A Consensus-Bayesian Framework for Detecting Malicious Activity in Enterprise Directory Access Graphs

Pratyush Uppuluri
Purdue University, USA
puppulur@purdue.edu

Shilpa Noushad
Purdue University, USA
snoushad@purdue.edu

Sajan Kumar
Purdue University, USA
kumar836@purdue.edu

ABSTRACT

This work presents a consensus-based Bayesian framework to detect malicious user behavior in enterprise directory access graphs. We instantiate heterogeneous belief dynamics by treating directories as influence-network agents and users as topics whose access propensities evolve across directories. Directory–directory influence is encoded by a shared matrix W, while each directory carries a user-logic matrix C_i describing dependencies among user access propensities. Malicious behavior is modeled as a cross-SCC logical perturbation in selected directory-specific logic matrices, violating the structural homogeneity conditions that support consensus. We use theorem-guided SCC decomposition from opinion dynamics to characterize healthy convergence regimes and detect anomalies through scaled cross-directory opinion variance. To quantify uncertainty, we introduce a Bayesian anomaly score with both static-prior and online-prior variants. Simulations on synthetic enterprise access graphs show that the detector responds to homogeneity-breaking logical perturbations and provides structurally localized anomaly evidence.

KEYWORDS

Anomaly Detection, Machine Learning, User Behavior Analytics, Graph Theory

ACM Reference Format:
Pratyush Uppuluri, Shilpa Noushad, and Sajan Kumar. 2026. A Consensus-Bayesian Framework for Detecting Malicious Activity in Enterprise Directory Access Graphs. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages.

1 INTRODUCTION

Security threats in both on-premise and cloud infrastructures have become increasingly sophisticated, ranging from ransomware and malware to insider threats and unintentional access breaches [7]. To address this, it is critical to adopt a **proactive security approach** rather than relying solely on **post-incident forensics**. Although existing system-level solutions such as **endpoint protection**, **data encryption**, and **user behavior analytics (UEBA)** offer some protection, encryption imposes performance overheads, especially for data in use. UEBA [3], which focuses on analyzing user-entity access behavior, is particularly suitable for detecting insider threats and behavioral anomalies.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus.

In enterprise environments, users typically operate in logical groups and interact with related entities such as shared directories, repositories, or cloud services. For instance, a team of engineers working on a common project may frequently access a set of GitHub repositories related to their codebase. Similarly, finance or HR personnel might regularly work with internal directories containing payroll, compliance, or onboarding documents. These co-access behaviors give rise to group-specific access patterns and interdependencies, forming natural multi-level relationships between users and resources. This hierarchical structure motivates the use of **graph-based machine learning methods** to model normal behavior and detect deviations indicative of potential security threats.

Traditional UEBA approaches often rely on static baselines or historical data and include machine learning models such as anomaly detectors and LSTM-based deep learning methods [2]. However, these models frequently require retraining due to shifting behavioral patterns. Other alternatives like particle swarm clustering [1] exist, but they also fail to fully leverage multilevel graph structure.

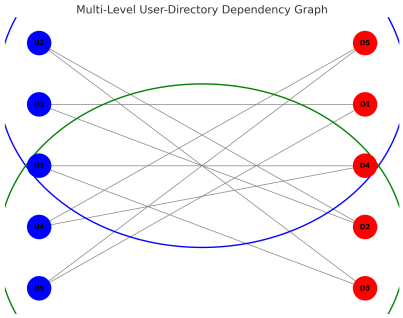
In previous work, we developed a solution combining **community detection** with **anomaly detection** on user-entity access patterns, resulting in multiple patents, as given here - [4–6]. Despite its effectiveness, the approach did not fully exploit the graph hierarchy embedded in user-entity relationships.

To address this, we propose leveraging **multi-level graph structures** and the theoretical foundation of **opinion dynamics** [8] to improve anomaly detection. In our formulation, directories or resources play the role of influence-network agents, while users play the role of topics whose access propensities evolve across directories. The directory-directory influence structure is represented by $G[W]$, and the user-topic logical dependency structure associated with each directory is represented by $G[C_i]$. This mapping may appear counterintuitive because users are often treated as the primary actors in access-control systems. An alternative formulation could instead treat users as influence-network agents and directories as topics, with $G[W]$ modeling user-user influence and $G[C_i]$ modeling directory-directory logical dependencies for each user. In this work, however, we use the complementary view: directories are the influence-network agents, and users are the topics. This choice is useful when the goal is to model how access behavior over a set of users evolves across related resources, and how a change in the user-dependency structure associated with selected directories can disrupt the expected consensus pattern. Opinion dynamics, commonly used in influence networks, therefore provides a formal framework to model evolving patterns of trust and deviation in access behavior. This can be adapted to capture access anomalies over time. The goal of this research is to bridge **opinion dynamics and UEBA** for cloud security by developing an **adaptive UEBA model**

that evolves with changing user behaviors and utilizes underlying relational structures in cloud environments.

2 PROBLEM FORMULATION

Multi-Level Directory–User Interaction Graph in an Enterprise Setting



Consider an enterprise consisting of $n \geq 2$ directories $D_1, D_2, \ldots, D_n$ and $m \geq 1$ users $u_1, u_2, \ldots, u_m$. Users interact with files in these directories through operations such as `read`, `write`, `delete`, or `update`. These interactions induce access patterns that can be used to infer both directory-to-directory and user-to-user relationships.

Directory-Level Graph $G[W]$. We define a directory similarity graph $G[W]$, where the adjacency matrix $W \in \mathbb{R}^{n \times n}$ captures content or behavior-based influence/similarity between directories. The matrix W is assumed to be **row-stochastic**:

$$w_{ii} > 0, \quad \sum_{j=1}^{n} w_{ij} = 1, \quad \forall i \in \{1, \ldots, n\}$$

This ensures that influence or similarity across directories is normalized, with each directory distributing its influence across all others.

User-User Dependency within a Directory via C_i

Each directory D_i is associated with a user-level interaction graph $G[C_i]$, whose structural form consists of a mixture of open and closed strongly connected components (SCCs), as illustrated in Figure 1. This structural pattern is assumed to be shared across all directories $D_1, \ldots, D_n$, though the specific edge weights may differ. The interaction graph for each directory is encoded by a matrix $C_i \in \mathbb{R}^{m \times m}$, where

$$c_{pq,i} \quad \text{quantifies the logical influence of user } u_q \text{ on user } u_p \text{ in } D_i.$$

This matrix C_i is dynamically updated based on observed access interactions among users within directory D_i. For users not directly interacting within D_i, the corresponding entries in C_i reflect steady-state values computed from previous interactions, and are periodically updated to reflect long-term behavioral trends.

Strongly Connected Components (SCCs). Users naturally cluster into interaction communities, such as development teams accessing the same repositories. We represent these as strongly connected components (SCCs) in $G[C_i]$. These SCCs may be:

- **Closed SCCs:** No dependencies on users outside the SCC; internal influence only.
- **Open SCCs:** Depending on at least one user outside the SCC.

Graph Interpretation (See Fig. 1):

- Users $\{u_1, u_2, u_3\}$ form a tight, closed SCC with high mutual influence.
- Users $\{u_5, u_6\}$ form a second SCC with balanced bidirectional dependencies.
- Users u_4 and u_7 show cross-SCC dependencies:
  - $u_4 \to u_5, u_7$ may suggest access escalation or transition to new roles.
  - $u_7 \to u_6$ indicates cascading influence.

Structural Properties (weights) of C_i: Row-Stochasticity and Symmetry. The matrix C_i is assumed to be **row-stochastic** within each SCC:

$$\sum_{q=1}^{m} c_{pq,i} = 1, \qquad c_{pq,i} \geq 0, \qquad \forall p \in \mathrm{SCC}_j(i).$$

Further, we assume bidirectional influence symmetry within SCCs:

$$c_{pq,i} = c_{qp,i} \quad \text{if } u_p, u_q \in \mathrm{SCC}_j.$$

Computing C_i from Access Data. Let $A^i_{pq}(t)$ denote the observed access-based influence from u_q to u_p at time t for directory D_i. Then each entry of $C_i(t)$ is computed as:

$$c_{pq,i}(t) = \frac{A^i_{pq}(t)}{\sum_{k \in \mathrm{CC}_j(i)} A^i_{pk}(t)}.$$

This definition ensures that $C_i(t)$ is **normalized per row within each connected component**, and that the influence distribution from each user is bounded and interpretable over time.

DETECTING ANOMALIES VIA MULTI-LEVEL INTERACTION GRAPHS

We define the multi-level interaction graph as $G(W, C)$, where:

- **User Set:** $U = \{u_1, u_2, \ldots, u_m\}$;
- **Directory Set:** $D = \{D_1, D_2, \ldots, D_n\}$;
- **Logical Dependency Graph per Directory D_i:** $G[C_i]$ represents user-user logical dependencies based on access behavior within directory D_i.

In this formulation, directories play the role of influence-network agents in the opinion-dynamics model, while users play the role of topics. Thus, $W \in \mathbb{R}^{n \times n}$ is the directory-level influence matrix and $C_i \in \mathbb{R}^{m \times m}$ is the user-topic logic matrix associated with directory D_i. The state $x_i(t) \in \mathbb{R}^m$ stores the access propensities of users $u_1, \ldots, u_m$ on directory D_i.

If a user u_k modifies their access behavior at time t, the logical dependency matrix for one or more directories may change:

$$C_i^t \neq C_i^{t-1} \quad \text{if user } u_k \text{ accesses a new directory or substantially alters behavior within } D_i.$$

Anomalies can be detected by observing:

- abrupt structural or weight changes in C_i^t over time;
- deviations from expected user/topic consensus in $G[C_i]$;

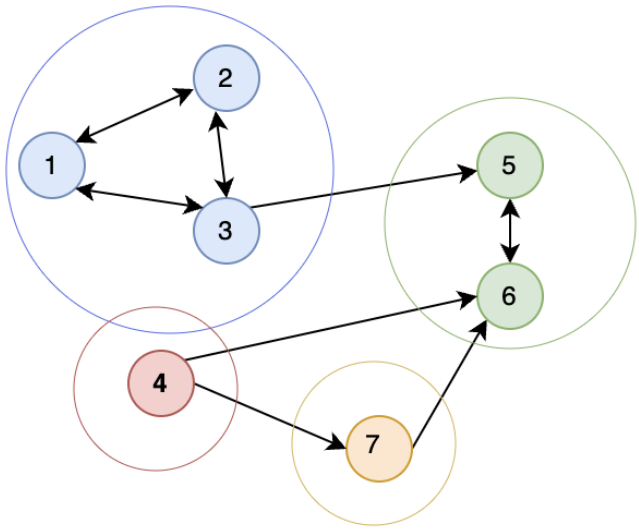


Figure 1: User-user logical dependency graph induced by a directory-specific logic matrix C_i. Nodes represent users; arrows indicate logical influence between user access propensities within directory D_i. Colored clusters highlight strongly connected components (SCCs).

- persistent disagreement in user behavior within strongly connected components.

Anomalous Changes in Logical Dependencies

We consider not only topological changes but also significant internal changes in $C_i(t)$. A simple structural baseline flags a directory D_i if

$$\|C_i(t) - C_i(t-1)\|_F > \delta,$$

where $\|\cdot\|_F$ is the Frobenius norm and δ is a sensitivity threshold.

The opinion-dynamics update used in our model is

$$x_i(t+1) = \sum_{j=1}^{n} w_{ij} C_i x_j(t),$$

where i, j index directories and the coordinates of $x_i(t)$ index users. Thus, changes in $C_i(t)$ alter how user-topic access propensities are coupled within directory D_i, and can change the resulting consensus behavior.

Visualization Example. Suppose $C_i(t)$ for directory D_3 at two time steps is

$$C_i(t_0) = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{bmatrix}, \qquad C_i(t_1) = \begin{bmatrix} 0 & 0.2 & 0.8 \\ 0.4 & 0 & 0.6 \\ 0.3 & 0.7 & 0 \end{bmatrix}.$$

The topology remains the same, but internal shifts in influence weights suggest behavioral drift. If $\|C_i(t_1) - C_i(t_0)\|_F > \delta$, the directory D_i may be flagged for further inspection.

Opinion Dynamics Formulations Aligned with [8]

We summarize the update forms corresponding to the theorem-guided simulation branches. In this section, directories are the influence-network agents and users are the topics.

Singleton User/Topic Dynamics. Let $\mathbf{x}(t) \in \mathbb{R}^n$ represent the opinions of the n directories on a singleton user/topic p. The closed-singleton update has the form

$$\mathbf{x}(t+1) = \Gamma_{pp} W \mathbf{x}(t), \tag{1}$$

where $W \in \mathbb{R}^{n \times n}$ is the row-stochastic directory-level influence matrix and

$$\Gamma_{pp} = \mathrm{diag}(c_{pp,1}, \ldots, c_{pp,n})$$

collects the self-dependencies of user/topic p across directories.

Open Singleton User/Topic with External Dependencies. When user/topic p depends on external user/topics J_p, and those external topics have reached consensus values α_q, the update becomes

$$\mathbf{x}(t+1) = \Gamma_{pp} W \mathbf{x}(t) + \sum_{q \in J_p} \alpha_q \Gamma_{pq}, \tag{2}$$

where

$$\Gamma_{pq} = \mathrm{diag}(c_{pq,1}, \ldots, c_{pq,n})$$

represents the logical influence of user/topic q on user/topic p across directories.

The open-singleton consensus condition requires a common value $\kappa_p \in [-1, 1]$ satisfying

$$\kappa_p (I_n - \Gamma_{pp}) = \sum_{q \in J_p} \alpha_q \Gamma_{pq}. \tag{3}$$

If this compatibility condition fails, consensus for the open singleton is not guaranteed under the theorem conditions.

Multi-Topic Open SCC Dynamics. Let $\mathbf{X}(t) \in \mathbb{R}^{n \times r}$ be the opinion matrix of n directories over r interdependent users/topics $\{p_1, \ldots, p_r\}$, forming an open SCC. For directory D_i and user/topic p, the update used in the theorem-guided simulation is

$$x_i^p(t+1) = c_{pp,i} \sum_{j=1}^{n} w_{ij} x_j^p(t) + \sum_{q \notin \{p_1, \ldots, p_r\}} c_{pq,i} \alpha_q^{(i)}. \tag{4}$$

This captures internal self-dependence, directory-level influence through W, and resolved external user/topic consensus values.

Multi-Topic Closed SCC Dynamics. Let $\mathbf{X}(t) \in \mathbb{R}^{n \times r}$ be the opinion matrix over a closed, strongly connected set of users/topics $\{p_1, \ldots, p_r\}$. The closed-SCC update is

$$x_i^p(t+1) = c_{pp,i} \sum_{j=1}^{n} w_{ij} x_j^p(t) + \sum_{q \in \{p_1, \ldots, p_r\} \setminus \{p\}} c_{pq,i} x_i^q(t). \tag{5}$$

Under the corresponding assumptions of [8], this branch models internal logical coupling among users/topics within the SCC and yields topic-wise consensus when the theorem conditions are satisfied.

Conditions Under Which Consensus Guarantees No Longer Apply.

- **Closed multi-topic SCC.** If the required shared structural conditions for the user/topic SCC are not satisfied across directories, the corresponding theorem no longer guarantees topic-wise consensus.
- **Singleton user/topic.** If the self-dependency and influence-graph assumptions are not satisfied, the singleton consensus guarantee no longer applies.
- **Open singleton user/topic.** If external user/topics $q \in J_p$ have not reached consensus, or if the compatibility condition in (3) fails, consensus of user/topic p is not guaranteed.
- **Open multi-topic SCC.** If external consensus values are inconsistent or the agent-uniform compatibility condition fails across directories, the theorem no longer guarantees consensus for the open SCC.

3 MAIN RESULTS

3.1 SCC Decomposition and Theorem Assignment

We adopt the decomposition and evaluation framework from [9], where the topic index set $\mathcal{J}$ is partitioned into disjoint blocks $\mathcal{J}_j$, each corresponding to a strongly connected component (SCC) in the logical dependency graph. In our enterprise formulation, these topics correspond to users, and the SCCs are therefore user/topic SCCs induced by the directory-specific logic matrices C_i. This is formalized as:

$$\mathcal{J}_j \triangleq \left\{ \sum_{i=1}^{j} s_{i-1} + 1, \sum_{i=1}^{j} s_{i-1} + 2, \ldots, \sum_{i=1}^{j} s_{i-1} + s_j \right\},$$

where s_j denotes the number of users/topics in SCC $\mathcal{J}_j$.

3.2 Decomposition and Evaluation Pipeline

To analyze opinion dynamics in a modular and theoretically justified manner, we follow these steps:

(1) **SCC Extraction:** For a representative directory-specific logic matrix C_i, build a directed user/topic dependency graph and extract all SCCs.
(2) **Open/Closed Classification:** Mark $\mathcal{J}_j$ as *open* if any user/topic in the block depends on a user/topic outside the block; otherwise, mark it *closed*.
(3) **Local Dependencies:** For each user/topic $p \in \mathcal{J}_j$, define

$$\hat{J}_p = \{q \neq p : C[p, q] \neq 0\}.$$

(4) **External Dependencies:** Set

$$\tilde{J}_j = \left(\bigcup_{p \in \mathcal{J}_j} \hat{J}_p \right) \setminus \mathcal{J}_j.$$

(5) **Dependency Conditions:** Construct a DAG over SCCs and evaluate each block only after the blocks on which it depends have been resolved.
(6) **Theorem Assignment:** Based on the dependency structure and numerical values in C_i, assign the corresponding theorem-guided case: closed multi-topic, closed singleton, open singleton, or open multi-topic.

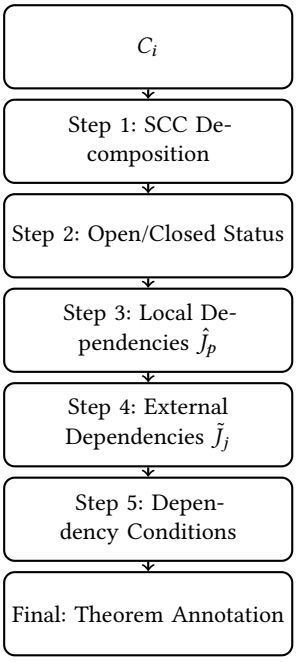


Figure 2: Pipeline for SCC decomposition and theorem annotation over the user/topic dependency graph induced by C_i.

3.3 Example: Theorem Mapping for Paper Simulation

The table in Figure 3 replicates the SCC analysis for the simulation in [9], aligning each user/topic block with its evaluation order and applicable theorem.

	J_index	Topic_Indices_0b	Topics	Status	Local_dependencies	External_dependencies	Dependency_Condition	Apply_Theorem	Minimal_External_Topics
0	J1	[0]	[1]	Closed	{1: []}	[]	[Evaluate First]	Theorem 3	[]
1	J2	[1]	[2]	Open	{2: [1]}	[1]	[J1]	Theorem 3 (via Corollary 2.1)	[0]
2	J3	[2]	[3]	Open	{3: [1, 2]}	[1, 2]	[J1, J2]	Unknown (external consensus required)	[0, 1]
3	J4	[3, 4]	[4, 5]	Open	{4: [2, 5], 5: [2, 4]}	[2]	[J2]	Theorem 4 / Eqn (12) (via Corollary 3.2)	[0, 1]

Figure 3: Annotated SCC blocks with assigned theorems, based on logical structure and dependency ordering in [9].

3.4 Main Algorithm: Bayesian Detection of Anomalies via Opinion Variance Shifts

The proposed anomaly detection algorithm builds upon the structure of opinion dynamics in multi-agent logical systems. In our formulation, directories play the role of influence-network agents, while users play the role of topics. Under stable conditions, opinions converge, or exhibit bounded disagreement, as characterized by the corresponding SCC cases in [8]. Deviations from these behaviors

Algorithm 1 Bayesian Anomaly Score Estimation via Scaled Variance with Online Prior Update

Input: $x_{\text{prev}} \in \mathbb{R}^{n\times d}$, $x_{\text{now}} \in \mathbb{R}^{n\times d}$, prior π_0, scale factor s, exponent α
Output: Anomaly score $\pi \in [0, 1]$
▷ n indexes directories; d indexes the monitored user/topic block.
if x_{prev} is 1D **then**
    $x_{\text{prev}} \leftarrow \text{reshape}(x_{\text{prev}})$
end if
if x_{now} is 1D **then**
    $x_{\text{now}} \leftarrow \text{reshape}(x_{\text{now}})$
end if
$\text{Var}_{\text{cur}} \leftarrow \text{Var}(s \cdot x_{\text{now}})$ ▷ Variance across directories
$\text{Var}_{\text{prev}} \leftarrow \text{Var}(s \cdot x_{\text{prev}})$
$v_{\text{cur}} \leftarrow \frac{1}{d}\sum_j \text{Var}_{\text{cur}}[j]$
$v_{\text{prev}} \leftarrow \frac{1}{d}\sum_j \text{Var}_{\text{prev}}[j]$
$\Delta v \leftarrow \max(v_{\text{cur}} - v_{\text{prev}}, 0)$
$L \leftarrow 1 - \exp(-\alpha \Delta v)$
$\pi_0 \leftarrow \pi_{t-1}$ ▷ Posterior from previous time or previous sweep point
$\pi \leftarrow \dfrac{L\pi_0}{L\pi_0 + (1-L)(1-\pi_0)}$
return π

are flagged as anomalies, especially when accompanied by a rise in opinion variance.

Variance as a Signature of Disruption. We use the per-user/topic opinion variance across directories as a signal of stability or disruption. Under normal operation, especially in closed SCCs, variance remains low and stable. A sudden spike in

$$\Delta v = \max(v_{\text{cur}} - v_{\text{prev}}, 0)$$

indicates that the system is deviating from expected behavior, often due to logic inconsistency, directory-specific drift, or injection of anomalous logical dependencies.

Change in Self-Dependencies $c_{pp,i}$. A key cause of increased variance is a change in the diagonal logic weights $c_{pp,i}$, which govern the self-dependency of user/topic u_p inside the logic matrix associated with directory D_i. In our setting, these weights may be estimated from normalized self-access intensity within a directory, for example

$$c_{pp,i} \propto A^i_{pp},$$

where A^i_{pp} represents the access intensity of user u_p within directory D_i. Increasing $c_{pp,i}$ for only a subset of directories can make the corresponding user/topic more self-dependent in those directories, thereby weakening consensus across the directory influence graph and contributing to higher opinion variance.

Exponential Likelihood Mapping and Bayesian Update. To detect anomalies, we use a scaled exponential likelihood:

$$L = 1 - \exp(-\alpha \Delta v)$$

and apply a Bayesian update:

$$\pi_t = \frac{L\pi_{t-1}}{L\pi_{t-1} + (1-L)(1-\pi_{t-1})},$$

where π_{t-1} is the prior anomaly probability and π_t is the posterior belief. This enables smooth, probabilistic anomaly tracking over time.

Connection to Simulations. In our simulation framework, changes in $c_{pp,i}$ values, new logical edges, or changed SCC dependencies can manifest as observable increases in opinion variance. The anomaly score π_t thus captures both local logic shifts and global structural instabilities.

This allows our model to:

- react to local user/topic behavior change within selected directories,
- detect structural disagreement through theorem-guided SCC outputs,
- update anomaly likelihoods recursively and online.

4 SIMULATIONS AND EXPERIMENTS

To validate our implementation, we replicated the simulation described by the authors in [9]. The implementation reproducing the simulations of [9] and the Bayesian anomaly detection pipeline is available and will be provided upon request. An extended version of this paper, available on arXiv under the same title, includes the section *'Simulation 1: Reproducing the Original Paper's Setup'*, which provides a detailed replication of the authors' environment in [9].

4.1 Bayesian Anomaly Detection: Design, Dynamics, and Inference

We model a cloud access behavior system where directories are represented as influence-network agents and users are represented as topics. Directories evolve their user-access propensity vectors over time using directory-specific user-topic logic matrices C_i and a shared directory-level influence matrix W, governed by the consensus dynamics model of Ye et al. [8]. Anomalous behavior is simulated by modifying the structure of cross-SCC user-topic logical dependencies at time $T = 5$.

Influence Matrix W:

$$W = \begin{bmatrix} 0.281 & 0.334 & 0.334 & 0 & 0 & 0.050 & 0 \\ 0.333 & 0.333 & 0.333 & 0 & 0 & 0 & 0 \\ 0.317 & 0.317 & 0.317 & 0 & 0 & 0.048 & 0 \\ 0.050 & 0 & 0 & 0.450 & 0.500 & 0 & 0 \\ 0.050 & 0 & 0 & 0.450 & 0.500 & 0 & 0 \\ 0.100 & 0.050 & 0 & 0.100 & 0.050 & 0.650 & 0.050 \\ 0 & 0 & 0 & 0 & 0 & 0.050 & 0.950 \end{bmatrix}$$

The matrix W is the directory-level influence matrix.

Logic Matrix $\hat{C}$ (Baseline, Row-normalized):

$$\hat{C} = \begin{bmatrix} 0.143 & 0.429 & 0.429 & 0 & 0 & 0 & 0 \\ 0.429 & 0.143 & 0.429 & 0 & 0 & 0 & 0 \\ 0.429 & 0.429 & 0.143 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.667 & 0.333 & 0 & 0 \\ 0 & 0 & 0 & 0.333 & 0.667 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

We present a simulated cloud system where directories are modeled as influence-network agents and users are modeled as topics. Each directory has an associated user-topic logic matrix C_i, representing user-user logical dependencies based on directory-level access behavior. The influence matrix W governs directory-level influence.

Steady-State Structure at $T = 0$: At initialization, each directory $D_1, \ldots, D_7$ uses the same logic matrix C_{hat}, forming closed user-topic strongly connected components (SCCs):

- Users u_1, u_2, u_3 form a closed SCC.
- Users u_4, u_5 form a second closed SCC with bidirectional dependencies.
- Users u_6 and u_7 are singleton SCCs with self-loops.

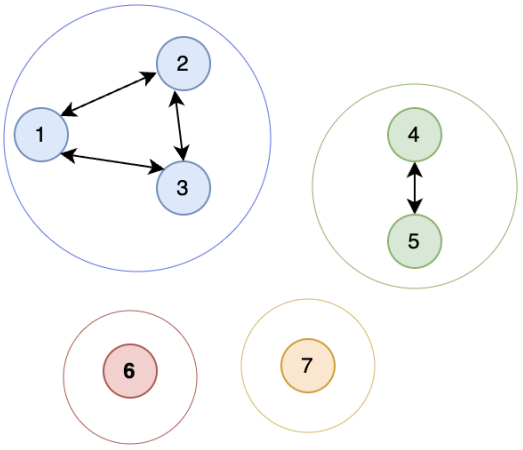


Figure 4: Initial user-topic SCC decomposition at $T = 0$. All directories use the nominal logic matrix C_{hat}, and no cross-SCC user-topic dependency is injected.

Anomaly Injection at $T = 5$: At $T = 5$, we simulate an abnormal access event by modifying the user-topic logic matrices assigned to directories D_4 and D_5. The perturbation strengthens a cross-SCC user-topic dependency from u_2 into the attacked user-topic block $\{u_4, u_5\}$, with tunable weight w_t. The affected logic matrices are row-normalized to produce C_{bar}.

Detection Pipeline:

- For each perturbation weight $w_t \in \{1, 2, 5, 10, \ldots, 1000\}$, we compare the current opinion matrix x_{now} with the baseline x_{prev}.

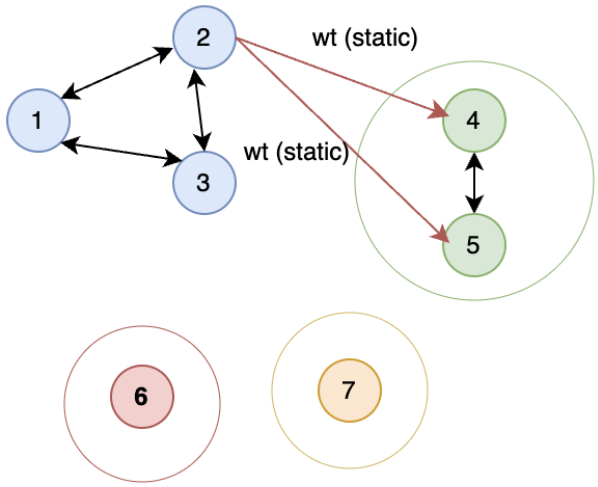


Figure 5: Abnormal access injected at $T = 5$. The red cross-component edge indicates the strengthened user-topic dependency $u_2 \rightarrow \{u_4, u_5\}$ inside the logic matrices assigned to directories D_4 and D_5.

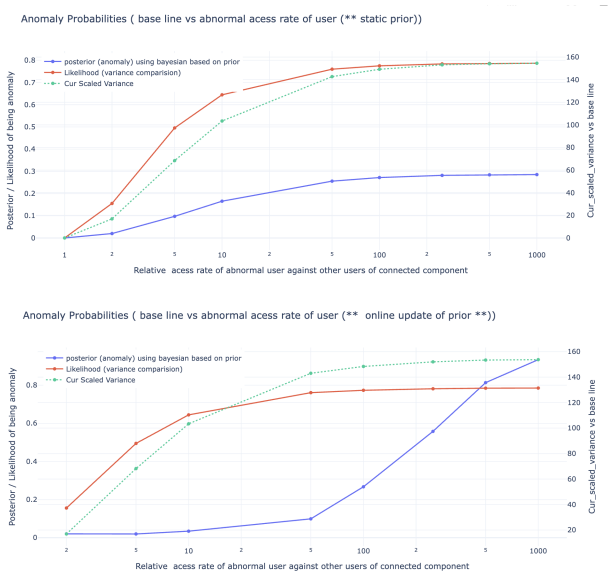


Figure 6: Bayesian Anomaly Detection under Static vs Online Prior Updates. Top: Static prior ($\pi_0 = 0.1$) kept constant. Bottom: Online prior update via posterior at each step. Increasing anomaly weights cause rising variance and anomaly probability. Online inference detects shifts earlier and reacts more sharply.

- A likelihood score is derived from the shift in scaled user/topic variance across directories.
- An anomaly score is computed using a Bayesian update. Two scenarios are evaluated:

(1) **Static prior:** fixed anomaly prior $\pi_0 = 0.1$;
(2) **Online prior:** the posterior from the previous step is reused as the next prior.

This setup captures both structural deviation through modified user-topic logic graphs and behavioral response through probabilistic scoring, enabling a fine-grained analysis of abnormal access propagation in interconnected opinion networks. The anomaly scores for varying perturbation weights are illustrated in Figure 6. The comparison shows a clear relationship between perturbation intensity and anomaly score, highlights the greater responsiveness of the online Bayesian update compared to the static prior, and provides a bounded metric for tracking uncertainty in dynamic enterprise access environments.

4.2 Observations

- As the perturbation weight w_t increases, both scaled variance and anomaly likelihood increase.
- Under the static prior, the posterior rises more gradually because each perturbation level is evaluated against the same fixed prior.
- Under online updating, posterior evidence accumulates across the ordered perturbation sweep, enabling a sharper response at lower weights.

5 CONCLUSION AND NEXT STEPS

5.1 Implementation Considerations

Implementing dependency-aware opinion dynamics requires recursive SCC decomposition, ordered dependency resolution, and conditional theorem application [9]. Correctness therefore depends on consistent handling of graph structure and external influence across time, motivating careful validation in large-scale deployments.

5.2 Open Issues

- **Scalability:** As the number of users and directories grows into the thousands, matrix-based simulations, consensus propagation, and dependency checks become compute-intensive. Optimization via lazy updates, SCC caching, or distributed processing is essential.
- **False Positives:** In real enterprise environments, user behavior often shifts due to benign reasons such as team transfers or new access roles. The system must balance sensitivity to anomalous logic with tolerance to legitimate variance.
- **Control Integration:** Operationalizing detection requires meaningful control responses. Flagging an anomaly is only useful if accompanied by alerting, remediation, or audit hooks.
- **Consensus Drift and Logical Updates:** Over time, even valid behavior may cause opinion drift. Securely updating and sharing C_i matrices across systems introduces challenges in recalibration, versioning, and privacy-aware synchronization.

5.3 Conclusion

This work presents a fusion of graph-based opinion dynamics and Bayesian inference for anomaly detection in user-directory systems. By using directory-specific logic matrices to model user-topic dependencies and grounding belief evolution in convergence theorems, we create a detection signal that is both structurally and statistically aware. The introduction of scaled variance as a volatility signal, combined with Bayesian updating in both static and online forms, allows us to flag anomalous behavior from deviations in consensus. This bridges formal opinion-dynamics modeling and applied behavior monitoring.

5.4 Next Steps

Future efforts will focus on expanding the robustness and deployability of this approach:

- **Enterprise-Scale Simulation:** Scale the system to simulate 1,000+ users and directories, and measure detection responsiveness, convergence behavior, and scoring variance.
- **Semantic Enrichment:** Integrate organizational context, such as roles, groups, and sensitivity levels, into logic construction and directory influence modeling.
- **Hybrid Detection:** Combine logic-based scoring with graph embeddings or time-series analysis to reduce false positives and improve interpretability.
- **Explainability:** Develop traceability tools to identify which users, logic dependencies, or directory interactions contributed most to an anomaly score.
- **Security Integration:** Package the framework into a plug-in module for UEBA systems or SIEM pipelines, with hooks for alerting, policy enforcement, and analyst triage.

Algorithm 2 Dependency-Aware Simulation of Opinion Dynamics

Input: Consensus frame `cdf`, directory-specific user-logic matrices $\{C_i\}$, directory influence matrix W, time steps T
Output: Final consensus values and opinion trajectories
Convert `cdf` to `cdfdct`
Initialize `pending_J_blocks`, `completed_J_blocks`, and `external_consensus_values`
Initialize `results`, `results_simple`; set `iteration` $\leftarrow$ 0, `max_iters` $\leftarrow$ 20
while `pending_J_blocks` not empty and `iteration` < `max_iters` **do**
    `iteration` $\leftarrow$ `iteration` +1
    Identify `ready_J` blocks whose dependency conditions are satisfied
    if `ready_J` is empty **then**
        raise dependency-resolution deadlock error
    end if
    for all $J \in$ `ready_J` **do**
        Extract user/topic indices in J, theorem type, and external dependencies
        Restrict each C_i to the current user/topic block and apply the directory influence update through W
        if Theorem = `Theorem 2` **then**
            Run `simulate_theorem2_multitopic`
        else if Theorem = `Theorem 3` **then**
            Run `simulate_opinion_dynamics_singleton`
        else if Theorem = `Theorem 3 (via Corollary 2.1)` **then**
            Gather external consensus values
            Run `simulate_opinion_dynamics_corollary2`
        else if Theorem = `Theorem 4` or external dependencies exist **then**
            Gather external consensus values
            Run `simulate_theorem4_multitopic`
        else
            `x_final`, `x_hist` $\leftarrow$ None
        end if
        for all user/topic t in J **do**
            Extract the final opinion values for user/topic t
            if values are nearly equal across directories **then**
                Store scalar consensus value in `external_consensus_values`
            else
                Store full opinion vector in `external_consensus_values`
            end if
        end for
        Record outputs in `results` and `results_simple`
        Mark J as completed and remove it from `pending_J_blocks`
    end for
end while
if `iteration` = `max_iters` **then**
    Emit early termination warning
end if
return `results`, `results_simple`

REFERENCES

[1] J. Cui, G. Zhang, Z. Chen, et al. 2022. Multi-homed abnormal behavior detection algorithm based on fuzzy particle swarm cluster in user and entity behavior analytics. *Scientific Reports* 12 (2022), 22349. https://doi.org/10.1038/s41598-022-26142-w Online; accessed 2025-03-15.

[2] Salman Khaliq, Zubair ul Abideen Tariq, and Adeel Masood. 2020. Role of user and entity behavior analytics in detecting insider attacks. In *2020 IEEE International Conference on Cyber Warfare and Security (ICCWS)*. IEEE, 1–6.

[3] Microsoft Security Team. 2024. *What is user and entity behavior analytics (UEBA)?* https://www.microsoft.com/en-us/security/business/security-101/what-is-user-entity-behavior-analytics-ueba Accessed: 2025-03-15.

[4] Pratyush Uppuluri and et al. 2022. Evolution of Communities from Access Patterns. https://patents.google.com/patent/US11468029B2

[5] Pratyush Uppuluri and et al. 2022. Online Anomaly Detection for File Access. https://patents.google.com/patent/US11363042B2

[6] Pratyush Uppuluri and et al. 2022. Overlapping Community Detection Algorithms. https://patents.google.com/patent/US11468124B2

[7] Vectra AI. 2023. 2023 State of Threat Detection Report. https://www.vectra.ai/resources/2023-state-of-threat-detection Accessed: March 14, 2025.

[8] Mengbin Ye, Ji Liu, Lili Wang, Brian D. O. Anderson, and Ming Cao. 2021. Consensus and Disagreement of Heterogeneous Belief Systems in Influence Networks. *IEEE Trans. Automat. Control* 66, 11 (2021), 5266–5281. https://doi.org/10.1109/TAC.2020.3046576

[9] Mingjun Ye, Sergey E Parsegov, Anton V Proskurnikov, and Roberto Tempo. 2018. Consensus and disagreement of heterogeneous belief systems in influence networks. *arXiv preprint arXiv:1812.05138* (2018). arXiv:1812.05138 [cs.SI]