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

Prayush 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. By modeling directories as topics and users as agents within a multi-level interaction graph, we simulate access evolution using influence-weighted opinion dynamics. Logical dependencies between users are encoded in dynamic matrices C_i, and directory similarity is captured via a shared influence matrix W. Malicious behavior is injected as cross-component logical perturbations that violate structural norms of strongly connected components (SCCs). We apply theoretical guarantees from opinion dynamics literature to determine topic convergence and detect anomaly via scaled opinion variance. To quantify uncertainty, we introduce a Bayesian anomaly scoring mechanism that evolves over time, using both static and online priors. Simulations over synthetic access graphs validate our method, demonstrating its sensitivity to logical inconsistencies and robustness under dynamic perturbation.

KEYWORDS

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

ACM Reference Format:
Prayush 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.

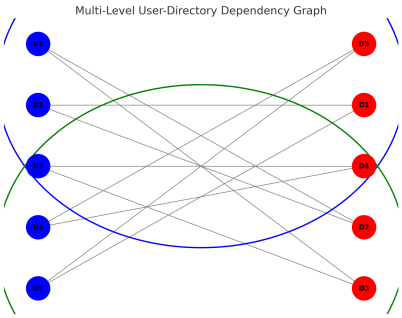
1 INTRODUCTION

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 [8]. 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.

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, 4]. 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 multi-level 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 - [5–7]. 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** [9] to improve anomaly detection. Opinion dynamics, commonly used in influence networks, provides a formal framework to model evolving patterns of trust and deviation among users. 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

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. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). This work is licenced under the Creative Commons Attribution 4.0 International (CC-BY 4.0) licence.

`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 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 2. This structural pattern is assumed to be shared across all directories $D_1, \ldots, D_n$, though the specific user assignments and 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.

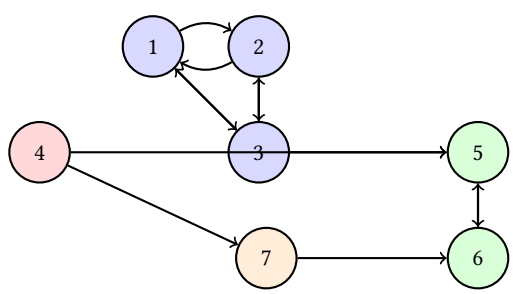


Figure 1: User-user logical dependency graph across directories

Strongly Connected Components (SCCs). Users naturally cluster into interaction communities (e.g., 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 incoming edges from other SCCs; internal influence only.
- **Open SCCs:** Receiving influence from users outside the SCC; indicating external behavioral shift or anomaly.

Graph Interpretation (See Fig. 2):

- 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:
  - $u_4 \rightarrow u_5, u_7$ may suggest access escalation or transition to new roles.
  - $u_7 \rightarrow u_6$ 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:

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

Further, we assume bidirectional influence symmetry within SCCs and no influence between disconnected users:

$$c_{pq,i} = c_{qp,i}, \text{\&if } u_p, u_q \in \text{same SCC}_j$$

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:

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

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.

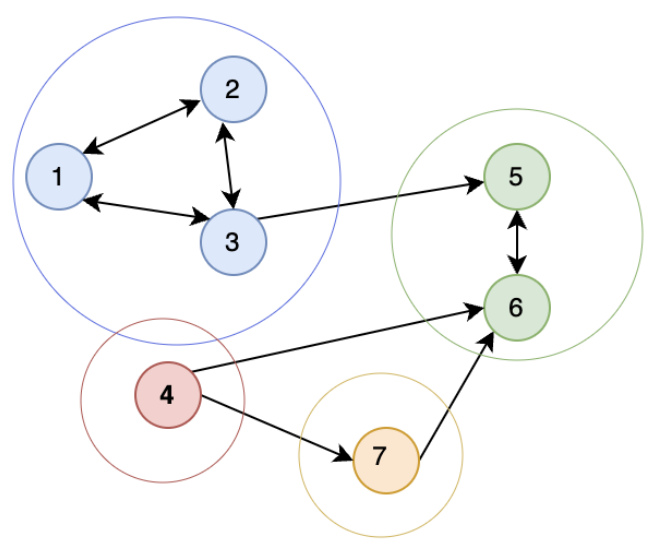


Figure 2: User-user logical dependency graph across directories. Nodes represent users; arrows show influence based on directory-level access behavior. Colored clusters highlight strongly connected components (SCCs).

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 interactions based on access.

If a user u_k modifies their access behavior at time t, the logical dependency matrix is updated:

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

Anomalies can be detected by observing:

- Abrupt structural changes in C_i^t over time.
- Deviations from expected user consensus in $G[C_i]$.
- Persistent disagreements in user behavior within strongly connected components.

Anomalous Changes in Logical Dependencies

We consider not only topological changes but also significant internal changes in $C_i(t)$. An anomaly is flagged if:

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

where $\|\cdot\|_F$ is the Frobenius norm and δ is a sensitivity threshold. These weight changes impact consensus behavior through:

$$y(t+1) = (W \otimes I_m) \cdot \mathrm{vec}(C(t)) \cdot y(t)$$

Visualization Example: Suppose $C_i(t)$ for directory D_3 at two time steps are:

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

The topology remains the same, but internal shifts in influence weights (e.g., from 0.5 to 0.8 or 0.2) suggest behavioral drift. If $\|C_i(t_1) - C_i(t_0)\|_F > \delta$, we flag D_i as anomalous.

By continuously monitoring the evolution of $G(W, C)$, anomalous access behaviors can be detected proactively.

Opinion Dynamics Formulations (Aligned with [9])

We summarize the update rules used in our simulation, each corresponding to the convergence theorems in [9].

Theorem 3: Singleton Topic Dynamics. Let $\mathbf{x}(t) \in \mathbb{R}^n$ be the opinion vector for a singleton topic p. If there are no external dependencies:

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

where $\Gamma_{pp} = \mathrm{diag}(c_{pp,1}, \ldots, c_{pp,n})$ and W is row-stochastic.

Corollary 2.1: Open Singleton Topic. If topic p depends on external topics $q \in J_p$, each with consensus α_q, then:

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

Theorem 3 (Necessity Condition). Topic p reaches consensus α_p if and only if:

$$\kappa_p (I_n - \Gamma_{pp}) = \sum_{q \in J_p} \alpha_q \Gamma_{pq}, \quad \text{for some } \kappa_p \in [-1, 1] \tag{3}$$

Otherwise, consensus is not guaranteed.

Theorem 4: Multi-Topic Open SCC.. For topics $\{p_1, \ldots, p_r\}$ forming an open SCC:

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

Theorem 2: Multi-Topic Closed SCC.. If the SCC is closed and all agents share identical logic over it:

$$x_i^p(t+1) = c_{pp,i} \sum_j w_{ij} x_j^p(t) + \sum_{q \neq p} c_{pq,i} x_i^q(t) \tag{5}$$

When Does Convergence Fail?

- **Theorem 2:** Divergence if logic matrices C_i differ across agents.
- **Theorem 3:** No convergence if Eq. (3) has no solution in $[-1, 1]$.
- **Corollary 2.1:** Divergence if external α_q values are unresolved or inconsistent.
- **Theorem 4:** Disagreement possible if agents have conflicting logic or differing $\alpha_q^{(i)}$.

Opinion Dynamics Formulations (Aligned with [9])

We present mathematical formulations corresponding to our simulation functions, aligned with Theorems 2–4 from [9].

Singleton Topic Dynamics (Theorem 3). Let $\mathbf{x}(t) \in \mathbb{R}^n$ represent the opinions of n agents on a singleton topic $p \in J$ at time t. The evolution is given by:

$$\mathbf{x}(t+1) = \Gamma_{pp} \cdot W \cdot \mathbf{x}(t) \tag{6}$$

where:

- $W \in \mathbb{R}^{n \times n}$ is a row-stochastic influence matrix,
- $\Gamma_{pp} = \mathrm{diag}(c_{pp,1}, \ldots, c_{pp,n})$ is the diagonal matrix of self-dependencies for topic p.

Under strong connectivity and aperiodicity of W, Theorem 3 ensures convergence to a steady state $\mathbf{x}^* \in [-1, 1]^n$.

Open Singleton Topic with External Dependencies (Corollary 2.1). When topic p depends on a set of external topics $J_p \subset J$, each with known consensus α_q, the update becomes:

$$\mathbf{x}(t+1) = \Gamma_{pp} \cdot W \cdot \mathbf{x}(t) + \sum_{q \in J_p} \alpha_q \cdot \Gamma_{pq} \tag{7}$$

where $\Gamma_{pq} = \mathrm{diag}(c_{pq,1}, \ldots, c_{pq,n})$ represents logical influence of topic q on topic p.

Multi-Topic Open SCC Dynamics (Theorem 4). Let $\mathbf{X}(t) \in \mathbb{R}^{n \times r}$ be the opinion matrix of n agents on r interdependent topics $\{p_1, \ldots, p_r\} \subset J$, forming an open SCC. The evolution for agent i and topic $p \in \{p_1, \ldots, p_r\}$ is:

$$x_i^p(t+1) = c_{pp,i} \sum_{j=1}^{n} w_{ij} x_j^p(t) + \sum_{q \notin \{p_1, \ldots, p_r\}} c_{pq,i}\, \alpha_q^{(i)} \tag{8}$$

with:

- $c_{pp,i}$ and $c_{pq,i}$ from agent-specific logic matrix C_i,
- $\alpha_q^{(i)}$ representing scalar or vector consensus values for external topics,
- $x_i^p(t)$ is the opinion of agent i on topic p at time t.

This corresponds to Equation (14) in Theorem 4 of [9] and supports modeling of joint logical influence and external dependency resolution.

Multi-Topic Closed SCC Dynamics (Theorem 2). Let $\mathbf{X}(t) \in \mathbb{R}^{n \times r}$ be the matrix of opinions over a closed, strongly connected set of topics $\{p_1, \ldots, p_r\}$. If all agents share the same logic submatrix (i.e., $C_i|_{\text{SCC}} = C_j|_{\text{SCC}}$ for all i, j), the update rule is:

$$x_i^p(t+1) = c_{pp,i} \sum_{j=1}^{n} w_{ij} x_j^p(t) + \sum_{q \in \{p_1, \ldots, p_r\} \setminus \{p\}} c_{pq,i} \cdot x_i^q(t) \tag{9}$$

This models internal logical coupling among the topics within the SCC, and under the assumptions of Theorem 2 from [9], the opinions $\mathbf{X}(t)$ converge to topic-wise consensus.

Conditions Leading to Divergence or Disagreement (from [9]):

- **Theorem 2 (Multi-topic Closed SCC)** requires all agents to share the same logic submatrix over the topics in the SCC.
  Violation: If $C_i|_{\text{SCC}} \neq C_j|_{\text{SCC}}$ for some $i \neq j$, then consensus is not guaranteed; topic-wise disagreement may persist.
- **Theorem 3 (Singleton Topic)** assumes the topic has no logical dependency on other topics and that the self-dependency values $c_{pp,i} \in (0, 1]$ are strictly positive.
  Violation: If $c_{pp,i} = 0$ for any agent i, or if W is not strongly connected and aperiodic, convergence may fail.
- **Corollary 2.1 (Open Singleton Topic)** requires external topics $q \in J_p$ to have already converged to consensus values α_q.
  Violation: If any dependent topic q has not reached consensus or exhibits disagreement, topic p may not converge.
- **Theorem 4 (Multi-topic Open SCC)** allows different agents to have different logic matrices. However, convergence requires:
  (1) The SCC is closed with respect to external dependencies once α_q values are injected,
  (2) The joint system (logic and influence) has bounded self-dependencies and stable external inputs.
  Violation: If external consensus values $\alpha_q^{(i)}$ are inconsistent across agents or if agents have conflicting logic structures (e.g., competing interdependencies), the SCC may exhibit disagreement or oscillatory behavior.

3 MAIN RESULTS

3.1 SCC Decomposition and Theorem Assignment

We adopt the decomposition and evaluation framework from [10], 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. 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 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 each logic matrix C_i, build a directed graph and extract all SCCs.
(2) **Open/Closed Classification:** Mark $\mathcal{J}_j$ as *open* if any topic depends on external topics; otherwise, mark it *closed*.
(3) **Local Dependencies:** For each topic $p \in \mathcal{J}_j$, define

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

(4) **External Dependencies:** Set $\tilde{J}_j = \bigcup_{p \in \mathcal{J}_j} \hat{J}_p \setminus \mathcal{J}_j$
(5) **Dependency Conditions:** Construct a DAG over SCCs; evaluate each block only after its DAG predecessors.
(6) **Theorem Assignment:** Based on the dependency structure and numerical values in C_i, assign Theorem 2, 3, Corollary 2.1, or 4.

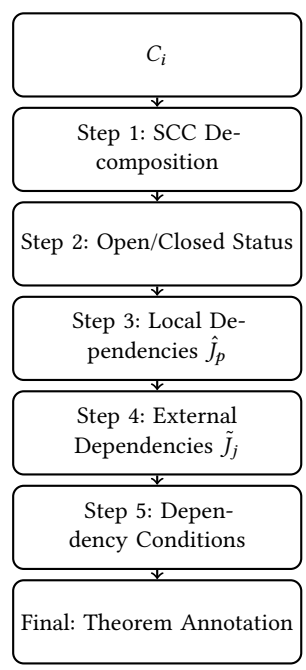


Figure 3: Pipeline for SCC decomposition and theorem annotation.

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 $\pi \in [0, 1]$
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
$\text{Var}_{\text{cur}} \leftarrow \text{Var}(s \cdot x_{\text{now}})$ ▹ Current variance (scaled)
$\text{Var}_{\text{prev}} \leftarrow \text{Var}(s \cdot x_{\text{prev}})$
$v_{\text{cur}} \leftarrow \frac{1}{d} \sum_j \text{Var}_{\text{cur}}[j]$
$v_{\text{prev}} \leftarrow \frac{1}{d} \sum_j \text{Var}_{\text{prev}}[j]$
$\Delta v \leftarrow \max(v_{\text{cur}} - v_{\text{prev}}, 0)$
$L \leftarrow 1 - \exp(-\alpha \cdot \Delta v)$ ▹ Exponential likelihood mapping
$\pi_0 \leftarrow \pi_{t-1}$ ▹ Posterior from previous time
$\pi \leftarrow \dfrac{L \cdot \pi_0}{L \cdot \pi_0 + (1 - L)(1 - \pi_0)}$ ▹ Bayesian update
return π

3.3 Example: Theorem Mapping for Paper Simulation

The table in Figure 4, replicates the SCC analysis for the simulation in [10] by authost themselves, Reference 29, aligning each 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 4: Annotated SCC blocks with assigned theorems, based on logical structure and dependency ordering in [10].

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. It assumes that under stable conditions, opinions converge (or exhibit bounded disagreement) as characterized by Theorems 2–4 in [9]. Deviations from these behaviors are flagged as anomalies, especially when accompanied by a rise in opinion variance.

Variance as a Signature of Disruption. We use the per-topic opinion variance across agents as a signal of stability or disruption. Under normal operation—especially in closed SCCs (Theorem 2 or 3)—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, agent drift, or injection of anomalous entities.

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 how much agent i relies on external opinions. In our setting, these are modeled as:

$$c_{pp,i} \propto a_{ii}$$

where a_{ii} represents the normalized self-access intensity of user i within a directory (e.g., read/write frequency). An increase in a_{ii} makes agent i more self-reliant, thereby dampening consensus behavior and causing divergence from the group. This shift directly contributes to higher opinion variance and will be clearly reflected in simulation outputs.

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

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

and apply a Bayesian update:

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

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 due to increased a_{ii}, or structural disagreements (e.g., new logical edges or SCC dependencies), both manifest as observable increases in opinion variance. The anomaly score π_t thus encapsulates both local logic shifts and global structural instabilities.

This allows our model to:

- React to local agent behavior change (via $c_{pp,i}$),
- Detect structural disagreement (via Theorem 3/4 outputs),
- Update anomaly likelihoods recursively and online.

4 SIMULATIONS AND EXPERIMENTS

To validate our implementation, we replicated the simulation described by the authors in [10]. The implementation reproducing the simulations of [10] and the Bayesian anomaly detection pipeline is publicly available at *Graph-Machine-Learning* 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 [10].

4.1 Bayesian Anomaly Detection: Design, Dynamics, and Inference

We model a cloud access behavior system where each directory is represented as a topic and users as agents. Agents evolve their opinions over time using logic dependency matrices C_i and a shared influence matrix W, governed by the consensus dynamics model of Ye et al. [9]. Anomalous behavior is simulated by modifying the structure of inter-topic logical dependencies at time $T = 5$.

Influence Matrix W:

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

D1 → small influence from D6, D2 unchanged, D3 sees D6, D4/D5 influenced by D1, D6 touches all, D7 receives only from D6.

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 depict a simulated cloud system where users are modeled as agents and directories as topics. Each directory has an associated logic matrix C_i, representing topic-topic dependencies based on user activity. The influence matrix W governs inter-directory similarity.

Steady-State Structure at $T = 0$: At initialization, each directory (D1–D7) uses the same logic matrix C_{hat}, forming closed strongly connected components (SCCs):

- D1–D3: Mutual influence among Users 1, 2, 3.
- D4–D5: Bi-directional mutual influence between Users 4 and 5.
- D6, D7: Isolated users with self-loop only.

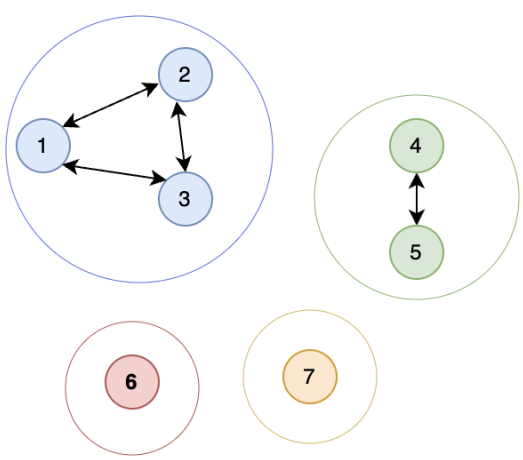


Figure 5: Initial SCC decomposition at $T = 0$: no cross-component influences. All directories use logic matrix C_{hat}.

Anomaly Injection at $T = 5$: At $T = 5$, we simulate an abnormal access event: Users 2 and 3 in D1–D3 begin influencing D4 and D5 (Users 4 and 5) with tunable weight wt. The logic matrices of D4 and D5 are modified and row-normalized to reflect this structural anomaly, producing a new matrix C_{bar}.

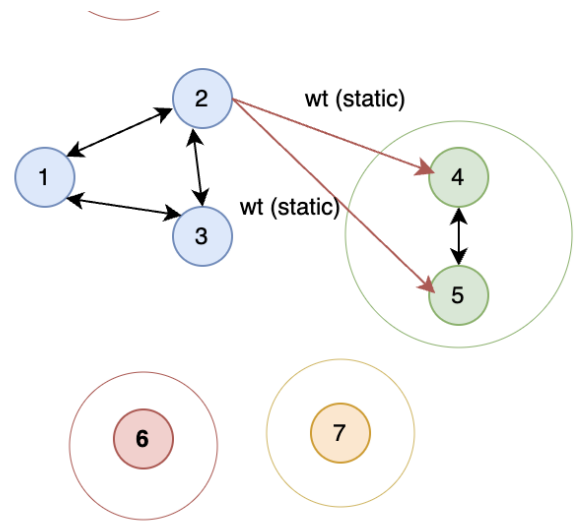


Figure 6: Abnormal access injected at $T = 5$: static cross-component influence from User 2 to Users 4 and 5. Affected directories switch to logic matrix C_{bar}.

Weighted Anomaly Variation: We vary the influence weight wt $\in \{1, 2, 5, 10, \ldots\}$ to study its effect on divergence. Each setting results in a different logic matrix $C_{\text{bar}}^{(\text{wt})}$. For large weights, the influence of abnormal users dominates and alters D4/D5 behavior significantly.

Key Notes:

- The rest of the directories (D1–D3, D6, D7) retain their original logic structure C_{hat}.
- The simulation pipeline checks the effect of anomaly injection on opinion evolution using Theorem 3 and scaled variance drift.

Perturbed Matrix $\bar{C}$ at $w_t = 2$:

$$\bar{C}^{(w=2)} = \begin{bmatrix} \cdots \\ \cdots \\ \mathbf{0} & \mathbf{0.571} & \mathbf{0} & \mathbf{0.143} & \mathbf{0.286} & \cdots \\ \mathbf{0} & \mathbf{0.571} & \mathbf{0} & \mathbf{0.286} & \mathbf{0.143} & \cdots \\ \cdots \end{bmatrix}$$

Perturbed Matrix $\bar{C}$ at $w_t = 50$:

$$\bar{C}^{(w=50)} = \begin{bmatrix} \cdots \\ \cdots \\ \mathbf{0} & \mathbf{0.971} & \mathbf{0} & \mathbf{0.010} & \mathbf{0.019} & \cdots \\ \mathbf{0} & \mathbf{0.971} & \mathbf{0} & \mathbf{0.019} & \mathbf{0.010} & \cdots \\ \cdots \end{bmatrix}$$

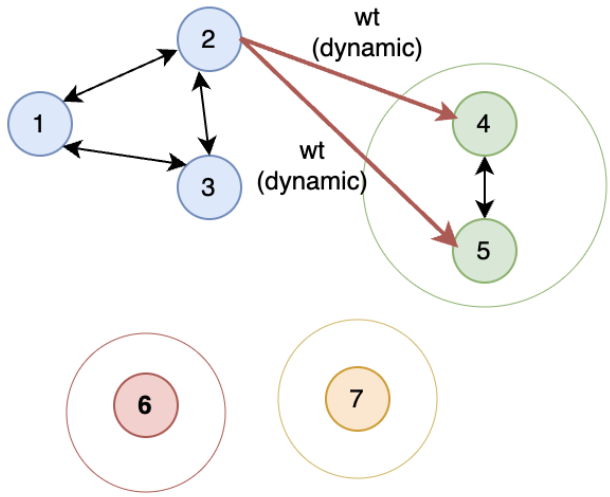


Figure 7: Illustration of dynamic weighted influence: cross-component edges from User 2 to D4/D5 are varied. Logic matrix becomes $C_{\text{bar}}^{(\text{wt})}$.

Bold rows highlight the abnormal influence injected from User 1 to D4, D5. As w_t increases, the normalized logic weights heavily shift toward user 1.

Setup at $T = 0$:

- All directories (topics) begin with stable logical dependency matrices C_i, where subsets of users form closed strongly connected components (SCCs).
- Steady state consensus is reached per directory by simulating opinion dynamics for 5000 steps.
- No anomalous cross-directory access is introduced yet. The system converges with all SCCs and topics stabilized.

Perturbation at $T = 5$:

- We inject an abnormal influence from user/topic 1 into directories D3 and D4 by modifying specific entries of the C_i matrix.
- The updated matrices, row-normalized, reflect strong cross-SCC influence, altering only topics 3 and 4.
- Remaining directories retain their structure:

$$\texttt{C_list_bar} = [C_{\text{bar}}^{(1)}] * 3 + [C_{\text{bar}}^{(2)}] * 2 + [C_{\text{bar}}^{(1)}] * 2$$

Detection Pipeline:

- For each influence weight $w_t \in \{1, 2, 5, 10, \ldots, 1000\}$, we compare the current opinion matrix x_{now} with the baseline x_{prev}.
- A likelihood score is derived from the shift in scaled topic variance.
- An anomaly score is computed using a Bayesian update. Two scenarios are evaluated:

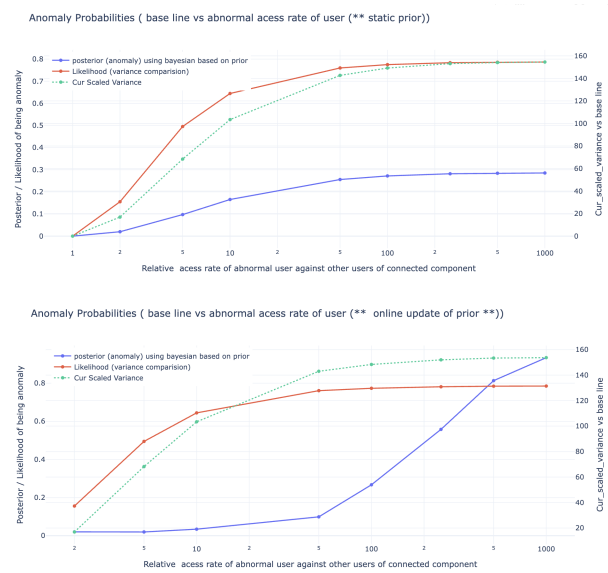


Figure 8: Bayesian Anomaly Detection under Static vs Online Prior Updates. Top: Static prior ($\pi_0 = 0.1$) kept constant. Bottom: Online prior update via posterior at each step. Increasing anomaly weights cause rising variance and anomaly probability. Online inference detects shifts earlier and reacts more sharply.

(1) **Static prior:** fixed anomaly prior $\pi_0 = 0.1$
(2) **Online prior:** $\pi_t \leftarrow \text{posterior}_{t-1}$

This setup captures both structural deviation (via modified logic graphs) and behavioral response (via probabilistic scoring), enabling a fine-grained analysis of abnormal access propagation in interconnected opinion networks.The anomaly score for different wights can be observed in Figure 8, ...

.

4.2 Observations

- As the influence weight w_t increases, both scaled variance and likelihood of anomaly increase.
- Under static prior, posterior saturates early—less sensitive to cumulative divergence.
- Under online updating, posterior compounds over time, enabling more responsive detection 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 [10]. 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. Our 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 novel fusion of graph-based opinion dynamics and Bayesian inference for anomaly detection in user-directory systems. By using logical matrices to model per-topic influence and grounding belief evolution in provable 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 (both static and online), allows us to flag anomalous behavior from deviations in consensus. This bridges the gap between formal multi-agent dynamics 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 latency, convergence behavior, and scoring variance.
- **Semantic Enrichment:** Integrate organizational context (e.g., roles, groups, sensitivity levels) into logic construction and topic similarity.
- **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, logic matrices {C_i}, topic similarity matrix W, time steps T
Output: Final consensus values and opinion trajectories
Convert cdf to cdfdct
Initialize pending_J_blocks, completed_J_blocks, external_consensus_values
Initialize results, results_simple, set iteration ← 0, max_iters ← 20
while pending_J_blocks not empty and iteration < max_iters do
    iteration ← iteration +1
    Identify ready_J blocks with dependencies satisfied
    if ready_J is empty then
        raise deadlock error
    end if
    for all J ∈ ready_J do
        Extract topics, theorem type, and external dependencies
        Slice C_i and W accordingly
        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 deps exist then
            Gather external consensus values
            Run simulate_theorem4_multitopic
        else
            x_final, x_hist ← None
        end if
        for all topic t in topics do
            Extract consensus from x_final
            if nearly equal across agents then
                Store scalar in external_consensus_values
            else
                Store full opinion vector
            end if
        end for
        Record outputs in results, results_simple
        Remove J 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] A. Names. 2023. Insider Threat Detection Based on User and Entity Behavior Analytics. In *Proceedings of the 2023 ACM Conference on Security and Privacy*. 123–134. https://dl.acm.org/doi/10.1007/978-3-031-75764-817 Online; accessed 2025-03-15.

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

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

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

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

[9] 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

[10] 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]