Anomaly Detection via Feature-Aided Tracking and Hidden Markov Models

Satnam Singh, Haiying Tu, William Donat, Krishna Pattipati, Fellow, IEEE, and Peter Willett, Fellow, IEEE

Abstract—The problem of detecting an anomaly (or abnormal event) is such that the distribution of observations is different before and after an unknown onset time, and the objective is to detect the change by statistically matching the observed pattern with that predicted by a model. In the context of asymmetric threats, the detection of an abnormal situation refers to the discovery of suspicious activities of a hostile nation or group out of noisy, scattered, and partial intelligence data. The problem becomes complex in a low signal-to-noise ratio (SNR) environment, such as asymmetric threats, because the “signal” observations are far fewer than “noise” observations. Further, the signal observations are “hidden” in the noise. In this paper, we illustrate the capabilities of hidden Markov models (HMMs), combined with feature-aided tracking, for the detection of asymmetric threats. A transaction-based probabilistic model is proposed to combine hidden Markov models and feature-aided tracking. A procedure analogous to Page’s test is used for the quickest detection of abnormal events. The simulation results show that our method is able to detect the modeled pattern of an asymmetric threat with a high performance as compared to a maximum likelihood-based data mining technique. Performance analysis shows that the detection of HMMs improves with increase in the complexity of HMMs (i.e., the number of states in a HMM).

I. INTRODUCTION

Anomaly detection research is carried out in disparate domains, such as monitoring business news, epidemic or bioterrorism detection, intrusion detection, hardware fault detection, network alarm monitoring and fraud detection [1]. The anomaly detection problem involves large volumes of time series data, which has a significant number of entities and activities. The main goal of anomaly detection is to identify as many interesting and rare events (e.g., intrusions, frauds and unusual business activities) as possible with minimum delay and fewest false alarms. In this paper, we propose to employ the well known Page’s test [2] (an efficient scheme for quickest detection of a change in distribution) to detect an anomaly (or abnormal event) when the pattern to be detected is modeled as a hidden Markov model (HMM). There are two basic ways to detect an anomaly: first, show that the observation process has similarity to an adversary pattern; second, show that the observation process is dissimilar to a benign (or normal) pattern. An intuitive approach to detect an abnormal situation is to use the likelihood ratio, i.e., the ratio of probability density (or mass) function (pdf or pmf) of observations under the assumption of abnormality to the pdf (or pmf) of the same observations under benign (or normal) conditions. Both Bayesian and Neyman-Pearson optimal hypothesis testing use likelihood ratio. However, in the case of sequential testing, the best procedure is the sequential likelihood ratio test; while for quickest detection of a change in distribution, the Page’s scheme uses the cumulative sum (cusum) statistic. Basically, the cusum statistic is a clamped log likelihood ratio (LR) such that it cannot be below zero. If the log likelihood ratio is sufficiently large, an abnormality is declared. The similarity to an adversary pattern amounts to an increase in the numerator quantity within the LR under the hypothesis of abnormality. Likewise dissimilarity to a normal pattern means a decrease in the denominator quantity of the LR under the assumption that no adversary is active. Both cause the LR to rise, which indicates an onset of an abnormal event.

As discussed above, our approach to abnormality detection focuses on the “numerator” of the LR: the degree of likelihood to which the observation sequence matches a HMM that depicts abnormal activity. We use a library of available HMMs to model adversarial activities. The HMM framework is used to compute the posterior probabilities of the hidden states, given a sequence of noisy and partial observations. Hidden Markov models (HMMs) constitute a principal method for modeling partially-observed stochastic processes. The premise behind a HMM is that the true underlying process, represented as a Markov chain depicting the evolution of true transactions as a function of time, is not directly observable (hidden), but it can be probabilistically inferred through another set of stochastic processes (observed transactions, for example). HMMs are a natural choice to detect an anomaly (e.g., a pattern of suspicious activities). Real-world adversary actions or events, such as terrorist attacks, are characterized as partially observable and uncertain signals. Their signals, or electronic signature, are a series of observations. HMMs provide a systematic way to make inferences about the evolution of such partially observable asymmetric threats.

The HMMs can solve three problems: (1) evaluate the probability of a sequence of observed events given a specific model; (2) determine the most likely evolution of an abnormal activity (state sequence) represented by the HMM; and (3) estimate HMM parameters that produce the best representation of the most likely state sequence. Here, we illustrate the capability of feature-aided tracking and HMMs to solve the

This work was supported by Aptima Inc., Woburn, MA 01801, USA. A version of this paper was published at IEEE Aerospace Conference, Big Sky, Montana, Paper no. 11.0404, March 2007.

The authors are with Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269-2157, USA. Haiying Tu now works at Qualtech Systems Inc., Wethersfield, CT 06109, USA. (e-mail: satnam@engr.uconn.edu, tu@teamqsi.com, wdonat@engr.uconn.edu, krishna/willett@engr.uconn.edu).

The expression “asymmetric threats” refers to tactics employed by countries, terrorist groups, or individuals to carry out attacks on a superior opponent, while trying to avoid direct confrontation.
first problem, i.e., evaluate the probability of a sequence of noisy observations given a model of an asymmetric threat. To solve the evaluation problem, we need to develop a signal model that can be used to distinguish between suspicious and real instances of abnormal activities. In doing so, the model must be able to: (1) detect potential abnormal activities (e.g. asymmetric threats) in a highly-cluttered environment; (2) efficiently analyze large amounts of data; and (3) generate hypotheses with only partial and imperfect information. The signal model is a transaction-based model that identifies relationships among nodes in a network to describe its structure and functionality. If we can identify the types of activities (or observations) that an adversary may be involved in, then we can construct a model solely on these.

Feature-aided tracking is the process of collecting data about the features of suspicious entities from one or more sources to enhance the knowledge about them such as age, citizenship or some other details, which are catalogued a priori by expert analysts generating hypotheses. For example, it is generally considered that suicide bombers are young. So, if an analyst is creating a model of a suicide bombing where he needs to describe a transaction involving the suicide bomber; he can describe the features of the suicide bomber such as age, skills, etc. Overall, the HMMs describe the dynamics of a terrorist network by including a priori information of the people involved, the temporal characteristics of the transaction, the geographical location, etc. These features are directly embedded within the underlying states of the HMM, and can be used to distinguish the targets of interest from ambient background noise. Next, we discuss the research work related to anomaly detection of asymmetric threats.

A. Related Work

HMMs are well-known and powerful statistical techniques and they have been widely applied in various fields such as speech recognition, DNA sequence analysis, robot control, fault diagnosis [3], signal detection ([4],[5]), to name a few. Excellent tutorials on HMMs can be found in ([6], [7]). In [8], Smyth described a method for extending the HMMs to allow for unknown states, which cannot be accounted for when the model is being designed.

The anomaly detection problem is widely studied in the machine learning literature. In [1], Fawcett considered anomaly detection as an on-line stream classification problem. The author argued that diverse domains such as intrusion detection, news story tracking, etc., can be naturally expressed in a framework whose central theme is to develop various evaluation metrics that can account for the temporal nature of the problem. Joshi et al. [8] used HMMs to build an anomaly detection system to discriminate between normal and abnormal behavior of network traffic. The authors used the 1999 knowledge discovery in databases (KDD) data set as an example. KDD is defined as an integrated approach to discover knowledge by combining ideas drawn from fields such as databases, machine learning, statistics, visualization, parallel and distributed computing. The authors used the standard Baum-Welch [22] (Expectation Maximization [23]) procedure for HMM parameter training, and performed hypothesis testing using the maximum likelihood (ML) principles to rate the traffic as either normal or having originated from an attack during the recognition phase of the algorithm.

Salvador et al. [10] considered the anomaly detection problem using segmentation or clustering techniques to dynamically divide the time series and to determine a reasonable number of clusters. Further, they considered these clusters as states of a finite state machine to track normal behavior and detect anomalies. The method was applied to the data obtained from the NASA space shuttle. Agarwal et al. [11] presented a holistic approach for simultaneously monitoring a large number of time series (or streams). Their method detects anomalies by applying control chart methodology to normal scores of p-values. The authors considered an adjustable five-parameter empirical Bayesian model for multiple comparisons at each time point. The procedure was illustrated on a bio-surveillance problem. Bay et al. [12] proposed a general solution for time-series data to discover anomalous regimes, which they defined as a change in the functional relationships between the variables, or by the introduction of a previously unseen causal effect. The key idea is to transform the time series data into a set of local models, where each model is trained on a set of small time-bounded data. The framework was used to compare models from the test set to those from the training set in the parameter space to detect anomalies.

Detection of a pattern of abnormal activity is also of a significant interest to the national security community and there are several research groups working on this problem. Godfrey et al. [13] have developed a software tool, termed TerrAlert, which can generate a large number of potential operational schedules via Monte Carlo simulations. It uses Bayesian likelihood theory to adjust the weight on each
schedule based on evidence. Another research effort in modeling asymmetric threats was pursued by Rosen et al. [14] using influence networks (a variant of a Bayesian network) to model suspicious events. The authors developed an influence networks-based software tool, called situational influence assessment module (SIAM), which provides the ability to model causal relationships among seemingly unconnected events and determine their effect on outcomes.

In the context of nation state stability analysis, Schrodt [15] employed HMMs to develop conflict measures based on observed event similarities to historical conflicts. Schrodt used a machine-coding program to perform linguistic parsing of the historical and current news reports. The machine-coded event sequences were generated using a large set of verbs commonly found in international conflicts. A combination of these machine-coded events was used to represent the states of HMM. Schrodt employed Baum-Welch algorithm to learn the HMM parameters from the historical conflict data. After learning the parameters, Schrodt used the forward HMM algorithm to compute the likelihood of observing a sequence of events given the model. The results were presented for several international conflicts, including the Israeli-Palestinian conflict.

The methods proposed in this paper is one of the modules of a software tool, termed the Adaptive Safety Analysis and Monitoring (ASAM) system. Please refer to ([17], [18], [19], [20]) for details on the software architecture, concept of operations, etc. of the ASAM system. Note that the focus of this paper is entirely different from our previous work discussed in [20]. In [20], we focused on information integration using hierarchical and hybrid Bayesian networks (HHBNs) (a hierarchical combination of regular HMMs with no features and Bayesian networks (BNs)). In the HHBN structure, HMMs function in the bottom (observation) layer to report processed evidence to the upper layer BN based on local information.

In this paper, we propose a sequential detection-based approach to detect HMMs, which are used to model asymmetric threats (e.g., terrorist events). As far as we are aware, this is the only work in the literature, which proposes a rigorous statistical framework to detect asymmetric threats. Our work is quite different from the existing literature for anomaly detection in both its application context and the representation details. Most of the previous work on anomaly detection is focused on finding outliers in a time series; however, in our application, an anomaly is a sequence of intelligence transactions. In addition, our HMM state representation is significantly different from that of Schrodt [15]. Our HMM state is depicted using an intelligence transaction, which contains information about transaction type, the entities (people, places, etc.) involved in it and their features. Our probabilistic transaction model also allows the missed detections and false alarms.

In our application, an anomaly is a sequence of very few interesting transactions embedded in a large number of noise (benign) transactions. We cannot apply the existing data mining techniques such as on-line classification [1] or clustering methods [10], because there is not adequate data available for supervised learning of the distribution of interesting transactions. In addition, the anomalies (i.e., asymmetric threats) tend to be sparse and they do not tend to form clusters. A similar conclusion is made by Schrodt [15] in the context of nation state conflict analysis, where the author acknowledges that analyzing the event sequences using clustering techniques [16] has several drawbacks. For example, clustering requires aggregated data whereas HMMs do not require any temporal aggregation. Hence, clustering techniques are infeasible and unrealistic to model threats, which can unfold in a few days to a few years. However, using HMMs we can process the data sequentially; hence HMMs provide a viable framework to model highly adaptive and abrupt threats. Another major drawback of clustering techniques is to determine the start time of a crisis, whereas in the HMMs it can be easily modeled by prefixing an HMM with a background state which represents the events of no crisis [15]. Schrodt further proposed HMMs for international conflict analysis [15].

The superiority of our proposed HMMFA-based anomaly detection method is demonstrated by comparing with a maximum likelihood-based data mining method, termed a Naïve method. The Naïve method models the asymmetric threat as an ergodic HMM with a doubly stochastic transition matrix, which considers the transitions among all the states as equally likely. In this case, all inferences are based on data only. In summary, the contributions of this paper are: (1) a rigorous statistical framework for the detection of asymmetric threats modeled using HMMs; (2) novel HMM state representation and a method to compute the likelihood of modeled activity using concepts from feature-aided tracking, (3) an algorithm for updating the likelihood after skipping a known number of missing transactions, and (4) a detailed performance analysis of the proposed anomaly detection algorithm an comparison with the Naïve method.

B. Organization of the Paper

We are using a HMM variant where observations are associated with arcs of the model instead of the states of the model (regular HMMs). The structure of such HMMs, along with feature-aided tracking, is discussed in Section II. A transaction-based probabilistic model is also discussed in Section II. Section III shows an application of our techniques to a hypothetical model of the development of a nuclear weapons program (DNWP) by a hostile country. A detailed description
of the modeling process using the Testability Engineering and Maintenance System (TEAM$\textsuperscript{®}$) [21] software is provided in Section IV. Section V describes the simulation results for the DNWP model. Some performance analysis is also presented in Section V. Finally, we conclude the paper with a summary and future research directions in Section VI.

II. A TRANSACTION-BASED PROBABILISTIC MODEL

In this section, we first discuss a variant of regular HMMs, where the observations are associated with arcs of the model instead of its hidden states. The state transition matrix of the underlying Markov chain associated with a discrete HMM, parameterized by $\Lambda = (A, B, \Pi)$, is given by

$$A = [a_{ij}] = \left[p(s(k + 1) = S_j | s(k) = S_i)\right]$$

where $s(k)$ is the state at time $k$, and $N$ is the number of states in the HMM. The observation process is represented via the emission matrix:

$$B = [b_{ij}] = \left[p(x_k = X_l | s(k) = S_j, s(k - 1) = S_l)\right]$$

where $x_k$ is the observation at time $k$, and $N_X$ is the number of observation types. The prior probabilities of the Markov states at time $k = 1$ are given by

$$\Pi = [\pi_i] = \left[p(s(1) = S_i)\right] (i \in \{1, 2, \cdots, N_S\})$$

Note in particular that the emission probabilities are slightly different from those of regular HMMs; here the observation is conditioned on both the current and previous states, whereas in regular HMMs the observation is conditioned merely upon the current state. The HMMs can be generalized to allow for continuous emissions, implying that $b_{ijl}$ in (2) could be a probability density function. A convenient choice of the initial probability is the stationary distribution of the underlying Markov chain. The joint probability of a HMM state-observation sequence is

$$p(s_1, s_2, \cdots, s_n, x_1, \cdots, x_{n-1}) = \pi_{s_1} \prod_{k=1}^{n-1} a_{s_k s_{k+1}} \prod_{k=1}^{n-1} b_{s_k x_{k+1} x_k}$$

and this can be considered as its defining property.

In the context of anomaly detection, $A, B,$ and $\pi$ represent, respectively, the probability of moving from the current state of abnormal activity to another (usually denoting an increase in threat), the probability of observing a new suspicious transaction given the current and previous states, and the initial probability, respectively. The forward variable is used to evaluate the probability of abnormal activity, because it is an efficient way to compute the likelihood of a sequence of observations. The forward, backward and termination steps are modified to handle the observation dependence on the previous state and current state. The details of these modified steps are included in Appendix A.

In the context of asymmetric threats, observations are represented by transactions, such as communication, travel, and financing between various entities (people, objects, activities, or places). The features associated with these entities are directly embedded within the underlying states of the hidden Markov models (HMMs). The HMM framework along with feature-aided tracking provides the capability to detect suspicious entities. The observations are collected from an intelligence space as shown in Fig. 1. Each observation (or transaction) is represented as a line connecting two shapes. The shapes (diamond, rectangle and triangle) represent the entities such as people, places and objects. The observations are of various types, such as communication, trust, travel, money and resources. The observation types are shown by different

---

**Fig. 3.** A HMM combined with feature-aided tracking
line types (e.g. solid, dotted, dashed, etc.) connecting the two entities.

To understand our notation for an observation (or transaction), let us consider a simple example. An intelligence analyst receives information that a suspicious person has entered the USA and he has plans to conduct the surveillance of potential targets to carry out a terrorist attack. As shown in Fig. 2, this example is represented by a dashed line (travel transaction) connected with a diamond (suspicious person) and a rectangle (USA). Each entity has generic features (e.g. age, citizenship of a person), which are denoted by $f_1, f_2$ and $f_3$. The features are shown by lines with circled ends.

Fig. 3 shows a HMM combined with feature-aided tracking. The shaded circles represent the states ($s_1, s_2, s_3, s_4, \ldots$, etc.) of a true process. Each state consists of a transaction, which represents only “new” information. A sequence of states (e.g., $s_1s_2s_3s_4$) represents a pattern (a sequence of true transactions), which describes the activities of an adversary. For example, a sequence of true transactions of communication, travel, trust and money type describes the HMM states $s_1, s_2, s_3$ and $s_4$, respectively, as shown in Fig. 3. The dotted transactions inside the HMM states represent previous transactions. For example, $s_2$ contains a solid line (communication transaction) connecting a dotted triangle (object) and a dotted diamond (person), which represents the transaction corresponding to previous state $s_1$. The current and previous states together show the evolution of a true pattern of modeled activity (asymmetric threat). This true “hidden process” is observed through an imperfect and partial observation process (an intelligence database containing noisy information), i.e., a series of observed transactions $(o_1, o_2, o_3, \ldots, \ldots)$, which is shown inside the unshaded circles in Fig. 3. The observations could be of many different types. For example, an observation could be somewhat similar to a true state (e.g., $o_2$ has the same transaction type (travel) as $s_2$, but it has different features). Another type of observation could be having a transaction type different from that of a true state, e.g., $o_3$ has money and resource type transaction where as true state $s_3$ has trust type. Our objective is to detect the hidden “true” pattern, which is a sequence of transactions (shown inside the shaded circles) via the observed process (observations in the unshaded circles). We can infer the existence of a true pattern based upon a set of observations, because the HMM states are statistically related to the noisy observation process.

Next, we explain a transaction-based model which utilizes intelligence information to detect the modeled asymmetric threat. This model is also used to generate the observations for simulation. Fig. 4 shows the discretized transaction space of intelligence information along with various types of transactions under two hypotheses, null $H$ (noise only) and alternative $K$ (HMM in the presence of noise). It is assumed that the HMM becomes active at discrete time index $t_{i0}$. For simplicity, we consider a single observation per scan. However, our formalism can be extended to multiple observations per scan *mutatis mutandis*. Each cell represents a transaction (intelligence) event in the database. The transactions are assumed to be stored in the repository at a specified sampling rate (e.g., one transaction per minute).

While referring to the transactions, we distinguish among true transactions, irrelevant (or null), gated false alarms and clutter. The *true transactions* represent a pattern of threat, which is defined by an analyst in the template (hypothesized) model. The *irrelevant or null* type is defined as those transactions whose details (transaction type, etc.) do not match the template model. A *gated false alarm* is a transaction, which has transaction details (transaction type, entity types and features) similar to those of true transactions, and the number of gated false alarms in the scenario is determined by the probability of gated false alarms ($P_{fa}$). To understand the irrelevant and gated false alarm type transaction, let us consider the example discussed in Fig. 2. In the example, the observation type is of travel type. If we get any observation which is other than travel, such as money or communication, etc. then it will be an irrelevant type transaction. A gated false alarm for this example would be any travel transaction, which would involve a “benign” person whose features are entirely different from those of a suspicious person. Let us say that suspicious person is of age between 20 to 30 years and is a citizen of one of the high-terrorist activity country (say country X). The gated false alarm type observation could have the benign person of the same age group as a suspicious person; however, he may be a citizen of a different country (say country Y). The clutter (or false transaction set) consists of irrelevant and gated false alarms. The objective of a HMM is to detect a pattern of true transactions (a graph inside the shaded circles in Fig. 3) embedded in a large number of extraneous transactions (irrelevant and gated false alarms).

In this paper, we consider the problem of tracking multiple independent HMMs. In the independent case, a binary hypothesis can be constructed for each HMM. Specifically, instead of evaluating the probability of a sequence of observations a specified discrete-time index $k$, given a particular HMM as in the usual evaluation problem, we are interested in a hypothesis testing problem with null hypothesis $H$ as pure noise (“benign transactions”) and alternative hypothesis $K$ as a HMM of interest (viz., “asymmetric threat”) being detected at a specified discrete time index. The details of a single HMM detection scheme based on Page’s test are given in Appendix B. We also propose an algorithm to reduce the inference computations by skipping over irrelevant type transactions, which is presented in Appendix C. Here, we discuss a transaction-based model for null and alternative hypotheses that is used to compute the inference.

A. Null hypothesis (“noise only”)

The null hypothesis is made up of gated false alarms and irrelevant (or null) transactions. Note that we are assuming false alarms as high-threshold false alarms, i.e., they look very
much like true signal transactions. Each gated false alarm has a likelihood of transaction type of \( \tau(k) \) denoted by \( g(\tau(k)) \), and entities have features with specified likelihood functions \( g_j^{(l)}(f_j(k)) \). The subscript \( j \) refers to the number of features, and the superscript \( l \) refers to the number of entities. We assume that the probability of gated false alarm for each epoch is \( P_{fa} \) (Fig. 4), which can be estimated using the large number of benign transactions; and it need not be a constant. The likelihood of transaction type and features for gated false alarms is given by

\[
g_{fa} = g_{\tau}(\tau(k)) \prod_j g_j^{(l)}(f_j(k)) \tag{5}
\]

\[
p(x(k)|H, s(k), s(k-1)) = \begin{cases}
P_{fa} g_{fa} & x(k) = \times \text{ (gated false alarm)} \\
1 - P_{fa} & x(k) = \phi \text{ (irrelevant transaction)}
\end{cases} \tag{6}
\]

B. Alternative hypothesis “HMM in the presence of noise”

When a HMM is present in the noise, we model the transactions according to (7) and (8). Equation (8) has two conditions. Under the first condition, the HMM remains in the current state, and behaves as if it is under the null hypothesis (“noise only”). Under the second condition, the state changes, and the HMM is under the alternative hypothesis (HMM in the presence of noise). We assume that the probability of detection for truth is \( P_d \). The likelihood of features of true transactions is denoted by.

\[
p_T = \prod_j p_j^{(l)}(f_j(k)) \tag{7}
\]

The likelihood of observations under the alternative hypothesis is given by (8).

III. EXAMPLE: DEVELOPMENT OF A NUCLEAR WEAPONS PROGRAM (DNWP)

There are many reasons a country may seek to develop nuclear weapons, but whatever the reasons, the development of nuclear capability by a country has vast implications for the US and its allies. The intelligence community’s ability to detect, analyze, and monitor the development of these programs is essential. The purpose of this model is to describe the evolution of a nuclear weapons program by a hostile country. The model developed herein describes the progress of a nuclear weapons program as a pattern of events grouped into three HMMs: the research and design (RD) HMM; the production of weapons grade material (PWGM) HMM; and the fabrication, test, and integration (FTI) HMM. This model does not attempt to enumerate every possible step or observation, but rather to capture key observable events in the process of developing nuclear weapons. The model is gleaned from open sources ([24], [25], [26], [27], [28]).

On the road to the development of nuclear weapons, a country must perform several tasks. The research and design (RD) HMM involves the processes and equipment used to enrich nuclear material, planning for integration of the weapons into the military portfolio, as well as the design of weapons themselves. The activities of key scientists within a country would reveal that it is developing nuclear weapons. We consider information requests and studies related to nuclear engineering as key indicators. Computer simulations are an integral part of the initial design process of both enrichment and weapons technologies. In addition, a country must gain proficiency with the development and detonation of chemically pure high explosives. The research, experimentation, and testing of these high explosives are observable. Experimentation requires the acquisition of many pieces of specialized test equipment, which, we assume, might be detected. Finally, the unsafeguarded experiment with refinement of nuclear material would be a strong indicator that a country was developing a nuclear weapons program. Fig. 5 shows a Markov chain diagram of the RD HMM.

A country wishing to develop nuclear weapons must acquire or produce weapons grade nuclear material. In the production of weapons grade material (PWGM) HMM, we assume that in order to remain clandestine and have adequate material to create even a meager arsenal, the country chooses to mine, mill, and refine its own uranium. However, the ultimate refinement to weapons grade is allowed to follow the uranium or plutonium enrichment technologies. The enrichment of uranium is assumed to be by means of gas diffusion or centrifuge. A gas diffusion facility would likely be co-located with a large power plant. Meanwhile, a centrifuge enrichment
facility would be very large, and its construction as well as heat signature would likely be observable. The enrichment of plutonium from uranium would require a nuclear reactor and a processing plant. The reactor would have no output power when it was being used to create plutonium and this would likely be observable. Throughout the progress of this group of events, we assume that the acquisition of specialized equipment and resources are observable. Finally, if a country has its own nuclear energy program, its conduct (e.g., lack of cooperation) with International Atomic Energy Agency (IAEA) regulations and inspections is also considered as an indicator of an active nuclear weapons program. The Markov chain diagram of PWGM HMM is shown in Fig. 6.

Once the nuclear material has been refined to weapons grade, the nuclear material must be assembled together with the non-nuclear components into the final nuclear weapon. The weapons grade material must be formed into a shape according to the particular design. This requires lathes, furnaces, an inert gas environment, and other special equipment. A mature nuclear weapons program will have tested, and may have acquired, an inert gas environment, and other special equipment.

A mature nuclear weapons program will have tested, and may have acquired, an inert gas environment, and other special equipment. Therefore, we assume signs of non-nuclear components into the final nuclear weapon. The weapons grade material must be formed into a shape according to the particular design. This requires lathes, furnaces, an inert gas environment, and other special equipment. A mature nuclear weapons program will have tested, and may have acquired, an inert gas environment, and other special equipment. Therefore, we assume signs of deployment of a nuclear weapon, once it has been developed, with boost technology to increase their weapon's yield. Deployment of a nuclear weapon, once it has been developed, requires a delivery system. Therefore, we assume signs of the development of certain delivery systems as indicators of a nuclear weapons program. At some point before the weapon is actually assembled, a political decision must be made to do so. This event is likely to coincide with the implementation of policy, strategic, and military integration of the nuclear weapon.
Fig. 6. Markov chain of PWGM HMM
program. The test of a nuclear weapon is the final state of fabrication, test, and integration (FTI) HMM. The Markov chain depicting the FTI HMM is shown in Fig. 7.

IV. MODELING HMMS VIA TEAMS

We have developed a graphical modeling tool using TEAMS® (Testability Engineering and Maintenance Systems). Using TEAMS®, an analyst can specify the transition probabilities, prior probabilities, and the nodes associated with an HMM state. Each HMM state consists of a transaction (signal) and a transaction is made of two nodes (or entities), as explained in Section III. Fig. 8 shows a HMM state which has two nodes, Nuclear Scientist and Conference.

Different features are available for each type of node. For example, a person type node may have features such as citizenship, age, skills, etc. Each feature must be given a confidence (a number between 0 and 1), which represents a likelihood that the feature information is available in the observation. If the confidence is close to 1, then it indicates that it is more likely that feature information is available in the observations; on the other hand, a confidence value close to 0 indicates that it is less likely that feature information will be available in the observations. Fig. 9 shows the features associated with Nuclear Scientist node. After inputting all the nodes, the analyst can add transactions associated with each state of the HMM. Finally, the model information is exported to the database for further analysis by the HMM and feature-aided tracking algorithm.

V. RESULTS

For the simulations, data is a combination of the underlying hidden states of three HMMs of the DNWP model embedded in background noise from a benign source. The feature-aided tracking and HMM inference model’s parameters (viz., transition probabilities, emission matrix, prior probabilities and feature likelihoods) could be estimated using a learning algorithm such as EM [23] (or Baum-Welch algorithm [22] in the classical HMM phraseology). In a data-scarce environment, such as asymmetric threats, it is doubtful if one can obtain enough training data to learn the model (including the structure and model parameters). Our approach has been to develop an initial model based on our understanding of the domain, and to seek a review and feedback from the subject matter experts (SMEs). We assumed the probability of detection of 80%, and probability of gated false alarm of 3% for generating the observations. In the real world, these probabilities could be estimated from data. The dataset contained 6000 observations for each HMM for a total of 18000 observations. The dataset included true transactions of about 58 and irrelevant/null transaction of about 17406 and gated false alarms of about 536. The following are typical results obtained using the simulations:

A. Results under null hypothesis

Fig. 10 shows the cusum statistic under null hypothesis. The cusum plot mostly remains close to zero; however, peaks occur when there is any kind of match between the pattern defined by the HMM and the observations. The PWGM HMM shows more peaks with high values, because its structure contains many parallel branches (Fig. 6), which increase the probability of matches between a pattern and observations. The cusum statistic for RD HMM (dash-dotted curve) shows higher peaks in the range from 6000 to 12000, because in this range gated false alarms were randomly generated using the true states of RD HMM. Similarly, the cusum statistic for FTI HMM (solid curve) shows higher peaks in the range 12000-18000.

B. Results under alternative hypotheses

Figs. 11, 12 and 13 show the cusum statistic under the alternative hypothesis. The observations were created by
Fig. 8. A HMM state consists of a transaction

Fig. 9. Features associated with Nuclear Scientist node
embedding the true states of the HMM in the noise. The observation range corresponding to PWGM HMM is 1 to 6000, RD HMM is 6001 to 12000 and FTI HMM is 12001 to 18000. The starting point of each HMM detection curve corresponds to the first time the HMM is detected; thus, we believe with certain probability that the modeled suspicious activity is in progress. A peak probability usually results when the pattern evolves into the absorbing state of the concomitant HMM, and we obtain maximum number of signal transactions for this HMM. All the cusum plots eventually level off because the HMM has most likely reached in its terminal state. Further, the false alarms change the numerator and denominator of the likelihood ratio nearly by the same amount; this results in leveling off behavior of the cusum statistic plot. We also compared the results of the proposed hidden Markov model and feature-aided tracking (HMMFA) method with the Naïve method. The Naïve method models the threat as a HMM with a doubly stochastic transition matrix. Hence, the Naïve method may be viewed as a maximum likelihood-based data mining method. Figs. 11-13 compare the cusum statistics of the HMMFA and the Naïve methods. The results demonstrate that HMMFA method is able to detect the modeled threat activity whereas Naïve method completely misses the pattern.

C. What-if analysis under alternative hypothesis

Next, we perform what-if analysis for the Research and Design (RD) HMM. In this analysis, we assume that the intelligence data contains transactions corresponding to only the first twelve states of the HMM and the rest of the pattern (i.e., the last six states of RD HMM) is missing from the observed data. Fig. 14 shows the cusum plot for this scenario. The cusum plot is changed from the plot (a solid curve in Fig. 12) when the entire pattern was present in the dataset. However, the HMM still detects the pattern associated with research and design activity of DNWP model. The fall in the cusum statistic occurs due to an increase in the number of gated false alarms.

Next, we also removed the transactions corresponding to the first two states of the RD HMM in addition to removing six transactions from the end. Fig. 15 shows the cusum plot for new what-if scenario. The plot demonstrates that HMMs combined with feature-aided tracking are still able to detect the pattern. These what-if scenarios illustrates that our method is robust to missing data, and it can detect the pattern even if only a part of the pattern was embedded in the noisy intelligence data.
Fig. 14. Cusum statistic of truncated RD HMM under what-if scenario no. 1

Fig. 15. Cusum statistic of truncated RD HMM under what-if scenario no. 2

D. Performance Analysis

The performance of Page’s test is measured in terms of average run length (ARL, the average number of observations it takes before declaring a detection) under the K and H hypotheses. It is always desired to have a small delay to detection (\(\bar{D}\)), while keeping the average time interval between false alarms (\(\bar{T}\)) as large as possible. Analogous to the conventional hypothesis testing problem, where we wish to maximize the probability of detection while keeping the false alarm rate under a fixed level, the trade-off amounts to the choice of the decision threshold \(h\). The relationship between \(h\) and the ARL is often calculated in an asymptotic sense using first or second order approximations, usually credited to Wald and Siegmund ([29], [30]). We conducted the performance analysis by varying several parameters, such as the probability of gated false alarm (\(P_{fa}\)), probability of detection for truth (\(P_d\)), Page’s test threshold (\(h\)), and the number of states in a HMM to compute the average time interval between false alarms (\(T\)), and the average delay to detection (\(D\)).

We considered the RD HMM, which contains eighteen states, for performance analysis. In order to compare the performance of a HMM with different number of states, we truncated the RD HMM to construct new HMMs containing twelve and fifteen states. The number of observations and Monte Carlo simulations was chosen such that it can achieve low \(\bar{D}\) and high \(\bar{T}\). The observations were created using a \(P_{fa}\) of 3%, and a \(P_d\) of 80%. The average delay to detection \(\bar{D}\) was obtained by varying the Page’s test threshold \(h\) and we used 100 Monte Carlo runs of each having 4000 observations under the alternative hypothesis (when the RD HMM is present in noise) to compute the statistic. The average time interval between false alarms (\(T\)) was obtained by using the same threshold \(h\) range as was used to get \(\bar{D}\). We performed 50 Monte Carlo simulations of each having 50000 observations under null hypothesis (noise only) to get \(\bar{T}\). The transaction feature probabilities were kept between 0.65 to 0.95.

Fig. 16 shows the plot of \(T\) vs. \(D\), which illustrates the exponential relationship between \(T\) and \(D\). The large values of \(\bar{D}\) indicate that the activities take a long time to unfold, and they were detected when the HMMs have observed some transactions corresponding to the HMM states. Fig. 16 shows that the performance of the RD HMM improves when the
number of states is increased. The shifting of (T) vs. (D) plot towards upper-left corner for various numbers of states indicates an improvement in the performance and it improves significantly when the number of states is increased from 12 to 18.

We also compared the performance of the HMMFA and the Naïve methods by plotting T vs. D as shown in Figs. 16 and 17. The T and D values were obtained using the above mentioned procedure. The results demonstrate that HMMFA method achieves higher values of T for as compared to the Naïve method for given D values. The performance results demonstrate that HMMFA method is superior and robust to false alarms as compared to the Naïve method.

VI. CONCLUSION

In this paper, we introduced feature-aided tracking combined with HMMs for analyzing asymmetric threats. HMMs can detect, track, and predict the potential threat activities in the presence of partial and imperfect sequential data. The proposed approach can also serve as a what-if analysis tool by allowing users to modify models (i.e., states in the HMMs) and/or transaction sequences. We utilized a transaction-based probabilistic method to detect and track a pattern consistent with the development of a nuclear weapons program (DNWP). The results associated with the DNWP model were presented using the plots of cusum statistic and the most likely state under the null and alternative hypotheses. The simulation results demonstrate that HMM combined with feature-aided (HMMFA) tracking is an effective method to track asymmetric threats with high accuracy. Performance analysis shows that the detection of HMMs improve with increase in the number of states in a HMM. We have also provided a detailed performance comparison between the hidden Markov models and feature-aided tracking (HMMFA) method and the maximum likelihood-based data mining method for all the HMMs in the development of a nuclear weapons program (DNWP) model. Performance analysis shows that the HMMFA method is superior to the Naïve method in terms of lower false alarms.

In this paper, we assumed that the HMMs are independent. The next challenge is to implement our techniques when multiple HMMs share a data source. In this case, the inference problem becomes essentially a multiple target tracking problem, meaning there is a competition among the HMMs for observations. In the target tracking arena, this is referred to as the problem of data association, or of measurement-origin uncertainty [31]. Here, we assumed that the model parameters are derived from interviews of subject matter experts (SMEs). We are currently expanding the proposed framework to other applications where the data is available (e.g. fault diagnosis) and can learn the parameters from data. Another extension could be to use a factorial hidden Markov modeling (FHMM) framework [32] to track specific entities involved in the threat activities. The FHMM framework provides a capability to factorized the hidden state into multiple layers and it therefore represents the hidden state in a distributed form. In this framework, inference and learning involves computing the posterior probabilities of multiple hidden layers (or states) given the observations. The exact algorithm inference is intractable. However, approximate inference can be computed using Gibbs sampling and structured approximation techniques [32]. In the context of asymmetric threats, the suspicious activities and various entities present in the activity could be represented as different layers of a FHMM. The lowest layer could denote a suspicious activity and the other upper layers could represent the presence of entities in that specific activity. The FHMM framework would allow features such as the people’s identities to become a part of the model in real-time instead of needing to be pre-specified.

VII. ACKNOWLEDGMENTS

This work was supported by Aptima Inc. as part of the NEMESIS (NEtwork Modeling Environment for Structural Intervention Strategies) project. The authors would like to thank Qualtech Systems Inc. for the TEAMS® software.

APPENDIX I

MODIFIED FORWARD, BACKWARD AND TERMINATION STEPS

In this Appendix, we discuss the forward, backward and termination steps when the observations are tied to the arcs of a HMM instead of the states of a HMM. Here, the forward and backward variables at discrete time epoch k, are denoted by \(\alpha_k\)’s and \(\beta_k\)’s respectively. A discrete HMM parameterized by \(\Lambda = (A, B, \Pi)\) is given by (1), (2) and (3). The total number of states in a HMM is denoted by \(N\), the state and observation at discrete time epoch \(k\), are denoted by \(s_k\) and \(x_k\) respectively.

Forward equations:

\[
\alpha_k(j) = p(x_1, ..., x_k, s_k = j | \Lambda)
= \sum_{i=1}^{N} p(x_1, ..., x_k, s_k = j, s_{k-1} = i | \Lambda);
\]
\[
= \sum_{i=1}^{N} p(x_k | s_{k-1} = i, s_k = j) \cdot p(s_k = j | s_{k-1} = i) \cdot p(x_1, x_2, ..., x_{k-1}, s_{k-1} = i | \Lambda)
= \sum_{i=1}^{N} b_{ij} \pi_j \alpha_{k-1}(i);
\]
\[
1 \leq j \leq N; 1 \leq k \leq n; \tag{9}
\]

Termination:

\[
\alpha_n(j) = p(x_1, ..., x_n, s_n = j | \Lambda)
\]
\[
p(x_1, ..., x_n | \Lambda) = \sum_{j=1}^{N} \alpha_n(j) \tag{10}
\]
**Backward equations:**

\[
\beta_k(i) = p(x_k+1, \ldots, x_n | \Lambda, s_k = i) \\
= \sum_{j=1}^{N} p(x_k+1, \ldots, x_n, s_{k+1} = j | \Lambda, s_k = i) \\
= \sum_{j=1}^{N} p(x_k+1 | s_k = i, s_{k+1} = j). \\
p(s_{k+1} = j | s_k = i) \cdot p(x_{k+2}, \ldots, x_n | \Lambda, s_{k+1} = j) \\
= \sum_{j=1}^{N} b_{ij} \alpha_{k+1}(j); \\
1 \leq i \leq N; \quad 0 \leq k \leq n - 1; \\
\]

where \( \beta_n(i) = 1. \)

**APPENDIX II**

**HMM DETECTION SCHEME**

Page’s test [2], also known as the cumulative sum or cusum procedure, is an efficient change detection scheme. A change detection problem is such that the distribution of observations is different before and after an unknown time \( n_0 \); and we want to detect the change, if it exists, as soon as possible. Casting it into a standard inference framework, we have the following hypothesis testing problem

\[ H: \quad x(k) = v(k) \quad 1 \leq k \leq n \]
\[ K: \quad x(k) = z(k) \quad n_0 \leq k \leq n \]

where \( x(k) \) are observations and \( v(k) \) and \( z(k) \) are independent identically distributed (i.i.d.) stochastic processes, with probability density functions (pdf) denoted by \( f_H \) and \( f_K \) respectively. Note that under the \( K \) hypothesis, the observations are no longer a stationary random sequence: their distribution has a switch at \( n_0 \) from to \( f_H \) to \( f_K \).

The Page decision rule, which can be derived from the generalized likelihood ratio test (GLRT), amounts to finding the stopping time \( (N_T) \) when the observations are i.i.d., standard recursion for the Page’s test can be easily written as

\[ N_T = \arg \min_n \{ S_n \geq h \} \]  

in which

\[ S_n = \max \{ 0, S_{n-1} + g(x_n) \} \]  

and

\[ g(x_n) = \ln \left( \frac{f_K(x_n)}{f_H(x_n)} \right) \]  

is the update nonlinearity. Page’s recursion assures that the test statistic \( S_n \) is "clamped" at zero; i.e., whenever the log likelihood ratio (LLR) of current observation would make the test statistic negative (which happens more often when \( H \) is true), Page’s test resets to zero. Thus, operationally, Page’s test is equivalent to a series of sequential probability ratio tests (SPRTs) with upper and lower thresholds \( h \) and 0. Whenever the lower threshold 0 is crossed, a new SPRT is initiated from the next sample until the upper threshold \( h \) is crossed.

Consider the case when \( f_H \) and \( f_K \) are general non-i.i.d. probability measures. In compact form, we can write, in a manner similar to the standard Page’s recursion (14),

\[ S_n = \max \{ 0, S_{n-1} + g(n; k) \} \]  

where

\[ g(n; k) = \ln \left( \frac{f_K(x_n|x_{n-1}, \ldots, x_k)}{f_H(x_n|x_{n-1}, \ldots, x_k)} \right) \]  

and \( x_k \) is the first sample after the last reset, i.e., \( S_{k-1} = 0 \).

For the hidden Markov model, the existence of the forward variable, together with its recursive formula as discussed here, enables efficient computation of the likelihood function of an HMM. Specifically, the likelihood function of an HMM with parameter triple \( \Lambda \) could be written as

\[ f(x_1, x_2, \ldots, x_K | \Lambda) = \sum_{i=1}^{N} \alpha_k(i) \]  

where \( N \) is the total number of states and the \( \alpha_k \)'s are the forward variables defined in (9). Now the conditional probability is readily solved as

\[ f_j(x_k|x_{k-1}, \ldots, x_1) = f_j(x_{k-1}|x_{k-2}, \ldots, x_1, \lambda_j) = \frac{\sum_{i=1}^{N} \alpha_k(i)}{\sum_{i=1}^{N} \alpha_{k-1}(i)} \]  

where \( j = H: K \).

In practice, it is found that the direct use of the likelihood function as defined in (15) will cause numerical underflow as the number of observations increases. For discrete HMMs, it is easily seen from the definition of the forward variable that the likelihood decreases monotonically (and generally geometrically) with the number of observations. The conditional likelihood function defined in (19) does not suffer from such a numerical problem. We therefore need to recursively compute the conditional likelihood function in (19) without the direct use of the forward variable. This can be achieved by scaling. Define \( \alpha_k^i \) such that \( \alpha_k^i(i) = \alpha_1(i) \), but for \( k > 1 \)

\[ \alpha_k^i(j) = \frac{\sum_{i=1}^{N} \alpha_k^i(i) a_{ij} p(x(k)|i, j)}{\sum_{i=1}^{N} \alpha_k^i(i)} \]  

To summarize, for the quickest detection of HMMs, we propose the following procedure:

1) Set \( k = 1, l_0 = 0 \), where \( l_k \) denotes the LLR at time \( k \).

2) Initialize the (scaled) forward variable \( \alpha_k^i \) using

\[ \alpha_k^i(j|K) = \pi_j, \]
\[ \alpha_k^i(j|H) = \alpha_k^i(1|H) = p(x(k)|H, i, j) \]  

(Under the "noise only" hypothesis \( H \), we model one state HMM with \( \alpha_{ij}^H = 1 \), and \( N_H = 1 \)).

3) For each possible state \( j \) and for both hypotheses \( H \) and \( K \), update the log likelihood ratio

\[ l_k = l_{k-1} + \ln \left( \frac{\sum_{i=1}^{N} \alpha_k^i(i|K)}{\sum_{i=1}^{N} \alpha_k^i(i|H)} \right) \]  

(22)
4) If \( l_k > h \), declare detection of a change, stop; If \( l_k < 0 \), set \( l_k = 0; k = k + 1; \) then go to 2; If \( 0 < l_k < h \), continue.

5) Set \( k = k + 1; \) update the scaled forward variable using (23) and (24):

\[
\alpha_{k+1}'(j|K) = \frac{\sum_{i=1}^{\alpha} \alpha_k'(i|K) a_{ij} \pi p(x(k)|K,i,j)}{\sum_{i=1}^{\alpha} \alpha_k'(i|K)}
\]  

(23)

\[
\alpha_{k+1}'(j|H) = \frac{\sum_{i=1}^{\alpha} \alpha_k'(i|H) a_{ij}^H p(x(k)|H,i,j)}{\sum_{i=1}^{\alpha} \alpha_k'(i|H)}
\]

(24)

\[ \text{then go to } 3. \text{ Here, } N_K \text{ and } N_H \text{ are the number of states in a HMM under } K \text{ and } H \text{ hypotheses.} \]

**APPENDIX III**

**CUSUM UPDATE WITH SKIPPED OBSERVATIONS**

In Appendix B, we discussed the inference algorithm which updates the cusum statistic for every observation. This kind of cusum update is slow and computationally intensive. We can accelerate the cusum update by skipping irrelevant (or null) observations i.e. we just perform the update when the observations are pertinent to the model and take into account for number of irrelevant (or null) transactions among them. To derive the cusum update for skipped observations case, first, we need to differentiate between the forward variable \( \alpha_k(i) \) and scaled forward variable \( \alpha_k'(i) \). Recall from the forward-backward algorithm (9), the forward variable is given by

\[
\alpha_k(i) = p(s(k) = i, X^k_1)
\]

(25)

and the scaled forward variable is given by ((20) in Appendix B)

\[
\alpha_k'(i) = \left( \sum_j \alpha_k'(j) \right)^{-1} \cdot \left( \sum_j \alpha_k'(j) a_{ij} p(x(k)|s(k) = i, s(k-1) = j) \right).
\]

(26)

Here, \( X^k_1 \) denotes the superimposed observations \( x_1 \) through \( x_n \). Let us propose a scaled forward variable as

\[
\alpha_k'(i) = p(s(k) = i, x(k)|X^k_1).
\]

(27)

Then, using (26), the scaled forward variable can be written as

\[
\alpha_k'(i) = \left( \sum_j p(s(k) = j, x(k)|X^k_1) \right)^{-1} \cdot \sum_j p(s(k) = j, x(k)|X^k_1) \cdot p(s(k+1) = i|s(k) = j) \cdot p(x(k+1)|s(k+1) = i, s(k) = j)
\]

\[
= (p(x(k)|X^{k-1}_1))^{-1} \cdot \sum_j p(x(k+1), s(k+1) = i, s(k) = j, x(k)|X^k_1) = p(x(k+1), s(k+1) = i, s(k) = j|X^k_1)
\]

\[
\text{so the conjectured scaled forward variable given by (26) is proven to be exactly what is desired, by induction. Further (26) can be used to check the cusum expression:}
\]

\[
\prod_{r=1}^{k} \left( \sum_i \alpha_k'(i) \right) = \prod_{r=1}^{k} p(x(r)|X^{r-1}_1) = p(X^r_k).
\]

(28)

Now let us consider that we have a sequence of \( Q \) “null” observation epochs. At the beginning, we have \( \alpha_k'(i) \) and \( p(X^k_1) \) for both hypotheses (the latter is available from the cusum). We need to compute \( \alpha_{k+Q}'(i) \) and \( p(X^{k+Q}_1) \). Consider

\[
p(s(k + Q), X^{k+Q}_1|X^k_1)
\]

\[
= (p(x(k)|X^{k-1}_1))^{-1} \cdot (p(s(k + Q), X^{k+Q}_1|X^{k-1}_1))
\]

\[
= (p(x(k)|X^{k-1}_1))^{-1} \cdot \left( \sum_{k+Q-1}^{k+Q} p(s_{k+Q}^{k+Q}, X^{k+Q}_1, x(k)|X^{k-1}_1) \right)
\]

\[
= (p(x(k)|X^{k-1}_1))^{-1} \cdot \sum_{k+Q-1}^{k+Q} p(x(k+n)|s(k+n)) \cdot p(s_{k+Q}^{k+Q}|x(k), s(k), X^{k-1}_1)
\]

and also have

\[
p(x(k)|X^{k-1}_1) = \left( \sum_j \alpha_k'(j) \right).
\]

(29)

\[
\prod_{n=1}^{Q} p(x(k+n)|s(k+n)) = (1 - P_f a)^Q (1 - P_d)^B,
\]

(30)

where \( B \) is the number of states difference between the states \( s(k) \) and \( s(k+Q) \). For example, if states \( s(k) \) and \( s(k+Q) \) contains three and five transactions respectively, then to get from \( s(k) \) to \( s(k+Q) \) in (say) eleven null observations, we would need 11 non-false-alarms and 2 missed detections. If the probability of detection \( P_f \) is different for different types of transactions, then it can be easily accommodated. Note also
that (30) does not depend on \( s_{k+1}^Q \) and it depends only on \( s(k) \) and \( s(k+Q) \).

Further,

\[
\begin{align*}
\sum_{x} p(s_{k+1}^{Q} | x(k), s(k), X_k^{k-1}) \\
= p(s(k + Q) | x(k), s(k), X_k^{k-1}) \\
= p(s(k + Q) | s(k)),
\end{align*}
\]

which involves the transition matrix raised to the \( Q \)th power and could be easily precomputed. Overall, we have

\[
p(s(k + Q), X_{k+1}^{k+Q} | X_k^k) = \left( \sum_j \alpha'_i(j) \right)^{-1} \cdot \left( \sum_j (1 - P_d)^{t-j} (A^Q)_{ji} \alpha'_k(j) \right) (1 - P_f a)^Q,
\]

where, \((i-j)\) is the number of transactions difference between states \( i \) and \( j \). We use (32) as

\[
p(X_1^{k+Q}) = p(X_1^k)p(X_{k+1}^{k+Q} | X_1^k) \\
= p(X_1^k) \sum_i p(s(k + Q) = i, X_{k+1}^{k+Q} | X_1^k).
\]  

For the \( Q \)-step-at-a-time cusum update:

\[
\beta''_{k+Q}(i) = \left( \sum_j p(s(k + Q) = j, X_{k+1}^{k+Q} | X_1^k) \right)^{-1} \cdot \left( p(s(k + Q) = i, X_{k+1}^{k+Q} | X_1^k) \right),
\]

also note that

\[
\beta''_{k+Q}(i) \triangleq \alpha'_k(i); \\
= p(s(k + Q) = i, x(k + Q) | X_1^k),
\]

because they differ by a normalization constant. However, since in all updates the term \( \left( \sum_j \alpha'_k(j) \right)^{-1} \) is used, the normalization to \( \alpha''_k(j) \) causes no problems.

REFERENCES


