta_n)$ and $Q_2(\mathcal{G}^s|\Theta;\Theta_n)$ with respect to $U^{s,l}$ and $\mathcal{G}^s$ separatively and setting them equal to zero, we can obtain their update rules as

$$U_{ij}^{s,l}\leftarrow U_{ij}^{'s,l}\sqrt{\frac{\left[X_{(l)}^sA_l^s+\alpha_lU^{*,l}\right]_{ij}}{\left[U'^{s,l}(A_l^s(A_l^s)^T+\alpha_lI_l)\right]_{ij}}}, \quad (12)$$

$$vec(\mathcal{G}^s)_k\leftarrow vec(\mathcal{G}'^s)_k\sqrt{\frac{\left[U^svec(\mathcal{X}^s)\right]_k}{\left[U^sU^{sT}vec(\mathcal{G}'^s)\right]_k}}. \quad (13)$$

Here, instead of updating the exact latent tensors, we offer the update rule for their corresponding vector obtained from vectorization operation. One more mapping is necessary from updated vector form to the latent tensor.

Calculating the Inconsistency Score

Note that all of the M views describe the behavior of the K hosts; therefore, we expect that they shall achieve the similar projection for each host. Joint probabilistic tensor factorization model maps the observed tensor $\mathcal{X}^s$ into an unobserved latent tensor $\mathcal{G}^s$. As the projection matrices are constrained to be similar, the differences across views appears more in $\mathcal{G}^s$. Denote $\mathcal{G}^*$ as the average latent tensor. We first calculate the similarity between $\mathcal{G}^s$ and $\mathcal{G}^*$, and then define host's inconsistency score as the variance of the similarity over the latent subspace. A higher inconsistency score means the variance of similarity between latent subspaces is bigger, which in return represents a bigger difference across views.

3.3 Complete Algorithm

Algorithm 1: Inference of Joint Probabilistic Tensor Model

Data: $\mathcal{X}^s$,C_N, C_K, and $\Lambda=[\lambda_1,\lambda_2,\lambda_3]$.
Result: $U^{s,l}$, $\mathcal{G}$, and Inconsistency Scores list I.
begin
  `/* Initialization */`
  initialize $U^{s,l}$ according to Eqn.(14);
1 **while** *not converged yet* **do**
    `/* Updating parameters */`
    for $s=1$ *to* M **do**
      for $l=1$ *to* 3 **do**
        update $U^{s,l}$ following Eqn.(12);
      update $\mathcal{G}^s$ following Eqn.(13);
  `/* Calculation of Inconsistency Score list` I `*/`
  $\mathcal{G}^*=\frac{1}{M}\sum_{s=1}^{M}\mathcal{G}^s$;
  for $k=1$ *to* K **do**
    for $s=1$ *to* M **do**
      $S(s)$ is the cosine similarity between $\mathcal{G}^s$ and $\mathcal{G}^*$;
    $I(k)=\mathrm{Var}(S)$

In this section, we provide an efficient initialization and give the complete algorithm in Algorithm 1.

Initialization

Since initialization plays an important role in the algorithm, it is important to set a proper starting point for optimization. Denote by $X_k\in\mathbb{R}^{N\times T}$ the original observed data for k-th host. We apply the basic clustering approach, k-means, on X_k and then achieve the final clustering index by via majority voting. Thus,

$$U_{ij}^{s,l}=\arg_{x\in\{0,1\}}\max_{k=1,\cdots,K}\{\#(u_{ij}^{k,l}=x)\}, \quad (14)$$

where $u_{ij}^{k,l}=1$ means that the i-th object belonging to the j-th cluster, otherwise, it is zero. More specifically,

- $u^{k,1}\in\mathbb{R}^{N\times C_N}$, represents the results of the K-means clustering algorithm on X_k treating its columns as attributes.
- $u^{k,2}\in\mathbb{R}^{K\times K}$ is identity matrix.
- $u^{k,3}\in\mathbb{R}^{T\times C_T}$ represents the results of the K-means clustering algorithm on X_k treating its rows as attributes.

Algorithm

Combining everything together, Algorithm 1 sketches the procedure of model inference. After proper initialization, it iterates between updating $U^{s,l}$ and $\mathcal{G}^s$ until the objective function converges.

4. EXPERIMENTAL ANALYSIS

In the previous section, we propose the framework,TMVID, to detect inconsistent hosts across multiple views. Now we present an empirical evaluation of the proposed framework via a set of experiments: (1) Compared with baseline methods, we demonstrate the effectiveness of TMVID on several synthetic data sets. We also show that the proposed framework is scalable to large-scale data set by conducting efficiency tests. (2) The advantage of TMVID in detecting inconsistent hosts is further demonstrated on two real data sets related to network flow traffic and domain name systems which are collected from IBM enterprise networks.

4.1 Synthetic Data

We begin with introducing the synthetic data sets and the baselines to evaluate. Experiments are conducted on these data sets to show the advantage of the proposed framework over baseline methods.

Description. We simulate the detector scores of hosts in cyber network scenarios. Data from 3 views are generated with 5 latent detector clusters and 7 timestamp groups. We assume detector scores follow Gaussian distributions with specific variances which depend on detectors clusters and time groups simultaneously. Besides, detectors from different clusters have different mean. We randomly select n hosts and generate detector scores using Gaussian distributions with different parameters. Thus, these n hosts can be regarded as inconsistent object, as their detector scores are inconsistent across different views. We expect that a good framework can find these inconsistent hosts. To better demonstrate the effectiveness, we vary the characteristics of the data set, such as the number of detectors, hosts, and timestamp of the synthetic data. The statistics of synthetic data sets are described in Table 1.

Table 1: Statistics of Synthetic Data Sets

	# of detectors	# of hosts	# of timestamp	# of views
SYNTH-1	50	1000	200	3
SYNTH-2	50	2000	300	10
SYNTH-3	100	5000	500	50

Evaluation. Denote by the inconsistent hosts positive samples, and others by negative samples, we can deduce the true positive, true negative, false positive and false negative rates, from which F_1 score is calculated. The higher the F_1 score is, the better the proposed method.

Baselines. We compare the experimental results of methods to show the advantage of the proposed joint probabilistic tensor model. The first baseline method is joint nonnegative matrix factorization (NMF) [11] which does not consider time evolution. It partitions several detectors into latent detector clusters by conducting joint nonnegative matrix factorization. We average the inconsistency score obtained by conducting NMF on each timestamp as the final result. Another popular method is majority voting. More specifically, if the majority of views claim an hosts as malicious, it is labeled as malicious. Otherwise it is labeled as noraml. There are three basic operations to handle static data on each snapshot, such as mean, minimization, and maximization. Therefore, we have VOTE/MEAN,VOTE/MIN, and VOTE/MAX. Also, instead of applying majority voting across views, we can take mean, minimization, and maximization inside each view as well as across views. Therefore, we have three more baselines: MEAN,MIN, and MAX.

Table 2: F-1 Score Comparison

	VOTE /MEAN	VOTE /MIN	VOTE /MAX	MEAN	MIN	MAX	NMF	TMSID
SYNTH-1	.81	.85	.80	.10	.15	.10	.80	1.0
SYNTH-2	.86	.92	.85	.20	.21	.20	.82	1.0
SYNTH-3	.90	.95	.85	.25	.30	.28	.87	1.0

As shown in Table 2, the proposed method has the highest F_1 score, as it considers the information across views and over time simultaneously. For MEAN,MIN and MAX, all of them have a very low F_1 score. The reason is that they do not consider the temporal pattern over time and simply take mean (minimization or maximization) over time and across views. For majority voting approach, it also has comparatively higher F_1 score, as they extract more reliable information across multiple views. Compared with those baseline methods, we demonstrates that the proposed framework works well on detecting inconsistent hosts.

Scalability

We evaluate the scalability of TMVID on synthetic data sets. Specifically, for model inference, we measure the average execution per task by TMVID while varying the number of hosts and views. Different data sets with different number of hosts are generated. As shown in Table 3, the proposed algorithm has almost linear complexity in the number of hosts. As for the baseline NMF, although its time complexity is also nearly linear, much more time is needed when compared to the proposed algorithm. As shown in Figure 4, the proposed algorithm has linear complexity with respect to the number of views. Based on the results on the synthetic data sets, it is safe to conclude that the proposed algorithm can scale up to large data sets.

Table 3: Running Time

Joint Probabilistic Tensor Factorization		Nonnegative Matrix Factorization	
# Hosts	Time (s)	# Hosts	Time (s)
1.0×10^2	1.3	1.0×10^2	489.3
1.0×10^3	10.5	1.0×10^3	601.1
5.0×10^3	50.8	5.0×10^3	704.2
1.0×10^4	107.8	1.0×10^4	1342.9
5.0×10^4	533.4	5.0×10^4	6324.0
1.0×10^5	1107.5	1.0×10^5	12822.0
Pearson Correlation	0.9998	Pearson Correlation	0.9992

4.2 Real Data

In this section, we show experimental results on two real data sets collected from IBM enterprise netowrks, Network

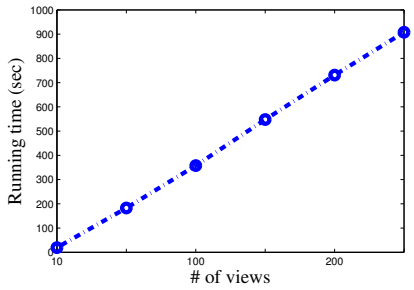


Figure 4: Average execution time with increasing number of views by Joint Probabilistic Tensor Factorization in model inference

Flow Traffic (NETFLOW) and Domain Names System Data (DNS). The statistics of the real data sets are shown in Table 4. The goal is to detect inconsistent hosts which behave inconsistently across different views. More details about the data are introduced in the following sections.

Table 4: Statistics of Real-World Data Sets

	# of detectors	# of hosts	# of timestamp	# of views
NETFLOW	50	500	500	4
DNS	50	500	500	5

Network Flow Traffic Data

Network flow traffic data, which is collected from IBM enterprise networks, consists of 4 views such as TCP incoming/outcoming traffic and UDP incoming/outcoming traffic for 500 hosts over 18 months. Each view contains 3 attributes: the number of bytes, flows, and packets. We first conduct 50 anomaly detectors based on these attributes on each day. Through joint probabilistic tensor factorization, the proposed framework TMVID projects the observed data into a latent subspace, where both detector and timestamp dimension are grouped simultaneously. Detector-Cluster (Timestamp-Cluster) means that the detector scores of detector (timestamp) clusters are considered, while Bi-Cluster means that we take both detector and timestamp clusters into consideration. From Bi-Cluster's perspective, we can compare hosts' behavior on the detector clusters and timestamp clusters simultaneously. Besides, we can compare their behavior on detector or timestamp cluster respectively.

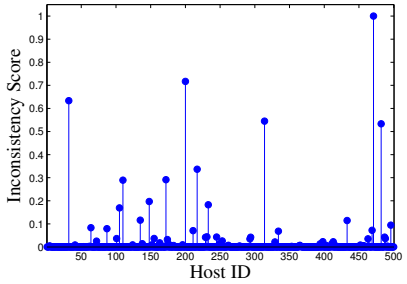


Figure 5: Inconsistent scores for host ID in network

We plot the hosts' inconsistency score in Figure 5, where x axis represents hosts' ID and y axis stands for the inconsistency score. As shown in Figure 5, most of hosts behave consistently, while only very few hosts are abnormal, whose inconsistency scores are much higher compared to the rest of hosts. To confirm that the abnormal hosts indeed behave inconsistently across multiple views, we conduct case study from three perspectives: Bi-Cluster, Detector-Cluster and Timestamp-Cluster perspectives.

Bi-Cluster Perspective. Semantically, joint probabilistic tensor model maps the original tensor into a unknown latent tensor subspace. $(U^{s,1})^T X_k^s U^{s,3}$ is an unique representation of X_k^s in the latent tensor subspace. The benefit of the proposed framework is that the inconsistent and consistent hosts can be well separated in the new subspace, which is confirmed by Figure 6 where x, y, and z axes refer to time cluster ID, detector cluster ID, and detector scores, respectively. In the figure, Figure 6(a) and 6(b) represent the top two inconsistent hosts found by TMVID, respectively. Figure 6(c) and 6(d) stand for the top two consistent hosts. For the inconsistent hosts, the bi-clusters' behavior is well separated in the subspace found by the probabilistic tensor factorization technique, while the behavior of consistent hosts is almost the same across views.

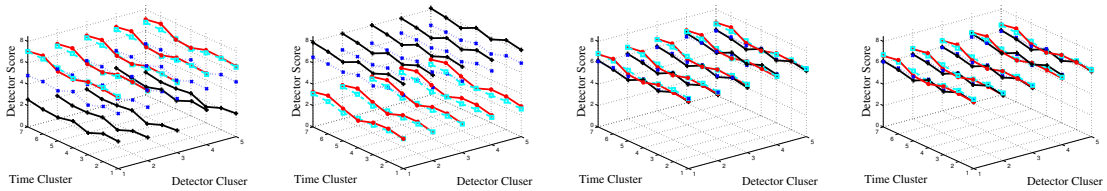
The results from NMF are shown in Figure 7. Figure 7(a) and 7(b) refer to the top two inconsistent hosts, while 7(c) and 7(d) are consistent hosts. From it, we can see that the patterns of both inconsistent and consistent hosts from multiple views are similar in the latent subspace. Thus, it is hard to separate inconsistent and consistent hosts by NMF.

Detector-Cluster Perspective. Note that the i-th row of $U^{s,1}$ represents the detector cluster distribution for the i-th detector. Namely, $U_{ij}^{s,1}$ means how likely it is that the i-th detector belongs to the j-th detector clusters. The j-th row of $(U^{s,1})^T X_k^s$ represents the expected detector score from j-th detector cluster for k-th host on s-th views over times. Therefore, we expect that for those inconsistent hosts found by the algorithms, at least one view's detector clusters' behavior is significantly different from other views. It is confirmed by Figure 8(a) and 8(b) where x and y axes refer to day ID and detector score respectively. We can see that for inconsistent hosts as shown in Figure 8(a), the patterns of detector clusters' are quite different across views, while the patterns are similar for consistent hosts (Figure 8(b)) across views, ignoring noise affected by randomly factors.

Timestamp-Cluster Perspective. $U^{s,3}$ describes the timestamp cluster distribution matrix. More specifically, $U_{ij}^{s,3}$ measures the probability that the i-th day shall belong to the j-th timestamp cluster. Moreover, the j-th row of $X_k^s U^{s,3}$ relates to the expected detector score from the j-th timestamp cluster for the k-th host on the s-th view over all individual detectors. It is confirmed by Figure 8(c) and 8(d) in which x axis represents detector ID and y axis refers to detector score separately. For the inconsistent hosts in Figure 8(c), the patterns of timestamp clusters' vary a lot across views, especially for view 2 and view 4, whose patterns are clearly different from that of view 1 and view 3. However,the patterns are quite similar for consistent hosts in Figure 8(d), which show the similar increasing trend.

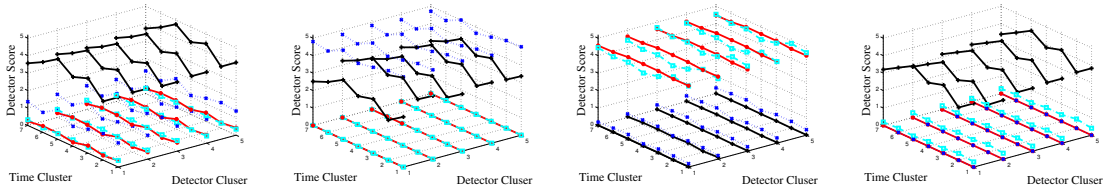
Domain Name System Data

In this experiment, we evaluate the proposed framework, TMVID, on the Domain Name System (DNS) data. The DNS data, which contains the number of URL requests related to Sciences, Arts, Shopping, Health, Sport, and others,is also collected from IBM enterprise networks. We also conduct 50



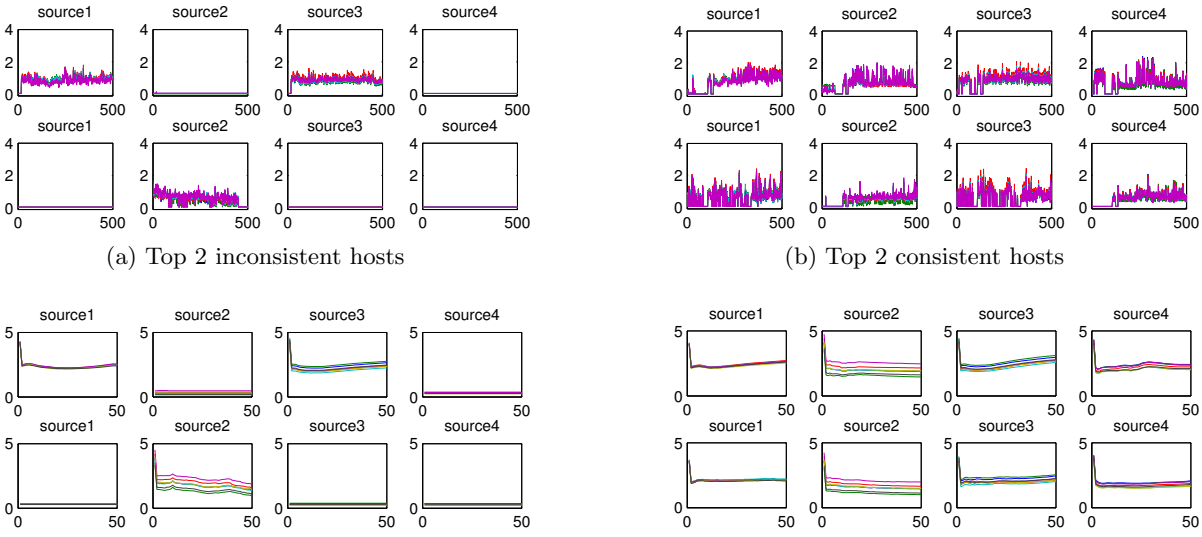
(a) 1^{st} inconsistent host (b) 2^{nd} inconsistent host (c) 2^{nd} consistent host (d) 1^{st} consistent host

Figure 6: Comparison of top two inconsistent and consistent hosts in network flow traffic data: detector score over both detector and time clusters



(a) 1^{st} inconsistent host (b) 2^{nd} inconsistent host (c) 2^{nd} consistent host (d) 1^{st} consistent host

Figure 7: NMF, Comparison between top two inconsistent and consistent hosts in network flow traffic data: Detector scores over both detector and time clusters



(a) Top 2 inconsistent hosts (b) Top 2 consistent hosts

(c) Top 2 inconsistent hosts (d) Top 2 consistent hosts

Figure 8: Comparison between top two inconsistent and consistent hosts in network flow traffic data. In all subfigures, y axis refers to detector score. In (a) and (b), x axis represents day ID while it stands for the detector ID in (c) and (d)

detectors and form the detector scores into 5 tensors according to the five URL request categories on each day. Based on the purpose of hosts' internet access, we partition the URL requests into five categories: Sciences, Arts, Shopping, Health, and Sport. In this application, each URL request category is regarded as a view. Namely, there are totally five views in DNS data.

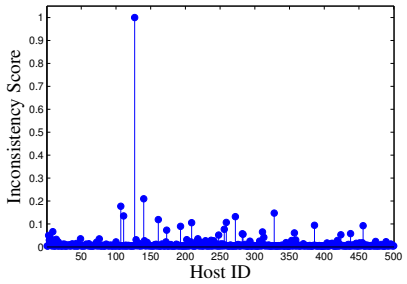


Figure 9: Inconsistency scores for host ID in DNS

We plot the inconsistency score distribution of 500 hosts obtained from TMVID on DNS data in Figure 9 where x and y axes refer to hosts' ID and inconsistency scores respectively. Figure 9 shows that most of the hosts are considered as consistent hosts while only a few hosts are concluded as inconsistent ones. This observation corresponds to our assumption that the majority of hosts behave consistently. Besides, the most consistent host found by our method is actually the primary server. This phenomenon is very interesting and realistic. The primary server processes numerous requests about various categories, making its behavior pattern consistent. This result furthermore confirms the accuracy of the proposed framework TMVID.

Similar to the previous experiments, we also conduct case studies on the top two inconsistent and consistent hosts to demonstrate that our method can well separate them. The results are shown in Figure 10. Figure 10(a) and 10(b) refer to the top two inconsistent hosts; 10(c) and 10(d) show the patterns of top two consistent hosts. The results from NMF are shown in Figure 11 from which we cannot separate inconsistent hosts from consistent ones. Thus, it furthermore confirms that NMF may not work well on detecting inconsistent hosts.

5. RELATED WORK

Anomaly Detection in Cyber Network

Anomaly detection [1, 6, 22] has become an important research topic in many domains over last decades. Local outlier factor (LOF) [4] is a popular detection algorithm for separating the outlier from normal hosts by measuring how isolated the object is with respect to the surrounding neighborhood. Many extensions based on LOF are also proposed, such as [4,16]. However, the limitation of existing approaches is that they only consider one view of hosts. To overcome this challenge, the proposed method can consider multiple views of hosts, which can output more trustable information. Some work [8] focuses on supervised learning approach to detect malicious hosts. The limitation of supervised learning is that they need labels which are time-consuming and money-consuming to obtain. To overcome this difficulty brought by the lack of labels, we introduce an unsupervised approach to detect hosts' inconsistency across views and over time.

Ensemble analysis [2, 9] has also been proposed in outlier detection. By incorporating various outlier algorithms, the authors show that the outlier ensemble learning approach is able to extract more reliable information. However, the proposed framework can go further. TMVID considers the combination of a various of anomaly detection and multiple views of an entity simultaneously.

Network anomaly detection has become an essential research field. In [5, 14], network anomaly defines abnormal behavior of network as the presence of an intruder or network flow traffic overload. Different from this definition, the proposed framework aims to find these abnormal hosts whose behavior is inconsistent across multiple views.

Tensor Factorization

Tensor factorization approaches [3, 7, 21, 25, 26, 29] have become popular in the data mining, as it maps from a high dimensional data space into a low dimension space. Probabilistic tensor factorization [24, 30] has attracted lots of attention during the last decade because of its ability of mapping the observed tensors into latent tensors. However, most of traditional tensor factorization methods simply work on one tensor from one single view. The proposed approach, however, stresses the importance of extracting consistency patterns from multiple views by constraining the projection matrices to be similar across views.

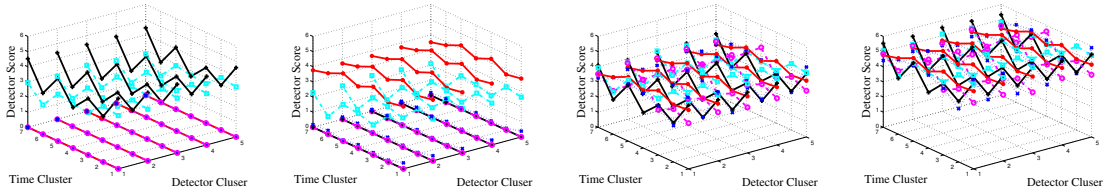
Another related technique is the matrix factorization proposed in [10, 11, 15, 19, 28]. The limitation of those works is that they fail to consider the temporal behavior of hosts over time. As a consequence, important temporal information is missing. The proposed work is able to cluster the timestamp such that days that share with similar behavior will be clustered together for further analysis.

Temporal Anomaly Detection

The importance of modeling temporal behavior has been realized by researchers [13,18,27,31] when conducting anomaly detection in network traffic or other domains. Suspicious objects are detected when their temporal behavior is different from their historical records or deviate from the pattern of normal objects. The framework TMVID we proposed calculates the inconsistent scores for each object by comparing the temporal patterns on timestamp and detector clusters from multiple views simultaneously.

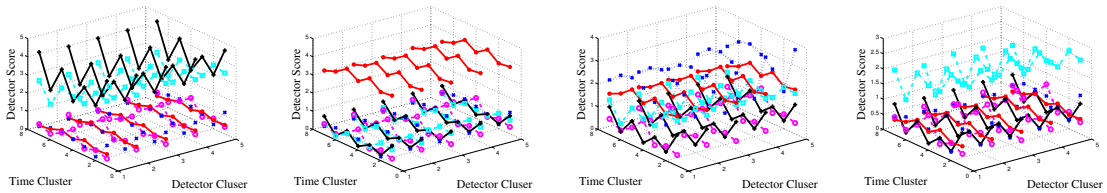
6. CONCLUSIONS

This paper presents a novel framework called TMVID to conduct inconsistency detection from multiple views of temporal data. Based on the raw data collected from multiple views, we first apply anomaly detection algorithms and obtain anomalous scores of hosts. The behavior of hosts can thus be summarized in a three-way tensor which consists of the anomalous scores for each host at each time snapshot by each detector. To extract common behavior hidden in multiple views, we propose a joint probabilistic tensor factorization approach to factorize the observed tensors together so that the projection matrices are similar across views. Inconsistency scores of hosts are then calculated by measuring the deviation from the mean latent tensor. We demonstrate the efficacy of TMVID to capture inconsistencies in multi-view



(a) 1^{st} inconsistent host (b) 2^{nd} inconsistent host (c) 2^{nd} consistent host (d) 1^{st} inconsistent host

Figure 10: Comparison of top two inconsistent and consistent hosts in DNS data: detector score over both detector and time clusters



(a) 1^{st} inconsistent host (b) 2^{nd} inconsistent host (c) 2^{nd} consistent host (d) 1^{st} consistent host

Figure 11: NMF, Comparison between top two inconsistent and consistent hosts in DNS data: Detector scores over both detector and time clusters

temporal data on synthetic data and two network traffic data sets. Results show that consistent and inconsistent hosts can be well separated in the new space that is obtained by the joint tensor factorization. By outperforming baseline approaches, the proposed approach demonstrates its effectiveness in cybersecurity applications.

APPENDIX

Surrogate Function.

First, we aim to find the relationship between the objective function $\mathcal{L}_\Lambda(U^{s,l}|\Theta)$ and its surrogate function $Q_1(U^{s,l}|\Theta;\Theta_n)$ as defined in Eqn.(10). According to the definition of $\mathcal{L}_\Lambda(U^{s,l}|\Theta)$, we have:

$$\begin{aligned}
\mathcal{L}_\Lambda(U^{s,l}|\Theta) &\propto \sum_{s=1}^{M}\left[\frac{1}{2}\|\mathcal{X}^s - \mathcal{G}^s\Pi_{\times d}U^{s,d}\|_F^2 + \frac{\lambda_l}{2}\|U^{s,l}-U^{*,l}\|_F^2\right]\\
&\propto \sum_{s=1}^{M}\left[\frac{1}{2}\|X^s_{(l)} - U^{s,l}A_l^s\|_F^2 + \frac{\lambda_l}{2}\|U^{s,l}-U^{*,l}\|_F^2\right]\\
&\propto \sum_{s=1}^{M}\frac{1}{2}\left[\sum_{ij}\left(U^{s,l}A_l^s(A_l^s)^T + \lambda_l U^{s,l}\right)_{ij}(U^{s,l})^T_{ij}\right.\\
&\quad \left. -2\sum_{ij}\left(X_{(l)}(A_l^s)^T + \lambda_l U^{s,l}\right)_{ij}(U^{s,l})^T_{ij}\right]\\
&\leq \sum_{s=1}^{M}\left[\sum_{i,j}\frac{\left[U_n^{s,l}(A_l^s(A_l^s)^T + \lambda_l I_l)\right]_{ij}(U_{ij}^{s,l})^2}{2{U_n^{s,l}}_{ij}}\right.\\
&\quad -2\sum_{i,j}{U_n^{s,l}}_{ij}\left[X^s_{(l)}(A_l^s)^T\right]_{ij}(1+\log\frac{U_{ij}^{s,l}}{{U_n^{s,l}}_{ij}})\\
&\quad \left. -2\lambda_l\sum_{i,j}{U_n^{s,l}}_{ij}U_{ij}^{s,l}(1+\log\frac{U_{ij}^{s,l}}{{U_n^{s,l}}_{ij}})\right]\\
&= Q_1(U^{s,l}|\Theta;\Theta_n).
\end{aligned}$$

In our proof, we replace the related term with its upper bound, when $U^{s,l} = U_n^{s,l}$ the equality is achieved. Therefore, we have $Q_1(U^{s,l}|\Theta_n;\Theta_n) = \mathcal{L}_\Lambda(U^{s,l}|\Theta_n)$. So far, we have proved the desired property of $Q_1(U^{s,l}|\Theta;\Theta_n)$. Namely,

$$\begin{cases} Q_1(U^{s,l}|\Theta;\Theta_n) \geq \mathcal{L}_\Lambda(U^{s,l}|\Theta), & \forall\Theta,\Theta_n;\\ Q_1(U^{s,l}|\Theta;\Theta_n) = \mathcal{L}_\Lambda(U^{s,l}|\Theta_n), & \forall\Theta_n. \end{cases}$$

Similarly, for $Q_2(\mathcal{G}^s|\Theta;\Theta_n)$, we have:

$$\begin{aligned}
\mathcal{L}_\Lambda(\mathcal{G}^s|\Theta_n) &\propto \sum_{s=1}^{M}\left[\frac{1}{2}\|\mathcal{X}^s - \mathcal{G}^s\Pi_{\times d}U^{s,d}\|_F^2\right]\\
&\propto \sum_{s=1}^{M}\left[\frac{1}{2}\|vec(\mathcal{X}^s) - U^s vec(\mathcal{G}^s)\|_F^2\right]\\
&\propto \sum_{s=1}^{M}\frac{1}{2}\left[\sum_k\left(vec(\mathcal{G}^s)U^s(U^s)^T\right)_{ij}vec(\mathcal{G}^s)_k^T\right.\\
&\quad \left. -2\sum_k U^s vec(\mathcal{X}^s)^T\right]\\
&\leq \sum_{s=1}^{M}\left[\sum_k\frac{\left[vec(\mathcal{G}_n^s)U^s(U^s)^T\right]_k vec(\mathcal{G}^s)_k^2}{2vec(\mathcal{G}^s)}\right.\\
&\quad -2\sum_k vec(\mathcal{G}_n^s)_k vec(\mathcal{X}^s)_k(U^s)^T\\
&\quad \left.\left(1+\log\frac{vec(\mathcal{G}^s)_k}{vec(\mathcal{G}_n^s)_k}\right)\right]\\
&= Q_2(\mathcal{G}^s|\Theta;\Theta_n).
\end{aligned}$$

Provided that $\mathcal{G}_n^s = \mathcal{G}^s$, we have that $Q_2(\mathcal{G}^s|\Theta_n;\Theta_n) = \mathcal{L}_\Lambda(\mathcal{G}^s|\Theta_n)$, which proves the desired property of $Q_2(\mathcal{G}^s|\Theta_n;\Theta_n)$, that is,

$$\begin{cases} Q_2(\mathcal{G}^s|\Theta;\Theta_n) \geq \mathcal{L}_\Lambda(\mathcal{G}^s|\Theta), & \forall\Theta,\Theta_n;\\ Q_2(\mathcal{G}^s|\Theta;\Theta_n) = \mathcal{L}_\Lambda(\mathcal{G}^s|\Theta_n), & \forall\Theta_n. \end{cases}$$

7. REFERENCES

[1] D. Agarwal. An empirical bayes approach to detect anomalies in dynamic multidimensional arrays. In *Proc. of the IEEE International Conference on Data Mining*, 2005.

[2] C. C. Aggarwal. Outlier ensembles: Position paper. *ACM SIGKDD Explorations Newsletter*, 2013.

[3] A. Bernardi, J. Brachat, P. Comon, and B. Mourrain. General tensor decomposition, moment matrices and applications. *Journal of Symbolic Computation*, 2013.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, volume 29, 2000.

[5] C. A. Catania, F. Bromberg, and C. G. Garino. An autonomous labeling approach to support vector machines algorithms. for network traffic anomaly detection. *Expert Systems with Applications*, 2012.

[6] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 2009.

[7] S. Chang, G.-J. Qi, C. C. Aggarwal, J. Zhou, M. Wang, and T. S. Huang. Factorized similarity learning in networks. In *Proc. of the IEEE International Conference on Data Mining*, 2014.

[8] S. R. Gaddam, V. V. Phoha, and K. S. Balagani. K-means+ id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods. *IEEE Transactions on Knowledge and Data Engineering*, 2007.

[9] J. Gao, W. Fan, D. Turaga, O. Verscheure, X. Meng, L. Su, and J. Han. Consensus extraction from heterogeneous detectors to improve performance over network traffic anomaly detection. In *Proc. of the IEEE International Conference on Computer Communications Mini-Conference*, 2011.

[10] L. Ge, J. Gao, X. Li, and A. Zhang. Multi-source deep learning for information trustworthiness estimation. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[11] L. Ge, J. Gao, X. Yu, W. Fan, and A. Zhang. Estimating local information trustworthiness via multi-source joint matrix factorization. In *Proc. of the IEEE International Conference on Data Mining*, 2012.

[12] J. Han and J. Z. Zhang. Network traffic anomaly detection using weighted self-similarity based on emd. In *Proc, of the IEEE SoutheastCon*, 2013.

[13] D. J. Hill and B. S. Minsker. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling and Software*, 2010.

[14] H. Huang, H. Al-Azzawi, and H. Brani. Network traffic anomaly detection. *arXiv preprint arXiv:1402.0856*, 2014.

[15] X. Jia, N. Du, J. Gao, and A. Zhang. Analysis on community variational trend in dynamic networks. In *Proc. of the ACM International Conference on Conference on Information and Knowledge Management*, 2014.

[16] W. Jin, A. K. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001.

[17] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 2009.

[18] T. Lane and C. E. Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information and System Security*, 1999.

[19] D. Liu, C.-H. Lung, I. Lambadaris, and N. Seddigh. Network traffic anomaly detection using clustering techniques and performance comparison. In *Proc. of the IEEE Canadian Conference on Electrical and Computer Engineering*, 2013.

[20] M. P. Mackrell, K. J. Twilley, W. P. Kirk, L. Q. Lu, J. L. Underhill, and L. E. Barnes. Discovering anomalous patterns in network traffic data during crisis events. In *Proc. of the IEEE Systems and Information Engineering Design Symposium*, 2013.

[21] E. Papalexakis, U. Kang, C. Faloutsos, N. Sidiropoulos, and A. Harpale. Large scale tensor decompositions: Algorithmic developments and applications. *IEEE Data Engineering Bulletin*, 2013.

[22] M. Roughan, T. Griffin, Z. M. Mao, A. Greenberg, and B. Freeman. Ip forwarding anomalies and improving their detection using multiple data sources. In *Proc. of the ACM SIGCOMM Workshop on Network Troubleshooting*, 2004.

[23] D. B. Roy and R. Chaki. State of the art analysis of network traffic anomaly detection. In *Proc. of the IEEE Applications and Innovations in Mobile Computing Conference*, 2014.

[24] M. Schmidt and S. Mohamed. Probabilistic non-negative tensor factorisation using markov chain monte carlo. In *Proc. of the European Signal Processing Conference*, 2009.

[25] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proc. of the ACM International Conference on Machine Learning*, 2005.

[26] J. Sun, S. Papadimitriou, and S. Y. Philip. *Tensor Analysis on Multi-aspect Streams*. Springer, 2007.

[27] A. Wagner and B. Plattner. Entropy based worm and anomaly detection in fast ip networks. In *Proc. of the IEEE International Workshops on Enabling Technologies*, 2005.

[28] F. Wang, T. Li, and C. Zhang. Semi-supervised clustering via matrix factorization. In *Proc. of the SIAM International Conference on Data Mining*, 2008.

[29] H. Xiao, Y. Li, J. Gao, F. Wang, L. Ge, W. Fan, L. Vu, and D. Turaga. Believe it today or tomorrow? detecting untrustworthy information from dynamic multi-source data. In *Proc. of the SIAM International Conference on Data Mining*, 2015.

[30] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proc. of the SIAM International Conference on Data Mining*, 2010.

[31] N. Ye et al. A markov chain model of temporal behavior for anomaly detection. In *Proc. of the IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, 2000.