# Event and anomaly detection using Tucker3 decomposition

**Hadi Fanaee Tork**[1], **Márcia Oliveira**[2], **João Gama**[3], **Simon Malinowski**[4] and **Ricardo Morla**[5]

**Abstract.** Failure detection in telecommunication networks is a vital task. So far, several supervised and unsupervised solutions have been provided for discovering failures in such networks. Among them unsupervised approaches has attracted more attention since no label data is required [1]. Often, network devices are not able to provide information about the type of failure. In such cases, unsupervised setting is more appropriate for diagnosis. Among unsupervised approaches, Principal Component Analysis (PCA) has been widely used for anomaly detection literature and can be applied to matrix data (e.g. Users-Features). However, one of the important properties of network data is their temporal sequential nature. So considering the interaction of dimensions over a third dimension, such as time, may provide us better insights into the nature of network failures. In this paper we demonstrate the power of three-way analysis to detect events and anomalies in time-evolving network data.

## 1 INTRODUCTION

Event detection can be briefly described as the task of discovering unusual behavior of a system during a specific period of the time. On the other hand, anomaly detection concentrates on the detection of abnormal points. So clearly it is different from event detection since it just considers the points rather than a group of points. Our work takes into account both issues using multi-way data analysis. Our methodology comprises the following steps: 1) Anomaly detection: detection of individual abnormal users 2) Generating user trajectories (i.e. behavior of users over time), 3) Clustering users' trajectories to discover abnormal trajectories and 4) Detection of events: group of users who show abnormal behavior during specific time periods. Although there is a rich body of research on the two mentioned issues, to the best of our knowledge we are the first ones applying multi-way analysis to the anomaly and event detection problem. In the remainder of this section we explain some basic and related concepts and works. Afterwards, we define the problem, and then discuss three-way analysis methods. Hereafter, we introduce the dataset and experiments. Finally, we discuss the results and point out possible future directions.

### 1.1 ANOMALY DETECTION

Anomaly is as a pattern in the data that does not conform to the expected behavior [1]. Anomaly detection has a wide range of application in computer network intrusion detection, medical informatics, and credit card fraud detection. A significant amount of research has been devoted to solve this problem. However our focus is on unsupervised methods. Anomaly detection techniques can be classified into five groups [1]: classification-based, clustering-based, nearest neighbor based, statistical methods, information theory-based methods and spectral methods. Based on this classification, our method is placed in the group of spectral methods. These approaches first decompose the high-dimensional data into a lower dimension space and then assume that normal and abnormal data points appear significantly different from together. This some benefits: 1) they can be employed in both unsupervised and supervised settings 2) they can detect anomalies in high dimensional data, and 3) unlike clustering techniques, they do not require complicated manual parameter estimation. So far, most of the work related to spectral anomaly detection was based on Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). Two of the most important applications of PCA during recent years has been in the domain of intrusion detection [2] [3] and traffic anomaly detection [4] [5].

### 1.2 EVENT DETECTION

Due to huge amount of sequential data being generated by sensors, event detection has become an emerging issue with several real-world applications. Event is a significant occurrence or pattern that is unusual comparing to the normal patterns of the behavior of a system [6]. This can be natural phenomena or manual system interaction. Some examples of events can be an attack on the network, bioterrorist activities, epidemic disease, damage in an aircraft, pipe-breaks, forest fires, etc. A real system behaves normally most of the time, until an anomaly occurs that may cause damages to the system. Since the effects of an event in the system are not known a priori, detecting and characterizing abnormal events is challenging. This is the reason why most of the time we cannot evaluate different algorithms. One solution might be injection of artificial event into the normal data. However, construction of a realistic event pattern is not trivial [7].

### 1.3 HIDDEN MARKOV MODELS

Hidden Markov Models (HMMs) have been used at least for the last three decades in signal processing, especially in domain of speech recognition. They have also been applied in many other domains as bioinformatics (e.g. biological sequence analysis), environmental studies (e.g. earthquake and wind detection), and finance (financial time series). HMMs became popular for its simplicity and general mathematical tractability [8].

[1] LIAAD-INESC TEC, FEUP- University of Porto, hadi.fanaee@fe.up.pt
[2] LIAAD-INESC TEC, FEP- University of Porto, marcia@liaad.up.pt
[3] LIAAD-INESC TEC, FEP- University of Porto, jgama@fep.up.pt
[4] INESC TEC, FEUP- University of Porto, sjfm@inescporto.pt
[5] INESC TEC, FEUP- University of Porto, ricardo.morla@inescporto.pt

HMMs are widely used to describe complex probability distributions in time series and are well adapted to model time dependencies in such series. HMMs assume that observations distribution does not follow a normal distribution and are generated by different processes. Each process is dependent on the state of an underlying and unobserved Markov process [7]. Markov process denotes that the value of a process $X_t$ only depends on the previous value of X. Using notations of [9] let:

T = Length of the Observation sequence
N = Number of states
Q = {$q_0$, q1, …, q $_{(N-1)}$ } = distinct states of Markov process
A = State transition probabilities
B = set of N observation probability distributions
π = Initial State Distribution
O = ($O_0$, O1,…, O $_{(T-1)}$) = observation sequence

A HMM Model is defined with the triple of λ= (A, B, π). It assumes that Observations are drawn using the observation probability distribution associated to the current state. The transition probabilities between states are given in matrix A.

The three main problems related with HMMs are the following. The first problem consists in computing the probability P(O) that a given observation sequence O is generated by a given HMM λ. The second problem consists in finding the most probable sequence of hidden states given an observation sequence O and λ and the third problem is related to parameter inference. It consists in estimating the parameters of the HMM λ that best fits a given observation sequence O. The mainly used algorithms to solve these problems are given in the last column of Table 1. More details about these algorithms can be found in [10]. In this paper, we deal with the third problem to estimate the HMM parameters that best describe time series, as it will be explained in Section 2.

**Table1.** Three HMM Problems

| Problem | Input | Output | Solution |
|---|---|---|---|
| Problem 1 | λ, O | P(O) | Forward Backward algorithm |
| Problem 2 | λ, O | Best Q | Viterbi algorithm |
| Problem 3 | O | λ | Baum-Welch algorithm |

## 1.4 THREE-WAY DATA ANALYSIS

Traditional data analysis techniques such as PCA, clustering, regression, etc. are only able to model two dimensional data and they do not consider the interaction between more than two dimensions. However, in several real-world phenomena, there is a mutual relationship between more than two dimensions (e.g. a 3D tensor (Users×Features×Time)) and thus, they should be analyzed through a three-way perspective. Three-way analysis considers all mutual dependencies between the different dimensions and provides a compact representation of the original tensor in lower-dimensional spaces. The most common three-way analysis models are Tucker2, Tucker3, and PARAFAC [10] which are generalized versions of two-mode principal component model or, more specifically, SVD. Following, we briefly introduce Tucker3 model as the best-known method for analysis of three-way data.

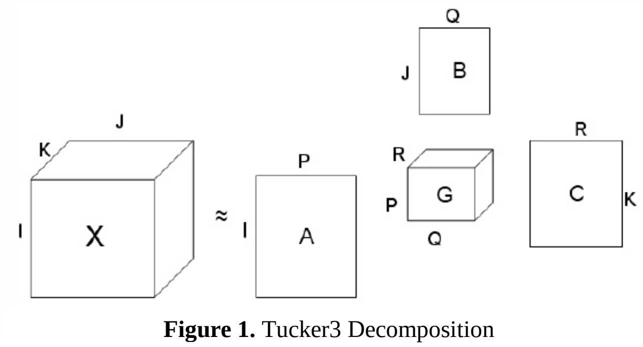

**Figure 1.** Tucker3 Decomposition

### *1.4.1 Tucker3 Model*

The Tucker3 model decomposes a three-mode tensor $\mathcal{X}$ into set of component matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and a small core tensor $\mathcal{G}$. The following mathematical equation reveals the decomposition:

$$\mathcal{X}_{ijk} \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} \mathcal{G}_{pqr} \times a_{ip} \times b_{jq} \times c_{kr} \quad (1)$$

Where P, Q and R are parameters of the Tucker3 model and represent the number of components retained in the first, the second and the third mode of the tensor, respectively. This decomposition is illustrated in Figure 1.

## 2 PROBLEM DEFINITION

One of significant issues in telecommunication systems, such as IP/TV, is to detect the anomalies at both network and user level. In order to study this, target users are usually equipped with a facility in their modem which sends an automatic notification message to the central server when the connection of a client in the network is lost or reestablished. These modems are ubiquitous and geographically dispersed.

The modeling of such behavior is not straightforward because the number of notification messages is not equal for each user during the time period under analysis. For instance, one user may face 40 connection problems in an hour, hence generating 40 messages, while others may face 5 or even no problems at all. In standard event detection problems, for each time point there is a measurement via one or multiple sensors. In the context of our application, such measurements do not take place at regular time points, since user modems (or sensors) only send messages to the server when something unexpected occurs. Figure 2 illustrates two sample users. Each circle represents the time stamp at which a notification relative to the given user is received, while ΔT represents the inter-arrival time between two consecutive messages. As it can be seen, 2 messages were related to user 1 in that period, while 4 were related to user 2 during the same period. Also, the ΔT between messages is larger for user 1 than for user 2. This means that user 2 sent messages more frequently than user 1. As in many other event detection problems, we could easily use the number of events per hour (measurement) at different users (sensors) to detect the events but this way we would lose the information content provided by the ΔT's.

As the number of ΔT is not the same for each user, this feature cannot be directly integrated in our model. Hence, this would cause

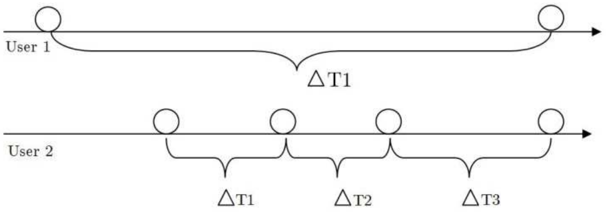

**Figure 2.** Two sample users with different number of messages and different intervals

some vectors to have different lengths, which is not supported by the Tucker3 analysis. To solve this, every time-series of ΔT relative to a given user is modeled by a 2-state HMM obtained by the Baum-Welch algorithms [11]. 6 parameters are extracted from the HMM and are used to describe the time-series of ΔT of the users. Using this approach we obtain the same number of features for each user and, then, include this information in our feature vectors.

**Table2.** Datasets in tensor format

| **Data** | **1st mode (I)**<br>Users | **2nd mode (J)**<br>Features | **3rd mode (K)**<br>Hours |
|---|---|---|---|
| X102 | 102 | 10 | 720 |
| X909 | 909 | 10 | 720 |

## 3 DATASET

Dataset is extracted from the usage log of a European IP/TV service provider. The raw dataset includes the notification messages of users in each line including their occurrence time. As previously mentioned, it is not possible to use this data directly in our modeling approach, so some pre-processing steps were performed. In addition to the obtained HMM parameters for each hour and for each user, we included another features, such as mean, variance, entropy and number of messages per hour, to our feature vector. We generated two separated datasets, each one spanning a time period of one month, which is equivalent to 720 hours. In one set we selected 102 users and in another we selected 909 users. The latter dataset is an extended version of the former. We then transformed both datasets to the tensor format. These datasets are shown in a format of Tucker3 input tensor (figure 1) in Table 2 where I, J, K represent users, features and hours modes, respectively.

## 4 EXPERIMENTS

This section is divided into three subsections, according to the steps mentioned in the Introduction section. In subsection 1, we explain how we detect the abnormal users. In the next subsection we describe how we generate user trajectories And in the last subsection we explain how we cluster the trajectories using hierarchical clustering and detect events using user trajectories.

### 4.1 Abnormal Users

We applied Tucker3 model to both datasets X102 and X909 by employing a MATLAB package called *Three-mode component analysis (Tucker3)* [10]. Before that, we performed ANOVA test [10] to see the significance of three-way and two-way interaction in the data. The results of this test are presented in Table 3. *ANOVA Max 2D* represents the maximum value obtained via different combinations of two-way modeling (e.g. I-J, J-K, I-K). As it can be seen, bigger numbers are obtained for three-dimension interaction (*ANOVA 3D*), which reveals that there is a mutual interaction between the three dimensions in both datasets that can be explained better with three-way modeling like Tucker3, than with two-way modeling like PCA.

**Table3.** ANOVA test and selected model parameters P-Q-R

| **Data** | **ANOVA**<br>max 2D | **ANOVA**<br>3D | **Selected Model**<br>**P-Q-R** | **fit** |
|---|---|---|---|---|
| X102 | 26.18% | 38.90% | 3-2-2 | 42.00 |
| X909 | 17.02% | 78.04% | 40-2-4 | 51.01 |

The next step is to estimate the best parameters P, Q, R of Equation 1. P-Q-R is similar to what we have in PCA. In PCA we just determine the number of PCs for one dimension but here we need to determine the number of principal components for each one of the three modes. P, Q and R can assume values that fall within the interval $[1, max]$, where $max$ denotes the maximum number of entities in the corresponding mode. For example, in terms of X102 the P-Q-R can go from 1-1-1 to 102-10-720. These parameters are chosen based on a trade-off between model parsimony, or complexity, and goodness of fit. For instance, regarding the mentioned dataset, 1-1-1 gives about 28% fit (less complete and less complex) and model 102-10-720 gives 100% fit (most complete and most complex). If we try parameters 3-2-2 the model has a 42% fit. So it can be more reasonable choice because it finds a good compromise between complexity and fit. In [10] the *scree test* method is proposed as a guideline to choose these parameters. We used this test to determine the best model for both datasets. The selected model parameters and their corresponding fits are presented in Table 3. This means that, for example, for dataset X102 if we choose Tucker3 model with 3, 2 and 2 components to summarize the users, features and hours modes, respectively, the model is able to explain 42% of the total variance contained in raw data. After the estimation of model parameters, we used the selected model to decompose the raw data into a lower dimensional subspace, as illustrated in Figure 1 and Equation 1. After the decomposition we obtained matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, a small core tensor $\mathcal{G}$ and a tensor of residual errors.

In order to detect the abnormal users we simply projected the users on the component space yielded by matrix $\mathbf{A}$. This projection is presented in Figure 3, for dataset X102. The three-dimensional subspace is given by the three obtained components by the model for the 1st mode (users). As mentioned earlier, this number of components is one of the parameters of the model, namely $\mathrm{P} = 3$, which corresponds to the first mode.

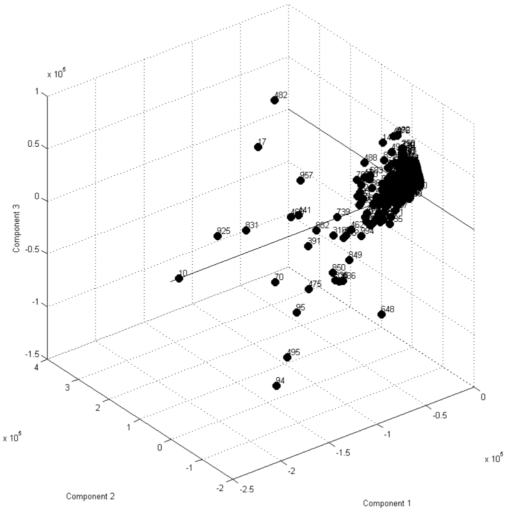

**Figure 3.** Projection of Users on Matrix **A** for dataset X102

In order to evaluate the reliability of the model we used the same procedure and applied a Tucker3 model to dataset X909, which includes all users of X102. Our idea was to see how this model can identify abnormal users from both datasets. For this purpose, we computed the Euclidean distance between each user in the projection space (see Figure 3) and the corresponding center (0, 0, 0), for both datasets X102 and X909. Then we normalized the distances for each dataset and computed the Pearson correlation for the common users of these two datasets, according to their distance to the center of the subspace. We obtained a correlation of 68.44%. Although, for X909 we just took 3 out of 40 main components to and model fit was different for both datasets (42% for X102 and 51.01% for X909), abnormal or normal users in X102 approximately appeared as the same way in X909 with 68.44% confidence. This denotes that Tucker3 is a robust model to detect the abnormal users.

### 4.2 User Trajectories

Visualization methods like the one we presented in Figure 3 are not able to show the evolving behavior of users over time. We need another solution to enable us understanding the behavior of users over time. One solution is to project the users on a decomposed feature space (matrix **B** of Figure 1) for each time point. Since both of our selected parameters have Q equal to 2 it means that after projecting Users on feature space we must have a coordinate of $(x, y)$ for each timepoint and for each user. The process of generating this coordinates is presented in Figure 4. $B_{:,1}$ and $B_{:,2}$ represent the two components that summarize the original entities of the features mode and $\mathcal{X}$ represent the three-order tensor (see Figure 1). The rows of the front matrix are the users, the columns correspond to the features and the third mode ($z$-axis) represents the hours. If we compute the dot product between each tensor's rows with the columns of the component matrix **B**, yielded by the Tucker3 model we obtain the coordinate$(x, y)$ for a given timepoint. If we repeat this procedure for all time points (e.g. hours), we are able to generate the coordinates of each user for the 720 hours. The user trajectories are obtained by sequentially connecting these coordinates. Formally we define user trajectories as:

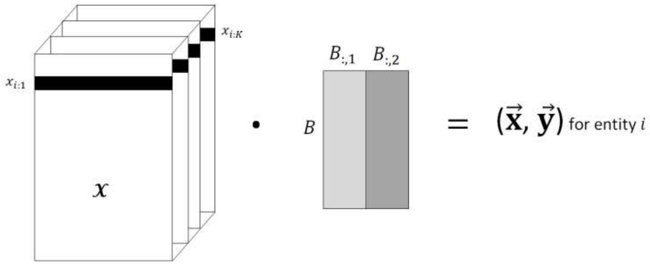

**Figure 4.** Generation process of user trajectories

**Definition 1 (User Trajectory)** : A sequence of time-stamped points, $Trj = p_0 \rightarrow \ p_1 \rightarrow \ ... \rightarrow p_i \rightarrow \cdots \rightarrow p_k$, where $p_i \ (x, y, t)$ $(i = 0,1, ... k)$, and $t$ is a time point.

Figure 5 shows two abnormal users appearing in the top-10 users ranked based on abnormality. These abnormal users were ranked based on decreasing values of distance to the center, as explained in subsection 4.1, user 10 (right) is ranked 2nd and user 95 (left) is ranked 4th. However, as it is clear from the figure, their behavior over time is completely different. User 95 just shows two abnormal behaviors that correspond to two time points, while user 10 shows this abnormal behavior almost in all time points. This means that user 10 is dealing with a stable problem while user 95 only has problems in specific points in time. This type of interpretation was not possible based only on the ranking list of abnormal users, obtained in subsection 4.1. Using user trajectories provides us richer insights into different kind of problems a user can experience. For instance, what made user 95 be identified as abnormal could be something that suddenly happened in the network and then was quickly solved, while for user 10, some problems occurred but they weren't solved until the end of the time period under analysis.

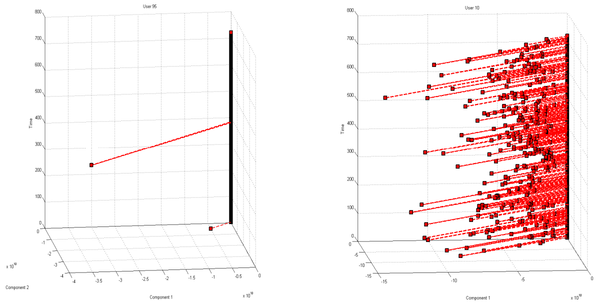

**Figure 5.** Two sample users trajectories in X909, Left) 4th ranked abnormal user Right) 2nd ranked abnormal user

### 4.3 Event Detection from user trajectories

Even though user trajectories can be useful, when the number of users is too large, the individual analysis of each trajectory can become a cumbersome task. If we notice that some group of users trajectories behave similarly, this can be understood as something abnormal happens in their network level. Then some prevention or surveillance operations can be conducted more quickly.

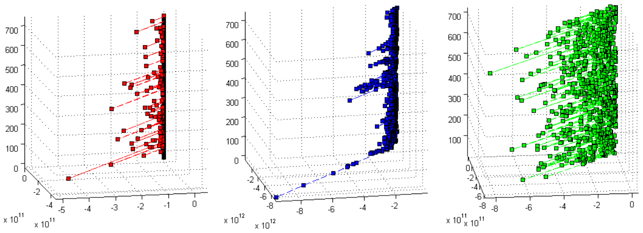

**Figure 6.** Center Trajectory of clusters, Left: 1 user, Center : 866 users and Right: 22 users.

To explore this goal, we employed Agglomerative Hierarchical Clustering toolbox from MATLAB to cluster user trajectories. We defined Euclidean distance between each point in trajectories as our distance function and Ward's criterion as the linkage criterion. We tested different values of cut-off from 0.6 to 1.2 to examine the clustering structure. The most suited clustering structure was obtained for a dendrogram distance of 1, which cuts the tree to level that, corresponds to three clusters. The average trajectory of these clusters is shown in Figure 6. Cluster red has1 user (0.1%), cluster blue comprises 866 users (97.4%) and cluster green includes 22 users (2.5%). As it can be seen, no specific pattern can be recognized from the green and the red cluster. The users in these two clusters show an abnormal behavior almost in all time points. Such event can be due to a stable specific problem such as a problem in the user device. Regarding the blue cluster, it is possible to detect three events. First significant event occurs between hours 350 to 400. Second and third events also occur between 450 to 480 and 520 to 560, respectively. However, the occurrence of the second and the third events should be assessed with hypothesis testing since they can be due to an accidental change.

## 5 CONCLUSIONS AND FUTURE WORK

In this paper, we present a study on using the Tucker3 decomposition to discover abnormal users in an IP/TV network. Our results indicate that Tucker3 is a robust method for detecting abnormal users in situations where interactions between the three dimensions are present. From the tensor decomposition, we can define user trajectories. The trajectories allow us to observe the behavior of these users over time. We were able to identify two kinds of abnormal users: those who show frequent abnormal behavior over the whole time period and those who are associated to one or few severe abnormal behaviors over the time period. Without resorting to the analysis of user temporal trajectories it would have been harder to uncover such facts. Furthermore, from the clusters of the users' trajectories, we have identified three events that occurred during three time points in the network. The result of this work can be used in a real network surveillance system to identify failures in the quickest possible time. In this work, we did not consider the spatial relation of users. Taking into account spatial relationships between network nodes could lead to a better clustering of users. Since some users might show similar behavior, with some delays, other distance measures for clustering should be tested. Currently we are employing another distance function using dynamic time warping, which assigns two users with same behavior but with a time shift in the same cluster. The solution we presented for detection of events was based on clustering of trajectories. We are going to apply sliding window on trajectories to find time periods that have the most compact trajectories, which would lead to the discovery of events in a more accurate and reliable way

## ACKNOWLEDGEMENTS

This work was supported by the Institute for Systems and Computer Engineering of Porto (INESC TEC) under projects TNET with reference BI/120033/PEst/LIAAD and project KDUS with reference PTDC/EIA-EIA/098355/2008. The authors are grateful for the financial support. This work is also financed by the ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project CMU-PT/RNQ/0029/2009. We'd like to acknowledge the support of J.A. at the European IP/TV service provider.

## REFERENCES

[1] Banerjee A., Kumar V. Chandola V., "Anomaly detection: a survey," *2009*, pp. 1–58, 2009.

[2] W. Wang and R. Battiti, "Identifying intrusions in computer networks with principal component analysis," in *International Conference on Availability, Reliability and Security*, Vienna, Austria, 2006, pp. 270 - 279.

[3] X. Guan, and X. Zhang W. Wang, "A novel intrusion detection method based on principal component analysis," in *IEEE Symposium on Neural Networks*, 2004, p. 2004.

[4] M.-L., Chen, S.-C., Sarinnapakorn, K., and Chang Shyu, "A novel anomaly detection scheme-based on principal component classifier," in *3rd IEEE International Conference on Data Mining*, Melbourne, Florida, 2003, pp. 353--365.

[5] C. Callegari, "A novel PCA-based Network Anomaly Detection," in *IEEE International Conference on Communications (ICC)*, Pisa, Italy , 2011, pp. 1 - 5.

[6] M. C. et al. Kerman, "Event Detection Challenges, Methods, and Applications in Natural and Artificial Systems," in *14th International Command and Control Research and Technology Symposium*, Washington, DC, 2009,

[7] G. Shmueli, "Wavelet-based Monitoring in Modern Biosurveillance," University of Maryland, College Park, Report Working Paper RHS-06-002, 2005.

[8] M. I. Zucchini W., *Hidden Markov Models for Time Series*. USA/FL: Chapman & Hall/CRC, 2009.

[9] M. Stamp. (2004, Jan) Department of Computer Science, San Jose State University. [Online]. http://www.cs.sjsu.edu/faculty/stamp/RUA/HMM.pdf

[10] H.A.L., & van Mechelen, I. Kiers, "Three-way component analysis: Principles and illustrative application," *Psychological Methods*, vol. 6, pp. 84-110, 2001.

[11] Lawrence R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, Feb 1989.