PNNL-24901

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

UQ Methods for HPDA and Cybersecurity Models, Data, and Use Cases

DW Engel
KD Jarman
Z Xu
B Zheng
AM Tartakovsky
X Yang
R Tipireddy
H Lei
J Yin

November 2015

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

UQ Methods for HPDA and Cybersecurity Models, Data, and Use Cases

DW Engel
KD Jarman
Z Xu
B Zheng
AM Tartakovsky
X Yang
R Tipireddy
H Lei
J. Yin

November 2015

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@ntis.fedworld.gov
online ordering: http://www.ntis.gov/ordering.htm

Contents

1.0 Introduction

Effective cyber defense requires rigorous solutions to the following fundamental and crosscutting research challenges:

- **Characterizing adversaries, s**uch as characterizing botnets as a class or classes of adversaries.
- **Characterizing cyber attacks, s**uch as characterizing commonalities and differences in the attacks on Sony and Anthem.
- **Expressing confidence in characterization and data, s**uch as providing the technical basis for determining low, medium, or high level of confidence or importance of given data.

A large number of models have been developed to simulate actions over the Internet. However, the current modeling of uncertainties associated with these models and data is very limited, and is typically applied to a very specific type of problem.

> *"The research for systematically handling uncertainty and risk in cyber space is still in a preliminary stage. However, without addressing this problem in a scientific manner, it will be difficult to achieve sustainable cyber defense."* [Li, 2013]

This report describes our initial research to quantify uncertainties in the identification and characterization of possible attack states in a network. The result of this work should ultimately enable estimates of the current state in which the network is operating, based on a wide variety of network data, along with a defensible measure of confidence to this state estimate. The output of this research will be new uncertainty quantification (UQ) methods to help:

- Develop a process for model development and apply UQ to characterize attacks/adversaries.
- Understand the degree to which methods scale to "big" data.
- Offer methods for addressing model approaches with regard to validation and accuracy.

In our research, we study a complementary suite of UQ approaches aimed at demonstrating how uncertainty quantification can be incorporated into existing cybersecurity anomaly detection and classification methodologies. The general scope of our work is based on the core processes of cybersecurity data estimation and classification, and the need to propagate uncertainty in the data through these two processes to a measure of confidence in classification or characterization of the state of a network that is reflected by the data.

Explicitly, we view characterization of "states" of the network as a two-step process:

1. Estimate a network state vector as a function of time. For example, we estimate a traffic matrix (a representation of volume of flow between routers, IPs, ports, or other network IDs) or corresponding graph, or a set of summary statistics such as information entropy, based on NetFlow data and a statistical model that captures the uncertainty in those data.
2. Classify the network state, based on prior training data containing ground truth, as being a particular type of network event/attack/anomaly (or unknown anomaly). Ideally, the outcome is not merely a decision as to the current state class, but a measure of confidence

in that decision, or better yet, a measure of likelihood or probability of being in each of the known or unknown state classes.

We are developing a state-space model for UQ in dynamic state estimation. This approach intends to implement a standard time-series model, such as the Kalman Filter (KF), to summary statistics on NetFlow data such as information entropy as they change in time. The KF is a Bayesian approach that estimates both the true state and a measure of uncertainty on the state based on collected data and a model of the underlying uncertainty in those data. Here we focus on the uncertainty that arises from the fact that NetFlow data may be incompletely sampled (missing data or sampling rates), and the data may be "corrupted" in certain ways, such as having inaccurate time stamps. The estimate of uncertainty produced by the KF can then feed into estimates of uncertainty/confidence in classification.

In classification, we are studying the quality of machine-learning-based classification as a function of uncertainty due to data sampling, potentially mislabeled training data, and the choice of algorithm parameter values in classification schemes, and other sources of uncertainty as they are identified. To make our study concrete, we begin our study with support vector machines (SVM), a popular machine-learning approach in many domains including cybersecurity, and consider the impact of these sources of uncertainty on classification boundaries, and how those propagate to confidence in classification.

Concurrently, we are working to extend network-traffic graph-based clustering to probabilistic traffic graphs that capture uncertainty in data collection and processing. Our goal is to develop stochastic graph-based clustering as a basis for classification with measures of confidence. Graphs generated from traffic matrices, for example, are already a popular tool for clustering to identify and characterize anomalous network behavior. Incorporating uncertainty in nodes and edges (e.g., existence, or weights) poses a challenge for clustering, especially in computational complexity. We will take advantage of well-established techniques such as the Karhunen-Loève (K-L) stochastic expansion to develop efficient spectral clustering algorithms that propagate data uncertainty to clustering. This approach can be viewed as both an alternative and complement to our state-space modeling and classification studies.

Each approach (UQ in state-space modeling, machine learning, and graph models) tackles a piece of the problem of propagating uncertainty to measures of confidence in state classification. We anticipate that the component methods can either be directly combined (e.g., KF-based state vector estimation with uncertainty estimate can be input to SVM-based classification with uncertainty) and/or benefit from concurrent development (e.g., the efficiency gained via K-L expansions in stochastic graph-based clustering might be applied in a similar way to represent uncertainty in the other approaches).

The specific algorithms for modeling cyber attacks that we are currently adapting are discussed in Section 2 of this document. In Section 3, we discuss the UQ development for each of the cyber models. Data and use-case scenarios for testing our models and UQ methods are discussed in Section 4, followed by an overall research discussion in Section 5.

2.0 Cyber Modeling

The objective of our research is to develop methods and tools to comprehensively quantify the different uncertainties in cyber systems and identify sources in need of additional modeling and/or data collection. To achieve this, we are applying advanced uncertainty quantification techniques to three different types of models (i.e., state-space models, graph models, and machine learning) that are currently used within the cybersecurity realm. In this section, we discuss our selection process for these three modeling areas; we also document the current state of the models, and our implementation and enhancement to the models.

2.1 State Space Model

We started our research with the study of existing methods in the literature for state-space models for Internet traffic data. We found that a promising methodology used in the literature is to represent the data as a traffic matrix. Each element in the traffic matrix represents amount (as number of bytes or packets) of traffic flowing from the source to the destination. Traffic matrices are usually constructed from Simple Network Management Protocol (SNMP) link data and/or NetFlow data. Since it is impractical to measure the traffic data for the entire network in real time due to storage and speed issues, it is usually sampled at certain time intervals. Traffic matrices are useful for network optimization, traffic design, protocol design, and anomaly detection. We studied various methods and models in the literature to estimate the traffic matrix as temporal, spatial and spatio-temporal [Tune, 2013] from the sampled traffic data. We also obtained synthesized NetFlow data from the 2013 VAST mini challenge-3. We will use this data to build and test UQ methods within state-space models for detection, identification, and characterization of cyber attacks during the duration of the data collected.

2.1.1 Modeling NetFlow data as traffic matrix (current methods)

The relation between the traffic matrix, the routing, and the link counts is $y = Ax$, where y is a vector of link counts, x is the traffic matrix organized as a vector, A is a routing matrix in which element A_{ij} is equal to 1 if OD (origin-destination) pair j traverses link i or zero otherwise. Link counts y can be obtained by standard SNMP measurements and the routing matrix A can be obtained by computing shortest paths using interior gateway protocol (IGP) link weights together with the network topology information. Then the problem at hand is to estimate the traffic matrix x given, y and A. The following methods are discussed in the remaining of this section: [Soule, Lakhina, et.al., 2005; Soule, Salamantian, Nucci, et.al., 2005; Soule, Salamantian, and Taft, 2005; Tune, 2013]

- second generation methods
  - tomogravity method
  - route change method
- third generation methods (uses partial flow measurements)
  - fanout method
  - PCA method
  - Kalman method

Notation

- x is a traffic matrix at a specific point in time represented as an Nx1 column vector,

- *y* is the column vector of *L* links at any point in time,
- *A* is the LxN routing matrix,
- *X* is the traffic matrix over time and is a TxN matrix where each column *j* corresponds to the time series of OD flow *j*,
- *Y* is a TxL multivariate time series of link traffic.

Tomogravity method
Let *x(i,*)* be the total traffic entering an ingress node *i* and *x(*,j)* be the total traffic departing from node *j*. Then the gravity model is

$$x(i,j) = x(i,*)\frac{x(*,j)}{\sum_j x(*,j)}$$

which implies that the total amount of data node *i* sends to node *j* is proportional to the amount of traffic departing the network at *j* relative to the total amount of traffic departing the entire network.

Route change method
The OD flow model is

$$x(i,j,t) = \sum_h \theta_h\,(i,j) b_h(t) + w(i,j,t)$$

where the first term is the Fourier expansion for the diurnal trends and the second term captures the stationary fluctuations.

Fanout method
This is purely a data-driven method that relies on measurements alone to obtain the traffic matrix

$$f(i,j,t) = \frac{x(i,j,t)}{\sum_j x(i,j,t)}$$

is the fraction of the traffic entering node *i* that will egress the network at node *j* at time *t*. The traffic matrix estimate is

$$\hat{x}(i,j,t) = \hat{f}(i,j,t)x(i,*,t)$$

where, *x(i,*, t)* is the total incoming traffic into node *i*.

Principal components method
Principal component analysis (PCA) is a dimension reduction technique that captures the maximum energy (or variability) in the data onto a minimum set of new axes called principal components. Write PCA of *X* as

$$X = USV^T$$

Select the top *k* principal components,

$$x_t = V'S'u_t', \qquad t = 1, \dots, T$$

and the link measurement is

$$y_t = AV'S'u_t', \qquad t = 1, \dots, T$$

Here, the first u_t' is computed using pseudo-inverse and is substituted to estimate the traffic demand $\hat{x}_t$. In order to obtain decomposition $X = USV^T$, we need the traffic matrix X. This is a prior traffic matrix obtained by using Netflow data. Like the fanout method, the PCA method also has a recalibration step.

Kalman Filter method

Let y_t be an observation vector at discrete time t and let $Y_t = \{y_t\}$ be the set of all observations up to time t, and x_t denotes the entire OD flow at time t. The Traffic state evolution matrix is then

$$x_{t+1} = Cx_t + w_t$$

where, C is the state transition matrix and w_t is the traffic system noise process and the measurement matrix is

$$y_t = Ax_t + m_t$$

To apply the Kalman filter we need C, Q and R and initial conditions, $\hat{x}_{0|0}$ and $\hat{P}_{0|0}$. As an example, 24 hours of Netflow data can be used to estimate these parameters using maximum likelihood estimation. The innovation process $i_{t+1} = y_{t+1} - Ax_{t+1}$ is used for change detection in order to recalibrate the system matrices.

2.2 Graph Model

In this task, we aim to develop numerical methods to study the anomaly/botnet detection in graph-based models for network traffic data superimposed with uncertainty. This enables us to quantify the stochastic nature of the anomaly/botnet information identified through graph models. More specifically, we focus on the development of multilevel spectral graph-clustering techniques. Graph clustering is one of the most popular tools for unsupervised data analysis and has been applied to anomaly detection in cybersecurity [Amini, 2014; Munz, 2007]. There are various clustering algorithms for graphs, such as k-means and spectral clustering. We consider spectral clustering as it is often more efficient than the traditional k-means algorithm when efficient linear algebra packages are employed [von Luxburg, 2007]. Another graph clustering algorithm is based on nonnegative matrix factorization (NMF), which provides a lower rank approximation of a nonnegative matrix [Ding, 2006]. In [Ding, 2005], it is shown that a symmetric NMF is equivalent to the Laplacian-based spectral clustering. An important class of graph clustering algorithms is the so-called hierarchical clustering algorithms which create a multilevel decomposition of the original graph from either top down (divisive) or bottom up (agglomerative), e.g., [Anders, 2003; Zhou, 2008]. In [Boley, 2001], a scalable algorithm has been developed for top-down hierarchical clustering. Using spectral clustering, a hierarchical representation of complex networks has been developed [Fang, 2013].

It is known that network traffic data is associated with many uncertainties arising from data collection, transportation, preprocessing, or attacks in progress [Li, 2013]. Probabilistic graph models (or uncertain graphs) have been developed for graphs with uncertainties, e.g., [Kollios, 2013; Moustafa, 2014; Potamias, 2010; Rotsos, 2010]. A probabilistic graph may be considered as a generative model for deterministic graphs. More precisely, let $\mathcal{G} = (V, E, P, W)$ be a probabilistic graph, where V and E denote the set of nodes and edges, P denotes the probabilities associated with the edges, $p(e)$ denotes the probability of edge $e \in E$, W denotes the weights and $w(e)$ is the weight of an edge e. If G is sampled from $\mathcal{G}$ according to the probabilities P, E_G denotes the set of edges of G, then the probability associated with G is:

$$\Pr[G] = \prod_{e \in E_G} p(e) \prod_{e \in E \setminus E_G} (1 - p(e))$$

There have been a few papers on clustering algorithms for probabilistic graphs, see e.g., [Kollios, 2013; Moustafa, 2014; Rotsos, 2010]. For other clustering algorithms developed for general uncertain data, we refer to the survey paper [Aggarwal, 2012]. In our research, we will take an approach that is different from existing ones which utilizes advance UQ techniques and scalable eigensolvers.

2.3 Machine Learning Model

Recently, we have witnessed technological advances unfolding on many fronts that enable capturing and collecting of complicated data sets on unprecedented scales. Resolution and speed of sensors continue to increase while cost and resource consumption of sensors continue to decrease. Storage capability continues to improve, reducing the cost of retaining big data. Similar data growth is common in many fields including cybersecurity. Analyzing and mining big data in those fields can uncover interesting patterns that can accelerate scientific discovery, prevent failures, and improve system efficiency. However, several challenges must be addressed to make big data analytics effective. First, large-scale data sets can contain much noise. This noise can include measurement noise or human errors in labeling data. Second, many complex data sets can have very high dimensionality and the interaction between these dimensions can be highly complicated. As a result, some parts of a sample space may not contain sufficient amount of data. Third, data-mining and machine-learning algorithms used in data analytics can affect predictions generated by these algorithms. If the algorithms are not carefully chosen, overfitting or underfitting can be possible, especially given the large number of attributes in big data. In this research, we will focus on the cyber modeling with emphasis on the network flow applications.

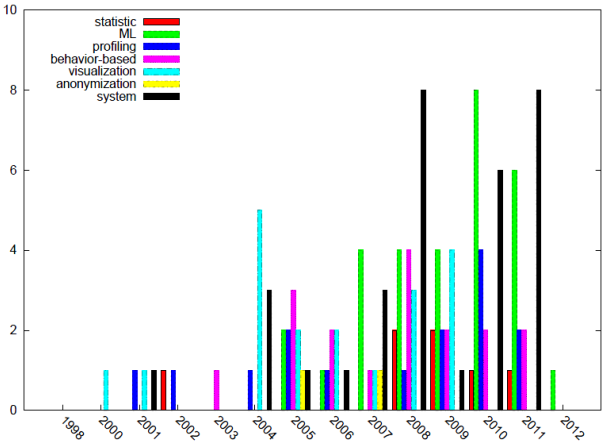


Figure 1 Methodologies for network traffic analysis by year.

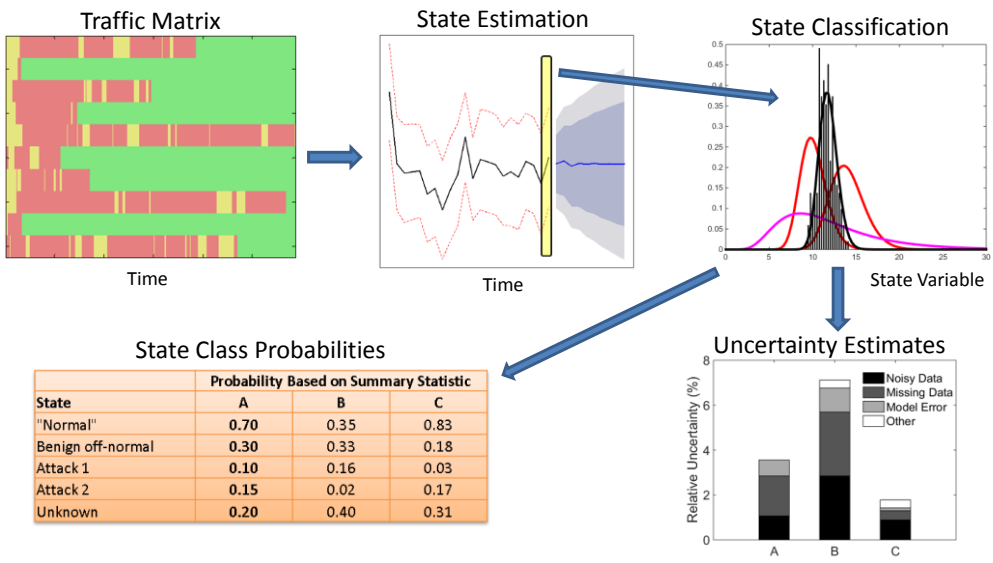
Network security has been an important area of research since the very beginning. Many advanced methodologies have been developed for NetFlow and network traffic analysis. A good survey of these methodologies has been provided in [Li, 2013]. A chronological plot of methods applied from 2000 to 2012 is shown in **Error! Reference source not found.**. A significant number of studies focus on using the machine learning algorithms, a very promising methodology with a lot research showing that machine learning approaches are better than statistical methods for NetFlow-based intrusion detection because of limited variables in the NetFlow data. In particular, [Kim, 2008] completed a comprehensive evaluation of seven machine-learning algorithms for the application of traffic classification. These algorithms include: Naïve Bayes, Naïve Bayes Kernel Estimation, Bayesian Network, Decision Tree, k-nearest Neighbors, Neural Networks, and SVM. Their evaluation indicates that SVM consistently achieved higher accuracy for this particular application. SVM is one of the most popular machine learning algorithms for classification and is widely used in cybersecurity applications, e.g., Email spam. Based on previous studies, we identify the support vector machine as the starting example for an extensive UQ study.

3.0 UQ Methodology

Recognizing that the kind of model development needed to enable UQ in this domain is in its very early stages, our research is exploratory in nature. In order to progress along several paths, we focus on three subtasks to develop UQ methods, to be coupled to existing modeling approaches. In all three subtasks, the scalability of UQ methods for cybersecurity models will be studied to understand when and at what scale limitations arise.

Figure 2 illustrates our view of how uncertainty propagates from data (e.g., NetFlow data) and models (e.g., data filters, distribution assumptions, and classification schemes) through to decisions (e.g., identification of a particular type of network attack). The figure also suggests capturing that uncertainty at the decision level in the form of, for example, a measure of likelihood or probability of the network being in a given state class, as well as a breakdown of the importance of the different underlying sources of uncertainty—such as data sampling, data quality, model uncertainty, and training data labeling errors—on that overall uncertainty estimate.

Using notional data in the insets within Figure 2, our example begins with a NetFlow-based traffic matrix (only a very tiny portion of which is indicated), here represented in vectorized form, as the matrix evolves over time. The second inset indicates estimation of the true state, represented, for example, by either an estimate of the true (incompletely known) traffic matrix or a summary statistic, along with an estimate of the uncertainty in that estimated state, over time. Based on training data, the state at any given time may be classified as one of many known (or an unknown) anomaly or attack states, characterized in terms of probability distributions. Ideally, this enables a probabilistic statement about the likelihood of the state being in each of the known (and an unknown) class as indicated in the table. Finally, the process should enable an uncertainty “budget”, identifying the major contributors to the decision uncertainty.



	Probability Based on Summary Statistic		
State	A	B	C
"Normal"	**0.70**	0.35	0.83
Benign off-normal	**0.30**	0.33	0.18
Attack 1	**0.10**	0.16	0.03
Attack 2	**0.15**	0.02	0.17
Unknown	**0.20**	0.40	0.31

Figure 2 Diagram illustrating our overall UQ methodology

Extending from this example, we view characterization of network states as a two-step process: (1) estimate a state vector as a function of time, and (2) classify the network state. Each UQ approach that we study tackles a piece of the problem of propagating uncertainty to measures of confidence in state classification. As noted in the Introduction, we anticipate that the component methods can either be directly combined and/or benefit from concurrent development.

The algorithms that we are developing will be implemented into prototype software and tested against available data. Results of these tests will be used to evaluate the effectiveness of each model and identify where new development and data are needed.

3.1 UQ for State-space model

One approach to modeling uncertainties in state space models is by using additive Gaussian noise and estimating the covariance matrices of this noise and the state transition matrix using maximum likelihood estimation. Then using a Kalman filter, one can estimate the state vector (traffic matrix) and its covariance matrix. Due to the additive Gaussian noise in state and measurement equations in the Kalman filter equations, the traffic matrix will follow a multi-variate Gaussian distribution.

Kalman Filter

Let y_t be an observation vector at discrete time t and let $Y_t = \{y_t\}$ be the set of all observations up to time t and x_t denote the entire OD flow at time t. The Traffic state evolution matrix is

$$x_{t+1} = Cx_t + w_t$$

where C is the state transition matrix and w_t is the traffic system noise process. The measurement matrix is

$$y_t = Ax_t + m_t.$$

The prediction step is

$$\hat{x}_{t+1|t} = Cx_{t|t}$$
$$\hat{P}_{t+1|t} = C\hat{P}_{t|t} + Q_t$$

And the estimation step is

$$\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + K_t(y_t - A\hat{x}_{t+1|t})$$
$$K_t = \hat{P}_{t+1|t}A_t(A\hat{P}_{t+1|t}A^T + R_t)^{-1}$$
$$\hat{P}_{t+1|t+1} = (I - K_tA)\hat{P}_{t+1|t}.$$

Expectation-Maximization

Consider state variables *X*, observed variables *Y* and latent variables *Z* and parameters of the model, θ. The Expectation-Maximization algorithm is then:

1. Set *k*=0, initialize θ_0 such that L_{θ_k} is finite,
2. Expectation step: compute

$$Q(\theta, \theta_k) = E_{\theta_k}[\log(p_0(Z,Y)|Y)]$$
$$= \int \log p_0(Z,Y)\, p_{\theta_k}(Z|Y)dZ$$

3. Maximization step: compute

$$\theta_{k+1} = argmaxQ(\theta, \theta_k)$$

4. If not converged, *k*=*k*+1, go to step 2

Traffic matrices obtained from training NetFlow data are large and very sparse. The challenges inherent in estimating the traffic matrices have led us to revise our approach to this problem in the following ways:

- Aggregate the source and destination nodes such that the size and sparsity of the traffic matrix is reduced. When information about a specific node is needed we can look inside the aggregated group in which that node lies.
- Estimate summary statistics on the traffic matrix, enabling filtering and tracking of a much smaller number of state variables. Possible summary statistics include
    - measures of entropy [Nychis, 2008],
    - the dispersion and smoothness measures of Joslyn and Hogan [Joslyn, 2014], and
    - the Ripley's K-function and L-function [Ripley, 1981], which give a measure of deviation from spatial homogeneity. This may be useful in observing patterns and changes in the structure of the traffic matrix.

3.2 UQ for graph models

We design stochastic spectral clustering algorithms for anomaly detection in network traffic data by utilizing scalable eigensolvers. We first introduce the deterministic spectral clustering algorithm. Given a traffic dataset $X \in \mathbb{R}^{n\times k}$, i.e.,

$$X = [x_1, x_2, \cdots, x_n]^T$$

consisting of n data points and k features for each data point, we can construct an undirected, weighted similarity graph $G = (V, E)$ with adjacency matrix W, where $w_{ij} = w_{ji} \geq 0$, and $w_{ii} = 0, \forall i$. Each vertex v_i in this graph represents a data point. Each edge is weighted by w_{ij} representing the similarity between the two corresponding data points x_i and x_j. The graph clustering refers to the partition of the graph such that points in different clusters are dissimilar from each other while points in the same cluster are similar. As an example, we can compute pairwise similarity between the two data points x_i and x_j by the Gaussian similarity function

$$w_{ij} = e^{-\|x_i - x_j\|^2/(2\sigma^2)}.$$

The unnormalized graph Laplacian matrix is defined by

$$L = D - W,$$

where D is the node degree matrix which is a diagonal matrix whose diagonal entries are row sums of the weighted adjacency matrix W. Define normalized graph Laplacian by

$$L_{rw} = D^{-1}L = I - D^{-1}W,$$

then the Shi-Malik normalized spectral clustering uses the positive/negative signs of the eigenvector associated with the second smallest eigenvalue to bipartition the graph [Shi, 2000].

Modeling the uncertainty within such a graph model is a non-trivial task. A straightforward way might be incorporating uncertainty into the graph constructed from the network traffic data, e.g., to the edges between the nodes. However, the dimension of the uncertainty introduced through this probabilistic graph approach is extremely large, e.g., $O(n^2)$, where n is the number of traffic data points. As a result, numerical quantification on the stochasticity on network traffic data is unsolvable. Alternatively, we propose to introduce the uncertainty to the network traffic data with reduced dimensionality. In particular, we can use the Karhunen-Loeve expansion to transfer the data uncertainty to parametric form. We define $w(x_i, \theta)$ as the randomness imposed on each data point x_i. We assume the random function $w(x, \theta)$ has an exponential correlation kernel

$$C(x_1, x_2) = e^{-c|x_1 - x_2|},$$

where c is the inverse of correlation length between the individual data point and the x_i is defined between [-a, a]. Then we can approximate $w(x, \theta)$ with finite dimension of stochasticity by

$$w(x, \theta) = \sum_{n=1}^{D} \xi_n(\theta)\sqrt{\lambda_n} f_n(x) + \sum_{n=1}^{D} \xi_n^*(\theta)\sqrt{\lambda_n^*} f_n{}^*(x)$$

where D is the dimension of the stochasticity and the corresponding eigenvalues are given by

$$\lambda_n = \frac{2c}{{w_n}^2 + c^2}, \quad {\lambda_n}^* = \frac{2c}{{w_n^*}^2 + c^2}$$

and the eigenfunctions are given by

$$f_n(x) = \frac{\cos(w_n x)}{\sqrt{a + \frac{\sin(2w_n a)}{2w_n}}}, \quad f_n^*(x) = \frac{\cos(w_n^* x)}{\sqrt{a - \frac{\sin(2w_n^* a)}{2w_n^*}}}$$

With the above formulation, we are able to investigate the stochasticity of the network traffic data with reduced dimension of uncertainty D.

With the Karhunen-Loeve expansion model for input data uncertainties, we can use Monte Carlo sampling to find uncertainties associated with the spectral clustering results. We will use multilevel eigensolvers to increase the computational efficiency [Urschel, 2015]. Furthermore, polynomial chaos techniques may further reduce the computational complexity of uncertainty propagation through graph clustering for anomaly detection.

Figure 3 illustrates possible notional results of graph clustering for the extension from deterministic to random graph models.

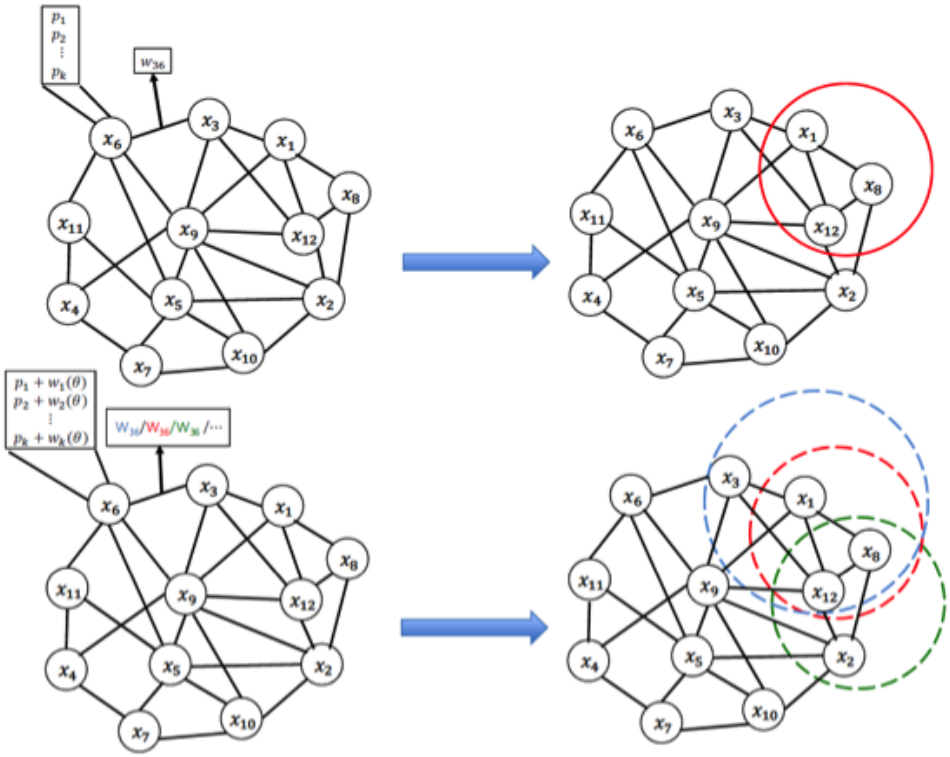


Figure 3 An example of graph clustering for deterministic (top) or random (bottom) graphs.

3.3 UQ for Machine learning-based models

Most data analytics such as the set of frequently used machine-learning algorithms generally only produces predicted values. The predicted values are the only information for users to take action. In many usage scenarios, taking action based on inaccurate predicted values can incur significant costs. For instance, false positives of network intrusions can cause many essential services to be shutdown or isolated, causing loss of productivity. Tuning data analytics to reduce false positives at the cost of more false negatives can cause even more harm. Thus, it is highly desirable for big data analytics to produce a confidence level of the predicted values, namely the prediction with quantified uncertainty. The confidence level can provide more information for users to fine tune their response to achieve optimal results.

There are several studies focused on the machine-learning algorithms with noise-corrupted input data [Pant, 2011], and mostly for neural network algorithms [Wright, 1999; Wright, 2000]. In addition, previous work in this area generally falls into two categories, statistical learning theory [Vapnik, 2013] and cross validation [Jiang, 2008]. In statistical learning theory, training data are assumed to be drawn independently from a fixed but unknown distribution. A very loose bound on the number of wrong predictions can be given that is not very useful in practice. On the other hand, cross validation can require a large number of training samples to be reasonably accurate, which may be impractical when the sample space is large and complex.

In our research, we will explore the effect of various sources of uncertainty including sample data, model parameters in soft margin SVM, and mislabeling data. We will explore different approaches to derive better estimations for the accuracy of SVM results for NetFlow data analysis. In our approach, we will quantify the influence of various factors including sample size and noise to accuracy of predictions. Instead of estimating overall mis-prediction rates, we will leverage attribute values of test samples to get a better estimation. We also design some simple models with analytical solutions available to help us understand the overall trend of accuracy of machine learning algorithms. Insight will be gained from relatively simple models and applied to complex situations where no analytical solution exists. We expect our approach will produce a better estimation of uncertainties than previous work [Jiang, 2008]. In particular, a model quantifying how various factors including sample size and noise affect prediction accuracy will be developed. If users want to achieve a certain level of prediction accuracy and they can control various parameters, they can use this model to change parameters to achieve desirable accuracy.

4.0 Use Cases

We are using NetFlow and similar traffic data to implement existing models that can be used in real time for characterizing the data and detecting and characterizing anomalies and cyber attacks, and enhance those models with UQ. To begin our study, we obtained synthesized NetFlow data from 2013 VAST mini challenge-3 and the NSL-KDD datasets.

Our plan is to use the VAST NetFlow data to build and test various state-space models. As an example, we are computing entropies of several features in the Netflow data such as source IP, destination IP, source port, destination port, flow size distribution, out-degree the host IP and in-degree of the destination IP. Entropies are computed for certain time interval (e.g. 5 minute bins)

so that we get a time series of entropy for each feature, as shown in **Error! Reference source not found.**. We then use

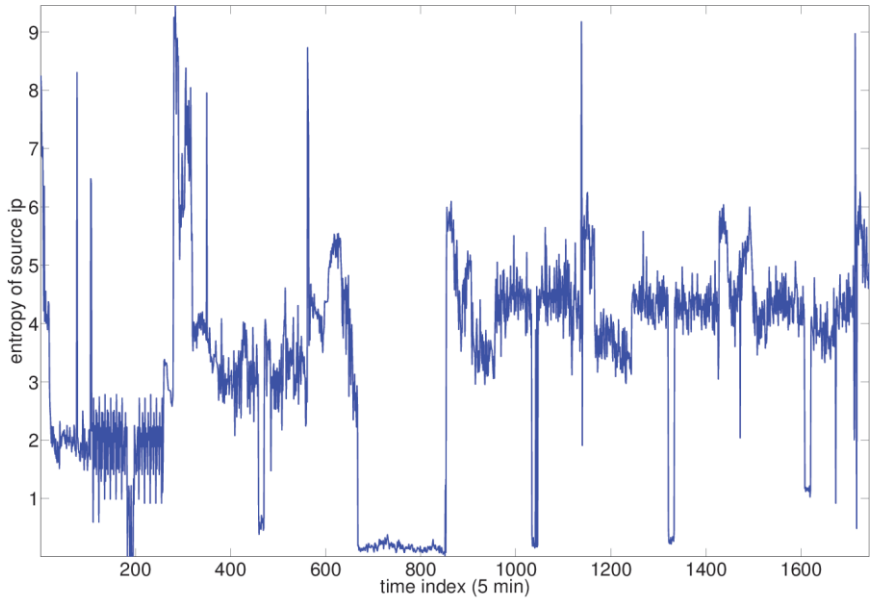


Figure 2 Entropy computed for source IP on first week of VAST data.

dynamic state-estimation methods for these time series for UQ and anomaly/attack characterization.

We plan to test UQ for graph models on the NSL-KDD dataset, which has been widely used for anomaly detection studies. It contains connection records based on the simulated raw (binary) TCP dump data of nine weeks of network traffic on a local area network. Each connection record consists of 41 features including length of time duration of the connection, the protocol used in the connection (TCP or UDP), number of data bytes transferred from source to destination in single connection, number of data bytes transferred from destination to source in single connection, and other statistics of the connection [Dhanabal, 2015]. We will use this dataset to build our stochastic graph model and apply a scalable eigensolver to find eigenvectors of the graph Laplacian for each sample graph. The results will provide a quantification of uncertainty in spectral graph clustering results of the network traffic data induced by the input data uncertainties. **Error! Reference source not found.**

We are working to identify prototype machine-learning problems with all necessary elements of SVM and UQ relevant to cybersecurity. A good prototype problem will serve as the benchmark problem to test and validate the effectiveness and efficiency of the UQ algorithms for SVM. Since research in this area is still very limited, this task is considerably important. We will then use NetFlow data as the test bed to examine the efficacy of the results we already obtained for the prototype problem, and transfer the algorithms developed for the prototype problem to a more realistic problem. Special attention will be put on the scalability in this procedure.

The more realistic use case scenario for the machine learning development will utilize NetFlow data analysis for large-scale distributed systems [Li, 2013; Li, 2011; Fahad, 2013; Hoque, 2014]. Monitoring large-scale distributed systems in various levels including hardware level, operating system level, middleware level, and application level can produce a large amount of data. Applying various unsupervised and supervised machine-learning algorithms can allow us to detect various performance anomalies or security breaches. For a given prediction such as a security breach, corresponding actions such as shutdown or isolation of certain systems may need to be taken. This action can render some parts of the system unusable, causing disruption and reducing productivity. Thus false positives can incur significant cost. If we can quantify the accuracy of predictions generated by data analytics, we can finely tune the response to minimize the disruption without compromising security.

5.0 Discussion

From our initial literature review and domain-specific education, we have come to an understanding about the overall status of, and approaches to, modeling within cybersecurity. We identify this modeling strategy as a classification problem and our research focuses on adding confidence to the classification results.

In the beginning of this project, we identified three seemingly distinct modeling methodologies to investigate and develop UQ methods for implementation. After this initial research, we have identified many overlapping synergistic elements within our modeling efforts. Each approach tackles a piece of the problem of propagating uncertainty to measures of confidence in state classification. We anticipate that the component methods can either be directly combined (e.g., KF-based state vector estimation with uncertainty estimate can be input to SVM-based classification with uncertainty) and/or benefit from concurrent development (e.g., the efficiency gained via K-L expansions in stochastic graph-based clustering might be applied in a similar way to represent uncertainty in the other approaches).

As our modeling direction continues to be guided by the overall needs within cybersecurity modeling and specifically from the subject matter experts we consult, our overall development strategy remains the same. This strategy is the development of UQ methods and tools, so that decision makers will be better informed to assess the reliability and probabilities of modeling results, identify network vulnerabilities, and develop computational strategies to improve cyber systems based on uncertainty quantification. This research will also lead to advances in the design of efficient optimization algorithms and a computational framework applicable for uncertainty quantification for large-scale complex network systems.

6.0 References

AGGARWAL, C., A Survey of Uncertain Data Clustering Algorithms, In ``Data Clustering: Algorithms and Applications", ed. C. Aggarwal and C. Reddy, CRC Press, 2013.

AMINI, P., R. AZMI and M. ARAGHIZADEH. 2014. Botnet Detection Using Netflow and Clustering. 2014:11.

ANDERS, K.-H. 2003. A Hierarchical Graph-Clustering Approach to Find Groups of Objects. In Proceedings Conference A Hierarchical Graph-Clustering Approach to Find Groups of Objects.

BOLEY, D. 2001. A Scalable Hierarchical Algorithm for Unsupervised Clustering. In Data Mining for Scientific and Engineering Applications. Springer US, pp. 383-400.

DHANABAL, L., S.P. SHANTHARAJAH, A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 6, June 2015.

DING, C., X. HE and H. SIMON. 2005. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In Proceedings Conference On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering.

DING, C., T. LI, W. PENG and H. PARK. 2006. Orthogonal Nonnegative Matrix Tri-Factorizations for Clustering. In Proceedings Conference Orthogonal Nonnegative Matrix Tri-Factorizations for Clustering, pp. 126--135.

FAHAD, A., et al., Toward an efficient and scalable feature selection approach for internet traffic classification. Computer Networks, 2013. 57(9): p. 2040-2057.

FANG, Y.-P. and E. ZIO. 2013. Unsupervised Spectral Clustering for Hierarchical Modelling and Criticality Analysis of Complex Networks. Reliability Engineering & System Safety, 116:64–74.

HOQUE, N., et al., Network attacks: Taxonomy, tools and systems. Journal of Network and Computer Applications, 2014. 40: p. 307-324.

JOSLYN, C., W. COWLEY, E. HOGAN and B. OLSEN. 2012. Discrete Mathematical Approaches to Graph-Based Traffic Analysis. In International Workshop on Engineering Cyber Security and Resilience (ECSaR'14). Stanford, CA.

KIM, H.,et al. Internet traffic classification demystified: myths, caveats, and the best practices. in 2008 ACM CoNEXT Conference. 2008.

KOLLIOS, G., M. POTAMIAS and E. TERZI. 2013. Clustering Large Probabilistic Graphs. IEEE Trans. on Knowl. and Data Eng., 25:325-336.

KUANG, D., H. PARK and C. DING. 2012. Symmetric Nonnegative Matrix Factorization for Graph Clustering. In Proceedings Conference Symmetric Nonnegative Matrix Factorization for Graph Clustering, pp. 106–117.

LI, B., J. SPRINGER, G. BEBIS and M. GUNES. 2013. Review: A Survey of Network Flow Applications. J. Netw. Comput. Appl., 36:567-581.

LI, J., X. OU and R. RAJAGOPALAN. 2010. Uncertainty and Risk Management in Cyber Situational Awareness. In Cyber Situational Awareness. Springer United States, pp. 51 -- 68.

MOUSTAFA, W., A. KIMMIG, A. DESHPANDE and L. GETOOR. 2014. Subgraph Pattern Matching over Uncertain Graphs with Identity Linkage Uncertainty. In Proceedings Conference Subgraph Pattern Matching over Uncertain Graphs with Identity Linkage Uncertainty.

MUNZ, G., S. LI and G. CARLE. 2007. Traffic Anomaly Detection Using K-Means Clustering. In Proceedings Conference Traffic Anomaly Detection Using K-Means Clustering.

NYCHIS, G., V. SEKAR, D. ANDERSEN, H. KIM and H. ZHANG. 2008. An Empirical Evaluation of Entropy-Based Traffic Anomaly Detection. In Proceedings Conference An Empirical Evaluation of Entropy-Based Traffic Anomaly Detection.

PANT, R., T.B. TRAFALIS, AND K. BARKER, Support Vector Machine Classification of Uncertain and Imbalanced data using Robust Optimization Recent Researches in Computer Science 2011: p. 369-374.

POTAMIAS, M., F. BONCHI, A. GIONIS and G. KOLLIOS. 2010. K-Nearest Neighbors in Uncertain Graphs. Proc. VLDB Endow., 3:997-1008.

RIPLEY, B. D. 1981. Spatial Statistics. New York, Wiley.

ROTSOS, C., J. VAN GAEL, A. MOORE and Z. GHAHRAMANI. 2010. Probabilistic Graphical Models for Semi-Supervised Traffic Classification. In Proceedings of the 6th International Wireless Communications and Mobile Computing Conference. ACM Caen, France, pp. 752-757.

SHI, J. and J. MALIK. 2000. Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22:888--905.

SOULE, A., A. LAKHINA, N. TAFT, K. PAPAGIANNAKI, K. SALAMATIAN, A. NUCCI, M. CROVELLA and C. DIOT. 2005. Traffic Matrices: Balancing Measurements, Inference and Modeling. SIGMETRICS Perform. Eval. Rev., 33:362-373.

SOULE, A., K. SALAMATIAN, A. NUCCI and N. TAFT. 2005. Traffic Matrix Tracking Using Kaman Filters. ACM SIGMETRICS Performance Evaluation Review - Special issue on the First ACM SIGMETRICS Workshop on Large Scale Network Inference, 33:24 -- 31.

SOULE, A., K. SALAMATIAN and N. TAFT. 2005. Combining Filtering and Statistical Methods for Anomaly Detection. In Proceedings Conference Combining Filtering and Statistical Methods for Anomaly Detection., pp. 331-334.

TUNE, P. and M. ROUGHAN. 2013. Internet Traffic Matrices: A Primer. In Recent Advances in Networking, SIGCOMM eBook Cambridge, UK.

URSCHEL, J., X. HU, J. XU and L. ZIKATANOV. 2015. A Cascadic Multigrid Algorithm for Computing the Fiedler Vector of Graph Laplacian. arXiv:1412.0565v1.

VAPNIK, V., The nature of statistical learning theory. 2013: Springer Science & Business Media.

VON LUXBURG, U. 2007. A Tutorial on Spectral Clustering. Statistics and Computing, 17:395--416.

WRIGHT, W.A., Bayesian approach to neural-network modeling with input uncertainty. Ieee Transactions on Neural Networks, 1999. 10(6): p. 1261-1270.

WRIGHT, W.A., et al., Neural network modelling with input uncertainty: Theory and application. Journal of Vlsi Signal Processing Systems for Signal Image and Video Technology, 2000. 26(1-2): p. 169-188.

ZHOU, H., X. YUAN, W. CUI, H. QU and B. CHEN. 2008. Energy-Based Hierarchical Edge Clustering of Graphs. In Proceedings Conference Energy-Based Hierarchical Edge Clustering of Graphs, pp. 55-61.

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by **Battelle** *Since 1965*

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)
www.pnnl.gov