Engineering and Technology Journal e-ISSN: 2456-3358

Volume 11 Issue 08 August-2026, Page No.- 11231- 11249

DOI: 10.47191/etj/v11i08.15, I.F. – 8.595

A Graph-Based Mathematical Modeling of Digital Identity for Anomaly Detection and Future Attack Prediction

Hoon Ko

Department of Computer Science & Engineering, Sunmoon University, Asan, Republic of Korea

ABSTRACT: This study proposes a graph-based framework for anomaly detection and future attack prediction by redefining Digital Identity (DI) as a dynamic and structural security entity rather than a simple authentication credential. The proposed model integrates user, device, network, session, and behavioral information to construct a unique digital fingerprint and represents the relationships among DI components as a graph structure. Normal DI maintains consistent structural patterns, whereas attack situations introduce changes in relational structures and behavioral transition patterns, which are utilized as key indicators for anomaly detection. In addition, temporal correlation analysis is incorporated to extend the framework toward proactive prediction of potential future attacks. Experimental evaluations are conducted using public cybersecurity datasets and DI-based synthetic attack scenarios, demonstrating that the proposed framework achieves higher detection accuracy and improved structural interpretability compared with conventional anomaly detection methods. This study is meaningful in that it reinterprets DI as an active security entity capable of continuous integrity verification, anomaly detection, and future attack prediction.

KEYWORDS: Digital Identity; Digital Fingerprint; Internet of Things; Anomaly Detection; Future Attack Prediction; Random Variable

INTRODUCTION

With the rapid growth of digital technologies, Digital Identity (DI) has become a fundamental component for user authentication and access control in various domains, including online financial systems, e-government services, cloud environments, and IoT/IoMT-based systems [1]. Ensuring the integrity and trustworthiness of user identity has therefore become a critical security requirement. However, conventional DI management approaches mainly rely on static authentication mechanisms based on account credentials, authentication tokens, or single authentication factors, making them vulnerable to various cyber attacks such as account hijacking, session hijacking, device spoofing, and behavioral imitation attacks [2]. In particular, existing approaches generally treat identity as a collection of independent attributes and fail to sufficiently reflect the structural relationships among those attributes [3]. To address these limitations, this study defines DI as a multidimensional attribute space and proposes a mathematical model that forms a unique digital fingerprint for each user. Specifically, the complete attribute set constituting DI is defined as U_{DI} in (1).

$$U_{DI} = P \cup D \cup N \cup S \cup B$$

where (P) denotes personal information, (D) denotes device information, (N) denotes network information, (S) denotes session and authentication information, and (B) denotes behavioral information. The DI of a specific user (u) at time (t) is defined as follows.

$$DI(u,t) \subseteq U_{DI}$$

According to this definition, each user possesses a distinct combination of attributes, which can be interpreted as a unique digital fingerprint [4]. Such a digital fingerprint is not merely authentication information but also reflects the complex structure of user environments and behaviors. Furthermore, this study focuses on the relational dependencies among the attributes constituting (U_{DI}). While normal user activities generally maintain consistent structural patterns, attackers have difficulty perfectly reproducing such relational structures [5]. Therefore, changes in inter-attribute relationships can serve as important indicators of anomalies. Based on this observation, this study models DI not as a simple set of attributes but as a structural object and proposes a method for detecting anomalies and cyber attacks by analyzing structural deviations from normal states. To further clarify the positioning of this study, Table 1 summarizes the differences between existing graph-security approaches, existing identity-centric security frameworks, and the proposed DI-based framework.

Comparison of the proposed framework with existing graph-security and identity-centric security frameworks

Aspect	Existing Graph-Security Approaches	Existing Identity-Centric Security Frameworks	Proposed DI-Based Framework
Primary target	Network nodes, logs, and attack paths	User authentication, authorization, and access control	DI composed of user, device, location, service, and behavior components
Analysis objective	Intrusion detection and attack-path analysis	Authentication, authorization, and policy enforcement	Anomaly detection based on DI state changes and relational inconsistency
Temporal aspect	Limited consideration of identity-state changes	Limited consideration of post-authentication identity transitions	Reflects DI changes and state transitions over time
Relational aspect	Focus on network-level relationships	Focus on user-permission or user-resource relationships	Focus on relationships among heterogeneous DI components
Prediction capability	Mainly detection-oriented	Mainly policy-enforcement-oriented	Supports anomaly detection and future attack vector prediction
Core contribution	Application of graph models to security entities such as hosts, flows, logs, or attack paths	Management of identity, authentication, authorization, and access-control policies	Formalization of DI itself as a dynamic relational graph for post-authentication integrity verification, anomaly detection, and future attack direction estimation

The novelty of this study does not lie in introducing graph analysis, digital fingerprinting, temporal anomaly scoring, or attack-path prediction as isolated techniques. Rather, the contribution lies in integrating these concepts into a DI-specific mathematical framework in which DI itself is treated as a dynamic, relational, and continuously verifiable security object. Existing graph-based anomaly detection approaches generally model network entities, traffic flows, logs, or attack paths as graph nodes and edges. In contrast, the proposed framework defines the identity state itself as a graph composed of heterogeneous DI components, including user, device, location, service, session, and behavior. Existing digital fingerprinting approaches mainly focus on identifying a user or device based on a fixed set of attributes, whereas the proposed model defines a structural digital fingerprint that includes both attributes and relationships among DI components over time. Furthermore, unlike conventional identity-centric security frameworks that primarily focus on authentication, authorization, or access-control enforcement, the proposed framework performs post-authentication integrity verification by examining whether the relationships among DI components remain consistent after successful login. Therefore, the main contribution of this study is the formulation of a DI-centered graph model that unifies structural identity representation, relational anomaly detection, temporal consistency analysis, and future attack direction estimation within a single mathematical framework.

The remainder of this paper is organized as follows. Section 2 defines a set-theoretic and relation-based mathematical model for DI. Section 3 presents a graph-based representation reflecting the structural characteristics of DI. Section 4 describes the anomaly detection and integrity verification model. Section 5 introduces attack modeling and future attack direction prediction based on structural changes. Section 6 presents the use cases and experimental evaluation. Finally, Section 7 concludes the paper and discusses future research directions.

Mathematical Modeling of User-Centric DI

Motivation and Modeling Assumptions

DI is not a static object defined by a single identifier. Rather, it is a dynamic and relational structure formed by the temporal combination of user, device, location, service, and behavioral information [6]. Conventional authentication-oriented identity models mainly focus on verifying the validity of individual attributes, such as user IDs, passwords, or authentication tokens. However, in real online environments, the trustworthiness of a user's identity cannot be determined solely by the validity of a single attribute.

Even when the same user ID and authentication token are used, the corresponding identity state may not be regarded as normal if it is observed together with an unfamiliar device, an IP range or geographical location that differs from the user's historical access records, an abnormal session-generation pattern, or a rapidly changing typing behavior [7]. Therefore, the trustworthiness of DI should be evaluated not by the mere

existence of individual attributes, but by whether the components of user, device, location, service, and behavior form a temporally and relationally consistent structure. In this study, DI is defined not as a simple set of independent attributes, but as a structural state composed of combinations of components belonging to different semantic spaces and the relationships among them.

Typed Attribute Domains for DI

Equation (3) defines the overall product space of DI, it specifies the five typed domains that constitute the DI space.

$$\mathcal{X} = \mathcal{X}_H \times \mathcal{X}_D \times \mathcal{X}_L \times \mathcal{X}_S \times \mathcal{X}_B.$$

- X_H: human-related identity domain,
- X_D: device-related identity domain,
- X_L: location and network domain,
- X_S: service and authentication domain,
- X_B: behavioral identity domain.

Each domain is further represented as a product of multiple attribute spaces, as defined in (4)–(6). The acceptable values of each attribute space depend on the type of the corresponding DI component and the observable logs available in the target system. Categorical attributes such as user ID, email address, MAC address, IMEI, domain name, session ID, token, JWT, and API key are represented as hashed or encoded categorical values. Network-related attributes such as IP address, geolocation, domain, and ASN are obtained from network and access logs and can be represented using categorical encoding, numerical geolocation features, or subnet-level grouping. Behavioral attributes are measured from user activity logs. For example, typing behavior can be represented by typing speed, inter-keystroke interval, error rate, and input rhythm, while access and usage patterns can be represented by access frequency, service-request interval, session duration, and time-of-day features. Biometric information is not stored as raw biometric data in this model. Instead, it is represented by a biometric matching score or an abstract feature vector extracted by a biometric authentication module, such as a normalized similarity score in the range [0,1]. All numerical attributes are normalized before being used in the anomaly detection model.

$$\mathcal{X}_H = \mathcal{A}_{\text{id}} \times \mathcal{A}_{\text{email}} \times \mathcal{A}_{\text{credential}} \times \mathcal{A}_{\text{biometric}}.$$

The device-related domain is defined as follows.

$$\mathcal{X}_D = \mathcal{A}_{\text{MAC}} \times \mathcal{A}_{\text{IMEI}} \times \mathcal{A}_{\text{fingerprint}} \times \mathcal{A}_{\text{OS}}.$$

Similarly, the location, service, and behavioral domains are defined as follows.

$$\begin{aligned} \mathcal{X}_L &= \mathcal{A}_{\text{IP}} \times \mathcal{A}_{\text{geo}} \times \mathcal{A}_{\text{domain}} \times \mathcal{A}_{\text{ASN}}, \\ \mathcal{X}_S &= \mathcal{A}_{\text{session}} \times \mathcal{A}_{\text{token}} \times \mathcal{A}_{\text{JWT}} \times \mathcal{A}_{\text{APIkey}}, \\ \mathcal{X}_B &= \mathcal{A}_{\text{typing}} \times \mathcal{A}_{\text{access}} \times \mathcal{A}_{\text{time}} \times \mathcal{A}_{\text{usage}}. \end{aligned}$$

Here, $\mathcal{A}_{\text{id}}$, $\mathcal{A}_{\text{IP}}$, and $\mathcal{A}_{\text{typing}}$ are not merely attribute names, but spaces of possible values that the corresponding attributes can take. For example, $\mathcal{A}_{\text{IP}}$ denotes the space of observable IP addresses, while $\mathcal{A}_{\text{typing}}$ denotes a behavioral feature space composed of typing speed, inter-keystroke intervals, error rates, and related behavioral characteristics.

DI as a Time-Dependent State

The DI state of user u at time t is defined as a tuple, as shown in (7).

$$DI(u,t) = \big(h_u, d_u(t), l_u(t), s_u(t), b_u(t)\big) \in \mathcal{X}.$$

- $h_u \in \mathcal{X}_H$,
- $d_u(t) \in \mathcal{X}_D$,
- $l_u(t) \in \mathcal{X}_L$,
- $s_u(t) \in \mathcal{X}_S$,
- $b_u(t) \in \mathcal{X}_B$.

The component h_u includes relatively stable attributes, such as user ID, email address, and biometric information. In contrast, $d_u(t)$, $l_u(t)$, $s_u(t)$, and $b_u(t)$ are dynamic components that may change over time, as represented in (8). For example, even for the same user, the access device, IP address, session token, access time, and typing pattern may vary depending on the context [8]. Therefore, DI is represented not as a static identifier, but as a sequence of states observed over time.

$$\mathcal{T}_u = \{DI(u,t_1), DI(u,t_2), \dots, DI(u,t_n)\}.$$

This representation enables DI to be interpreted not as authentication information at a single point in time, but as a dynamic state that can be compared with the accumulated history of a user's normal behavior.

Relational Structure of Identity Components

The core of DI lies in the relationships among its components rather than in the individual attributes themselves. In other words, the device normally used by a particular user, the IP ranges commonly accessed by that device, the service environment in which sessions are generated [9], and the behavioral patterns observed after session creation are not independent of one another. Therefore, this study models DI as a relational structure. The set of relationships among identity components is defined in (9).

$$\mathcal{R} = \{R_{HD}, R_{DL}, R_{LS}, R_{SB}, R_{BH}, R_{HS}\}.$$

- $R_{HD} \subseteq \mathcal{X}_H \times \mathcal{X}_D$: relationship between a user and a device,
- $R_{DL} \subseteq \mathcal{X}_D \times \mathcal{X}_L$: relationship between a device and a location/network,
- $R_{LS} \subseteq \mathcal{X}_L \times \mathcal{X}_S$: relationship between a location and a service session,
- $R_{SB} \subseteq \mathcal{X}_S \times \mathcal{X}_B$: relationship between a service session and behavior,
- $R_{BH} \subseteq \mathcal{X}_B \times \mathcal{X}_H$: relationship between a behavioral pattern and a user,
- $R_{HS} \subseteq \mathcal{X}_H \times \mathcal{X}_S$: relationship between a user and an authentication session.

These relationships may be predefined policy-based rules or empirical relationships learned from a user's historical logs.

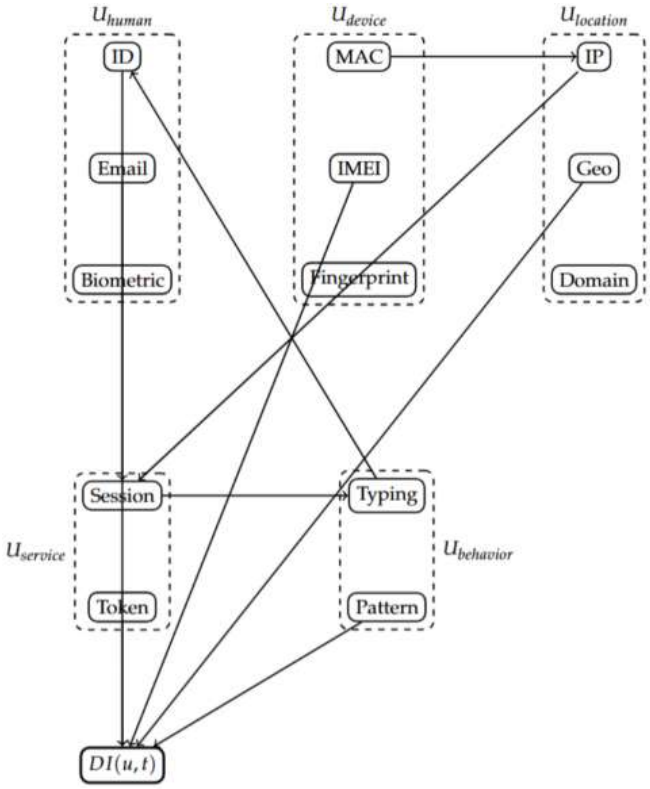

Figure 1. Multi-layer relational structure of user-centric DI

Figure 1 illustrates the multi-layer relational structure of DI. Each dashed box represents a different semantic domain, and each arrow denotes a dependency or constraint relationship between identity components. In this structure, DI is interpreted not as a simple enumeration of ID, MAC address, IP address, and token, but as a relational state in which the user, device, location, service, and behavior are interconnected.

Graph-Based Interpretation of DI

The relational structure described above can also be represented as a graph. The DI graph of user u at time t is defined as shown in (10).

$$G_u(t) = (V_u(t), E_u(t), \omega).$$

Here, $V_u(t)$ denotes the set of nodes corresponding to identity components, and $E_u(t)$ denotes the set of edges representing relationships among the components, as shown in (11).

$$V_u(t) = V_H \cup V_D \cup V_L \cup V_S \cup V_B.$$

Each node represents an individual attribute or an attribute group, such as a user ID, email address, device fingerprint, IP address, session, token, or typing pattern. Each edge represents an observed relationship between two components, as defined in (12).

$$E_u(t) \subseteq V_u(t) \times V_u(t).$$

In addition, $\omega.E_u(t) \to \mathbb{R}^+$ is a weight function that represents the reliability or strength of each relationship. For example, a device–IP combination that has been repeatedly used by the same user over a long period may have a high weight, whereas a newly observed device or a session generated from an abnormal location may have a low weight [10]. This graph representation plays an important role in the subsequent anomaly detection stage. Normal users tend to generate similar graph structures repeatedly over time. In contrast, attacks such as account takeover, session hijacking, device spoofing, and location obfuscation generate new edges or abnormal node combinations that deviate from the existing graph structure [11, 12].

Normal Identity Region and Constraint Function

The normal DI region of user u is defined from the distribution of previously observed normal states. This is expressed in (13).

$$\mathcal{N}_u = \{x \in \mathcal{X} \mid P_u(x) \geq \epsilon \ \wedge \ C_u(x) = 1\}.$$

Here, x denotes an identity state, $P_u(x)$ denotes the probability that the corresponding state is normally observed for user u, and ϵ is the minimum probability threshold required to accept a state as normal. The function $C_u(x)$ indicates whether the identity state satisfies the constraints among identity components, as defined in (14).

$$C_u(x) = \prod_{R_{ij} \in R} \mathbb{I}\left[(x_i, x_j) \in R_{ij}^{(u)}\right].$$

Here, x_i and x_j denote two DI components included in the identity state x, such as user–device, device–location, location–session, or session–behavior pairs. The relation $R_{ij}^{(u)}$ represents the set of acceptable component pairs for user u, which can be defined from historical normal logs or system security policies. The indicator function returns 1 when the observed pair (x_i, x_j) belongs to the acceptable relation set and returns 0 otherwise. Since $C_u(x)$ is defined as a product of all indicator values, $C_u(x) = 1$ only when all relational

constraints are satisfied. If any one relationship is invalid, the product becomes 0, indicating a relational integrity violation. The notation $\mathbb{I}[\cdot]$ denotes an indicator function that returns 1 if the condition is true and 0 otherwise. Thus, $C_u(x) = 1$ means that the identity state x satisfies all predefined relational constraints, whereas $C_u(x) = 0$ means that at least one relationship among the identity components is abnormal. For example, constraint violations may occur in the cases shown in (15).

$$\begin{aligned} (h_u, d_u(t)) &\notin R_{HD}, \\ (d_u(t), l_u(t)) &\notin R_{DL}, \\ (s_u(t), b_u(t)) &\notin R_{SB}. \end{aligned}$$

These cases may correspond to access from an unfamiliar device, an abnormal device–location combination, and an abnormal behavioral pattern after session creation, respectively [13].

Temporal Transition of DI

Since DI changes over time, not only a single identity state but also transitions between states should be analyzed. The identity state transition of user u is represented in (16).

$$DI(u, t_k) \rightarrow DI(u, t_{k+1}).$$

Normal users generally change their identity states within a certain range. For example, they access services from the same device at similar times, generate sessions from similar network ranges, and exhibit usage patterns similar to their previous behavior [14]. This can be represented as the transition probability in (17).

$$T_u(x_{k+1} \mid x_k) = P(DI(u, t_{k+1}) = x_{k+1} \mid DI(u, t_k) = x_k).$$

An abnormal identity state may arise not only when the current state itself is unfamiliar, but also when the transition from the previous state to the current state is abrupt or difficult to explain. Therefore, the temporal anomaly score can be defined in (18).

$$A_{\text{temp}}(u, t_{k+1}) = -\log T_u\big(DI(u, t_{k+1}) \mid DI(u, t_k)\big).$$

The lower the transition probability is, the larger the value of A_{temp} becomes. This indicates that the current identity state deviates from the user's normal transition pattern.

Digital Fingerprint Representation

To be used as input to an anomaly detection model, a DI state must be transformed into a numerical feature space. This representation is defined as a digital fingerprint in (19).

$$FP(u, t) = \phi\big(DI(u, t), G_u(t)\big) \in \mathbb{R}^m.$$

Here, ϕ is a feature mapping function that transforms the identity state and the relational graph into a fixed-length vector. The digital fingerprint $FP(u, t)$ may include the information shown in (20).

$$FP(u,t) = \left[\phi_H(h_u), \phi_D\big(d_u(t)\big), \phi_L\big(l_u(t)\big), \phi_S\big(s_u(t)\big), \phi_B\big(b_u(t)\big), \phi_R\big(G_u(t)\big), \phi_\Delta(u,t)\right]$$

- $\phi_H(h_u)$: user-related features,
- $\phi_D\big(d_u(t)\big)$: device-related features,
- $\phi_L\big(l_u(t)\big)$: location and network features,
- $\phi_S\big(s_u(t)\big)$: service and authentication-session features,
- $\phi_B\big(b_u(t)\big)$: behavior-based features,
- $\phi_R\big(G_u(t)\big)$: relational graph-based features,
- $\phi_\Delta(u, t)$: features representing changes from the previous state.

Therefore, the digital fingerprint is not merely a device fingerprint. Rather, it is a structural identity representation that includes the user, device, location, service, behavior, and the relationships among them [15]. Through this representation, an identity state can be used as input for machine-learning-based anomaly detection, graph-based anomaly detection, or rule-based integrity verification.

Anomaly Score of DI

Finally, the anomaly score of the DI of user u at time t is defined by jointly considering state rarity, relational-constraint violations, and temporal-transition abnormality, as shown in (21).

$$A(u, t) = \alpha A_{\text{state}}(u, t) + \beta A_{\text{rel}}(u, t) + \gamma A_{\text{temp}}(u, t).$$

Here, α, β, and γ are weights that control the relative importance of each term. The term A_{state} represents the abnormality of the current identity state itself, A_{rel} represents the degree of violation of relational constraints among identity components, and A_{temp} represents the abnormality of the transition from the previous state to the current state. The state-based anomaly score can be defined in (22).

$$A_{\text{state}}(u, t) = -\log P_u\big(DI(u, t)\big).$$

The relational anomaly score is expressed in (23).

$$A_{\text{rel}}(u, t) = \sum_{R_{ij} \in \mathcal{R}} w_{ij} \left(1 - \mathbb{I}\left[\big(x_i(t), x_j(t)\big) \in R_{ij}\right]\right).$$

Here, w_{ij} denotes the importance weight of each relationship. For example, in systems where the user–session relationship or the device–location relationship is particularly important, the values of w_{HS} or w_{DL} can be set relatively high. Finally, an identity state is classified as abnormal if the following condition is satisfied.

$$A(u, t) > \tau.$$

Here, τ is the anomaly detection threshold. This formulation differs from conventional static identity models because DI is evaluated not merely by authentication success or failure, but by the relationships among identity components, temporal changes, and the degree of deviation from the user-specific normal identity region (24).

Graph-Based DI Representation

In Section sec.mathematical_modeling, DI was defined as a structural tuple composed of heterogeneous identity components. More specifically, the DI of user u at time t is represented as a state in the product space of human-related, device-related, location-related, service-related, and behavior-related domains as shown in (25).

$$DI(u, t) \in \mathcal{X}_H \times \mathcal{X}_D \times \mathcal{X}_L \times \mathcal{X}_S \times \mathcal{X}_B.$$

Equation (25) indicates that DI is a structural state defined over five typed component spaces. Equivalently, the same identity state can be expressed in tuple form, as shown in (26).

$$DI(u, t) = \big(h_u, d_u(t), l_u(t), s_u(t), b_u(t)\big).$$

Here, h_u, $d_u(t)$, $l_u(t)$, $s_u(t)$, and $b_u(t)$ denote the human-related, device-related, location-related, service-related, and behavior-related components, respectively. However, this tuple-based representation alone is not sufficient to capture the relationships among components or their temporal evolution. Therefore, this section extends the tuple-based formulation and models DI as a probabilistic dynamic graph.

Definition of a Probabilistic DI Graph

The DI graph at time t is defined as follows.

$$G_{DI}^{t} = (V, E^{t}, X^{t}),$$

where V is the set of nodes, E^{t} is the time-dependent set of edges, and X^{t} is the node feature matrix. Based on the component domains defined in Section sec.mathematical_modeling, the node set is defined as the union of heterogeneous identity component sets.

$$V = V_H \cup V_D \cup V_L \cup V_S \cup V_B.$$

Each node $v_i \in V$ represents an identity component, such as a user, device, location, service, or behavior. The node feature matrix is defined as follows (29).

$$X^{t} = \begin{bmatrix} (x_1^t)^{\top} \\ (x_2^t)^{\top} \\ \vdots \\ (x_{|V|}^t)^{\top} \end{bmatrix} \in \mathbb{R}^{|V| \times d},$$

where $x_i^t \in \mathbb{R}^d$ denotes the feature vector of node v_i at time t.

Probabilistic Edge Modeling

In the proposed DI graph, relationships between nodes are not defined deterministically. Instead, each relationship is modeled as a probabilistic variable. Each element of the adjacency matrix A^t is defined as follows (30).

$$A_{ij}^{t} \sim \text{Bernoulli}(p_{ij}^{t}).$$

Here, $A_{ij}^t = 1$ indicates that a relationship exists between nodes v_i and v_j, whereas $A_{ij}^t = 0$ indicates that no such relationship is observed. The edge-existence probability is defined as follows (31).

$$p_{ij}^{t} = \sigma\left(f_{\theta}(x_i^t, x_j^t, r_{ij})\right),$$

where p_{ij}^t is the probability that the relationship exists, r_{ij} is the relationship type, $f_\theta(\cdot)$ is a parameterized relational function, and $\sigma(\cdot)$ is the sigmoid function. The relationship type is defined as follows (32).

$$r_{ij} \in \{\text{uses}, \text{accesses}, \text{located\_at}, \text{exhibits}, \text{authenticated\_by}\}.$$

Accordingly, the probabilistic edge set can be represented as follows (33).

$$E^{t} = \{(v_i, v_j, r_{ij}, p_{ij}^{t}) \mid v_i, v_j \in V\}.$$

This formulation enables DI to be interpreted not as a simple collection of attributes, but as a probabilistic relational structure.

Graph Generation Likelihood

The generation likelihood of the DI graph is defined as follows (34).

$$P(G_{DI}^{t} \mid \theta) = \prod_{i,j} P\left(A_{ij}^{t} \mid x_i^t, x_j^t, r_{ij}; \theta\right).$$

By applying the Bernoulli edge model, the graph likelihood can be written as follows (35).

$$P(G_{DI}^{t} \mid \theta) = \prod_{i,j} \left(p_{ij}^{t}\right)^{A_{ij}^{t}} \left(1 - p_{ij}^{t}\right)^{1 - A_{ij}^{t}}.$$

The corresponding log-likelihood is given by (36).

$$\log P(G_{DI}^{t} \mid \theta) = \sum_{i,j} \left[A_{ij}^{t} \log p_{ij}^{t} + \left(1 - A_{ij}^{t}\right) \log\left(1 - p_{ij}^{t}\right)\right].$$

This value can be used as an indicator of how well the current DI structure matches the learned normal pattern.

Temporal Evolution Model

DI is a dynamic system that changes over time. Therefore, the temporal sequence of DI graphs is defined as follows (37).

$$\mathcal{G}_{DI} = \{G_{DI}^{1}, G_{DI}^{2}, \ldots, G_{DI}^{T}\}.$$

The temporal evolution of the graph can be modeled as follows (38).

$$G_{DI}^{t+1} \sim \mathcal{T}_{\theta}(G_{DI}^{t}),$$

where $\mathcal{T}_\theta(\cdot)$ denotes a parameterized temporal transition model. Under the Markov assumption, the next graph depends only on the current graph.

$$P(G_{DI}^{t+1} \mid G_{DI}^{t}, G_{DI}^{t-1}, \ldots) = P(G_{DI}^{t+1} \mid G_{DI}^{t}).$$

The temporal change in the edge-existence probability can be expressed as follows (40).

$$p_{ij}^{t+1} = g_{\theta}\left(p_{ij}^{t}, \Delta x_i^t, \Delta x_j^t, r_{ij}\right),$$

where the feature variation of node v_i is defined as follows (41).

$$\Delta x_i^t = x_i^{t+1} - x_i^t.$$

Equation (41) allows the model to represent how the relational structure changes in response to behavioral changes, environmental changes, or attack events.

Graph Signal Representation

The node feature matrix X^t can be interpreted as a graph signal defined over the DI graph. Given the adjacency matrix A^t, the degree matrix D^t is defined as follows (42).

$$D_{ii}^{t} = \sum_{j} A_{ij}^{t}.$$

The graph Laplacian is then defined as follows (43).

$$L^{t} = D^{t} - A^{t}.$$

The smoothness of the graph signal is defined as follows (44).

$$\mathcal{S}(X^{t}) = \text{Tr}((X^{t})^{\top} L^{t} X^{t}).$$

Equivalently, the smoothness can be written as follows (45).

$$\mathcal{S}(X^{t}) = \frac{1}{2} \sum_{i,j} A_{ij}^{t} \left\| x_i^t - x_j^t \right\|_2^2.$$

In a normal DI state, connected nodes tend to have mutually consistent feature values; therefore, the smoothness value remains low. In contrast, abnormal states may introduce abrupt inconsistencies among connected components, increasing the graph-signal smoothness value.

User-Level Subgraph

The DI of a specific user u is defined as a user-level subgraph of the global DI graph (46).

$$G_{DI}^{t}(u) = (V_u^t, E_u^t, X_u^t) \subseteq G_{DI}^{t}.$$

The user-level node set is defined as follows (47).

$$V_u^t = \{h_u, d_u(t), l_u(t), s_u(t), b_u(t)\}.$$

The corresponding edge set is defined as follows (48).

$$E_u^t = \{(v_i, v_j, r_{ij}, p_{ij}^t) \mid v_i, v_j \in V_u^t\}.$$

Thus, the tuple-based representation provides a node-level definition of DI, whereas the proposed graph model extends it by incorporating relationships and temporal changes among identity components. The proposed graph-based probabilistic DI model interprets DI not as a static collection of attributes, but as a dynamic relational system. Nodes represent heterogeneous identity components, edges represent probabilistic relationships, and temporal evolution reflects changes in user behavior and environmental context.

Consequently, DI can be redefined as follows (49).

$$DI(u,t) \Rightarrow G_{DI}^t(u) \subseteq G_{DI}^t.$$

This model provides a mathematical foundation for the anomaly detection and attack prediction methods developed in the subsequent sections.

Anomaly Detection and Integrity Verification Model

Based on the mathematical model and graph-based representation of DI defined in Section sec.mathematical_modeling and Section sec.graph_based_di, this section sec.anomaly_detection proposes an integrated framework for anomaly detection and integrity verification. The objective of the proposed model is to detect deviations from normal DI patterns and to determine whether a DI state has been tampered with, misused, or generated under abnormal conditions.

Normal DI Modeling

The DI of user u at time t is defined as shown in (50), where h_u, $d_u(t)$, $l_u(t)$, $s_u(t)$, and $b_u(t)$ represent the human-related, device-related, location-related, service-related, and behavior-related components, respectively.

$$DI(u,t) = \big(h_u, d_u(t), l_u(t), s_u(t), b_u(t)\big).$$

The behavioral distribution under a normal state is defined in (51). This distribution models the expected behavioral pattern under a given contextual condition.

$$P_{\text{normal}}\big(DI(u,t)\big) = P\big(b_u(t) \mid h_u, d_u(t), l_u(t), s_u(t)\big).$$

In this formulation, an identity state is considered normal when the observed behavior is sufficiently consistent with the user, device, location, and service context. Conversely, a low probability indicates that the observed behavioral component is unlikely to occur under the given DI context.

Anomaly Detection Model

An anomaly is defined as a significant deviation from normal DI patterns. In this study, the abnormality of DI is not determined by a single criterion. Instead, it is quantified from three complementary perspectives: a probability-based model, a distance-based model, and a graph-based model. Each model detects a different aspect of abnormality. The probability-based model evaluates how likely the current DI state is under the normal distribution. The distance-based model measures the deviation from the user's historical normal pattern. The graph-based model evaluates abnormalities in the relational structure among identity components.

Probability-Based Anomaly Detection

Probability-based anomaly detection evaluates how likely the currently observed $DI(u,t)$ is to be generated from normal user patterns. A normal distribution P_{normal} is learned from DI data collected under normal conditions, either for an individual user or for a population of users. The occurrence probability of the current DI state is then calculated, as shown in (52).

$$A_p(u,t) = \begin{cases} 1, & \text{if } P_{\text{normal}}\big(DI(u,t)\big) < \theta, \\ 0, & \text{otherwise.} \end{cases}$$

Here, θ is the threshold for anomaly detection. If $P_{\text{normal}}\big(DI(u,t)\big)$ is smaller than θ, the corresponding DI state is regarded as unlikely to occur under the normal distribution and is therefore classified as anomalous. For example, if a user simultaneously exhibits an unfamiliar device, an unusual location, and an abnormal service-access pattern, the probability of the corresponding DI state may become low. Therefore, the probability-based model is suitable for determining whether the overall DI combination is normal.

Distance-Based Anomaly Detection

Distance-based anomaly detection measures how far the current DI state deviates from the historical normal pattern of user u. Since a DI state is originally defined as a heterogeneous tuple, it is first transformed into a numerical feature vector through the feature mapping function $\phi(\cdot)$. The resulting vector is denoted by (53).

$$z_u(t) = \phi\big(DI(u,t), G_{DI}^t(u)\big) \in \mathbb{R}^m.$$

Let μ_u denote the centroid of the user's historical normal DI feature vectors. The distance-based anomaly decision is then defined in (54).

$$A_d(u,t) = \begin{cases} 1, & \text{if } \|z_u(t) - \mu_u\|_2 > \epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

Here, ϵ is the distance-based anomaly threshold. If the current DI feature vector is farther than ϵ from the user's normal centroid μ_u, the state is regarded as deviating from the user's normal pattern. This model is suitable for personalized anomaly detection. For example, the same service-access behavior may be normal for one user but abnormal for another user. Therefore, the distance-based model detects personalized deviations by comparing the current DI state with the user's own behavioral history.

Graph-Based Anomaly Detection

Graph-based anomaly detection evaluates whether the relational structure among the components of DI is normal. When DI is represented as a graph $G_{DI}^t(u) = (V_u^t, E_u^t, X_u^t)$, nodes represent the user, device, location, service, and behavioral components, while edges represent the relationships among them. In this setting, anomalies may appear not only as changes in individual attributes, but also as changes in the relational structure.

The graph-based anomaly score is defined in (55).

$$A_g(u,t) = \sum_{(v_i,v_j)\in E_u^t} w_{ij} \cdot \delta(v_i, v_j, t).$$

Here, $\delta(v_i, v_j, t)$ is a deviation function that measures how much the relationship between nodes v_i and v_j deviates from the normal relationship, and w_{ij} denotes the importance weight of the corresponding relationship. Thus, when a large deviation occurs in a highly important relationship, the graph-based anomaly score $A_g(u,t)$ increases. For example, if a user accesses a sensitive service through a new device and an unfamiliar location, the change may appear to be a partial attribute variation when viewed individually. However, from a graph perspective, it can be interpreted as the creation of a new relationship or an abnormal access path. Therefore, the graph-based model is suitable for analyzing connectivity, relationship strength, relational changes, and abnormal paths among DI components.

Integrated Anomaly Decision

The probability-based, distance-based, and graph-based anomaly detection models measure different aspects of abnormality. Therefore, the final anomaly score can be defined by combining these three models, as shown in (56).

$$A(u,t) = \alpha A_p(u,t) + \beta A_d(u,t) + \gamma A_g(u,t).$$

Here, α, β, and γ are weights that control the relative importance of each anomaly detection model. The final anomaly decision is performed in (57).

$$A_{\text{final}}(u,t) = \begin{cases} 1, & \text{if } A(u,t) > \tau, \\ 0, & \text{otherwise.} \end{cases}$$

Here, τ is the final anomaly decision threshold. Since this integrated model simultaneously considers probabilistic rarity, distance from personalized normal patterns, and abnormalities in relational structures, it can detect complex DI anomaly scenarios more effectively than a single-criterion detection model.

Attack Modeling and Future Attack Prediction

Based on the DI-based anomaly detection results defined in the previous section, this section presents a mathematical model for attack behavior and a method for predicting future attack directions. Since DI is a structural object formed by the temporal combination of user, device, location, service, and behavioral attributes, an attacker does not necessarily manipulate only a single attribute. Instead, an attack may be performed by modifying the relationships among multiple DI components or by generating abnormal access paths within the DI graph.

Attack Surface Definition

In a DI-based environment, the attack surface is defined by the attribute domains constituting DI and the relationships among them. The DI of user u at time t is represented in $DI(u,t) = \big(h_u, d_u(t), l_u(t), s_u(t), b_u(t)\big)$, where h_u, $d_u(t)$, $l_u(t)$, $s_u(t)$, and $b_u(t)$ denote the human-related, device-related, location-related, service-related, and behavior-related components, respectively. Therefore, the DI attack surface $\mathcal{A}_s$ is defined in $\mathcal{A}_s = \{\mathcal{X}_H, \mathcal{X}_D, \mathcal{X}_L, \mathcal{X}_S, \mathcal{X}_B, \mathcal{R}_{DI}\}$, where $\mathcal{R}_{DI}$ represents the set of relationships among DI components. This means that an attacker can compromise the normal structure of DI not only by manipulating individual attributes, but also by modifying the relationships among those attributes.

Attack Vector Modeling

Let the DI graph at time t be denoted by $G_{DI}^t = (V^t, E^t, X^t)$. An attack vector can then be defined as a transformation function that converts a normal DI graph into a compromised DI graph, as shown in $\Phi_a: G_{DI}^t \to \tilde{G}_{DI}^{t+1}$, where Φ_a denotes an attack transformation and $\tilde{G}_{DI}^{t+1}$ represents a compromised DI graph at the next time step. The attack vector is categorized into three types, as defined in $\Phi_a = \{\phi_v, \phi_e, \phi_b\}$, where ϕ_v denotes node manipulation, ϕ_e denotes edge manipulation, and ϕ_b denotes behavioral-trajectory manipulation. First, node manipulation refers to attacks that falsify the values of DI components, such as devices, locations, or services, or insert new malicious nodes. Second, edge manipulation refers to attacks that create abnormal relationships between DI components that should not normally be connected. Third, behavioral-trajectory manipulation refers to attacks that modify DI movement paths or access patterns over time so that malicious behavior appears similar to normal user behavior.

Attack State Transition Model

An attack is not an isolated event occurring at a single time point. Rather, it can be interpreted as a sequential state-transition process that includes reconnaissance, credential compromise, lateral movement, service access, and data exfiltration or service abuse. For this purpose, the attack-state set is defined in $\mathcal{S}_a = \{s_0, s_1, s_2, s_3, s_4\}$, where s_0 denotes the normal state, s_1 denotes the reconnaissance state, s_2 denotes the credential-compromise state, s_3 denotes the lateral-movement state, and s_4 denotes the data-exfiltration or service-abuse state. The attack-state transition is determined by the anomaly score and structural changes observed in the DI graph. To avoid notational ambiguity with the weighting parameter α used in Section sec.anomaly_detection, let a_t denote the DI anomaly score at time t, i.e., $a_t = A(u,t)$. The attack-state transition probability is then defined in $P(s_{t+1} \mid s_t, G_{DI}^t) = f_\theta(a_t, \Delta E_t, \Delta V_t, \Delta B_t)$ where ΔE_t denotes edge variation, ΔV_t denotes node variation, and ΔB_t denotes behavioral variation at time t. Therefore, as the structural change in the DI graph becomes larger at a given time, the probability of transition to the next attack stage increases.

Correlation-Based Future Attack Prediction

To predict future attack directions, this study analyzes the correlation between DI anomaly scores and network signals interacting with DI. The set of network signals is defined in $\mathcal{N}(t) = \{n_1(t), n_2(t), \dots, n_k(t)\}$, where $n_i(t)$ denotes the i-th network signal observed at time t, such as access frequency, authentication failure count, packet volume, service-request frequency, or abnormal session duration. The correlation between the DI anomaly score and each network signal is computed in $\rho_i = \frac{\mathrm{Cov}(a_t, n_i(t))}{\sigma_a \sigma_{n_i}}$ where ρ_i denotes the correlation coefficient between the anomaly score a_t and the i-th network signal. Network signals with high correlation are

regarded as attack indicators closely related to DI anomalies. Therefore, the set of candidate attack indicators is defined in $\mathcal{P}_{\text{attack}} = \{n_i \mid |\rho_i| \geq \theta_\rho\}$ where θ_ρ is a predefined correlation threshold.

Future Attack Direction Estimation

Attack-direction estimation is defined as the problem of estimating the service, device, domain, or DI component that is most likely to be accessed next from the current abnormal state of the DI graph. The set of possible next attack-target candidates at time $t+1$ is defined in $\mathcal{T}_{t+1} = \{v_j \in V^{t+1} \mid P(v_j \mid G_{DI}^t, s_t) \geq \theta_p\}$ where v_j denotes a candidate target node and θ_p denotes a prediction threshold. The attack-likelihood score of each candidate node is calculated in $\text{Score}(v_j) = \lambda_1 a_t + \lambda_2 C(v_j) + \lambda_3 R(v_j) + \lambda_4 H(v_j)$, where $C(v_j)$ denotes the correlation strength with abnormal network signals, $R(v_j)$ denotes the risk level of the target node, $H(v_j)$ denotes historical attack relevance, and λ_1, λ_2, λ_3, and λ_4 are weighting parameters. Finally, the node with the highest score is predicted as the most likely future attack target in $v_{t+1}^* = \arg\max_{v_j \in \mathcal{T}_{t+1}} \text{Score}(v_j)$.

Interpretation of Predicted Attack Paths

The predicted attack direction is not interpreted merely as a single target node. Instead, it is interpreted as a path in the DI graph along which an attacker is likely to move. The predicted attack path is defined in $\text{Path}_a = (v_0, v_1, \ldots, v_m)$, where v_0 denotes the initially compromised DI component and v_m denotes the final target node. This path indicates how an attacker may move from one DI component to a service or data resource. For example, if abnormal location information and a new device node are observed simultaneously, and high-risk service-access requests subsequently increase, the corresponding path can be interpreted as an account-takeover scenario followed by privilege escalation or unauthorized data access. In this section, a DI-graph-based attack modeling and future attack prediction method was presented. An attacker compromises the normal identity structure by manipulating the nodes, relationships, and behavioral trajectories of DI components. The proposed model integrates anomaly scores, graph-structural changes, and correlations among network signals to estimate attack-state transitions and predict target nodes and attack paths with high future attack likelihood. Therefore, the proposed approach extends DI-based security analysis beyond simple anomaly detection and supports proactive attack response and risk mitigation in DI-based environments.

Integration with Current Identity-Management Ecosystems

The proposed framework can be integrated with current identity-management ecosystems as an additional continuous verification layer rather than as a replacement for existing authentication protocols. In conventional identity-management systems, such as SSO, OAuth/OIDC-based authentication, enterprise IAM, and access-control platforms, the authentication result mainly confirms whether a credential, token, or session is valid at a specific time. The proposed DI graph model can use these authentication events as input signals and then evaluate whether the relationships among user, device, location, service, session, and behavior components remain consistent after authentication. Therefore, the proposed framework complements existing identity-management systems by adding post-authentication relational integrity verification. In a Decentralized Identity (DID) or Verifiable Credential (VC) ecosystem, DID subjects, credentials, issuer-verifier relationships, wallet devices, service providers, and credential-presentation sessions can be represented as DI graph nodes and edges. For example, a credential presentation from an unfamiliar wallet device, an unusual verifier service, or an abnormal location can be modeled as a low-weight or newly created relationship in the DI graph. This allows the proposed framework to analyze not only whether a credential is cryptographically valid, but also whether the surrounding identity context is structurally consistent with the user's historical DI profile. The proposed framework is also compatible with Zero Trust architectures. In Zero Trust, access decisions are not based on one-time authentication alone, but on continuous evaluation of identity, device, context, and behavior. The DI anomaly score and integrity violation score defined in this study can be used as additional risk signals for a policy decision point. For example, when the DI graph indicates an abnormal user–device edge, an unlikely location transition, or an abnormal session–behavior relationship, the system may trigger step-up authentication, restrict access to sensitive services, terminate the session, or generate an alert for security monitoring. Thus, the proposed model can practically support identity-centric Zero Trust enforcement by providing graph-based, interpretable, and continuously updated risk evidence.

Use Cases and Experimental Evaluation

This study evaluates the proposed DI-based graph anomaly detection and future attack prediction framework by combining public cybersecurity datasets with synthetic attack scenarios. The experiments are designed from three perspectives: (1) anomaly detection performance based on DI graph representation, (2) integrity verification performance of DI structures, and (3) feasibility of future attack prediction based on attack-state transitions.

Experimental Dataset Composition

The experimental dataset was constructed by combining real-world network security datasets with DI-based synthetic scenarios. The real-world datasets reflect abnormal behavior at the network traffic level, whereas the synthetic scenarios represent complex attack patterns from the perspective of DI.

CICIDS2017 Dataset

CICIDS2017 is a representative network intrusion detection dataset widely used in cybersecurity research. It contains benign traffic as well as various types of attack traffic. In this study, attack scenarios that are directly relevant to DI-based

security analysis were selected. The attack types used in the experiment are as follows.

- Distributed Denial of Service (DDoS),
- PortScan,
- brute-force attack.

Each flow record includes attributes such as source IP address, destination IP address, protocol, timestamp, packet-level statistics, and flow duration. In this study, these flow-level records were reconstructed as DI events.

IoT-23 Dataset

The IoT-23 dataset contains benign and malicious network behavior observed in IoT environments. In this study, scenarios related to botnet infection, abnormal external access, and automated attack behavior were selected. Since device-based identity and behavior-based profiling are particularly important in IoT environments, this dataset was used to evaluate device identity anomalies.

DI-Based Synthetic Attack Scenarios

Public datasets alone are insufficient to fully represent high-level DI attacks such as DI theft, session hijacking, impossible travel, and device spoofing. Therefore, this study additionally constructed synthetic DI-based attack scenarios. Normal scenarios were generated to reflect the following characteristics.

- normal authentication based on the same user,
- access from familiar devices,
- normal geographic transitions,
- regular access-time patterns,
- normal service-access behavior.

Attack scenarios were constructed as follows.

- credential theft,
- device spoofing,
- impossible travel,
- session hijacking,
- bot-driven abnormal access,
- lateral movement simulation.

It should be noted that the synthetic DI-based scenarios were not intended to replace real-world DI datasets. Rather, they were used to complement public network-security datasets that do not explicitly contain high-level identity semantics such as credential theft, impossible travel, session hijacking, and device spoofing. In this study, CICIDS2017 and IoT-23 provide real-world network-level attack patterns, while the synthetic DI scenarios provide identity-level relational structures that are not directly available in existing public datasets. Therefore, the evaluation should be interpreted as a controlled feasibility analysis of the proposed DI representation, rather than as a complete validation of generalization performance in all real-world DI environments.

Use Cases Definition for DI-Based Security Analysis

The synthetic DI attack scenarios are composed of six representative use cases to clearly demonstrate the practical applicability of the proposed graph-based DI framework. Each use case is defined based on the manipulated DI components, the resulting changes in graph structure, and the expected anomaly signals. Unlike existing authentication-centric analyses, the use cases proposed in this study focus on analyzing whether the relationships among user, device, location, service, session, and behavior components remain consistent even after authentication.

Use Case 1: Credential Theft

This use case assumes a situation in which an attacker obtains valid user credentials and attempts to access a service as if they were a legitimate user. In this case, since correct account information is used, the user identity component may appear normal. However, other DI components such as device, location, session, and behavior may differ from the user's past DI profile. In the proposed DI graph, this situation is represented as abnormal relationships among the user, session, service, and behavior nodes. Therefore, this anomalous behavior is detected not through authentication failure, but through relational inconsistencies that occur after successful authentication.

Use Case 2: Device Spoofing

This use case assumes a situation in which an attacker attempts to access a service using a forged or unfamiliar device identifier. In this case, the user identity may remain unchanged, but the device-related components differ from the past DI profile. In the proposed DI graph, this situation is represented as a newly created edge or a low-weight edge between the user node and an unfamiliar device node, so this anomalous behavior is detected not as a simple attribute change, but as a device identity mismatch.

Use Case 3: Impossible Travel

This use case represents a situation in which the same user account accesses a service from geographically distant locations within a time interval that is realistically impossible. Each individual login event may appear normal on its own, but the temporal transition between location components does not match a normal DI movement path. In the proposed DI graph, this situation is represented as an abnormal transition between location nodes and a violation of temporal consistency, so this anomalous behavior is detected by jointly analyzing location changes, access times, and user-device relationships.

Use Case 4: Session Hijacking

This use case assumes a situation in which an attacker hijacks an authenticated session and performs abnormal service access or behavior using a valid session token. In this case, the authentication information may remain valid, but the relationships among the session, service, and behavior components no longer match the user's normal pattern. In the proposed DI graph, this situation is represented as an anomalous edge between the session node and an abnormal service or behavior node, so the proposed framework can detect session misuse even when the session itself is not immediately invalidated.

Use Case 5: Bot-Driven Abnormal Access

This use case assumes a situation in which an automated bot repeatedly accesses a service using abnormal request frequency, access intervals, or behavior patterns. While user, device, or service information is not always forged, the behavior component differs significantly from normal human activity. In the proposed DI graph, this situation is represented by abnormal behavior pattern nodes, excessive service interaction frequency, and irregular temporal transitions, so this anomalous behavior is detected by analyzing behavior-based graph characteristics and temporal access consistency.

Use Case 6: Lateral Movement

This use case assumes a situation in which an attacker first compromises one DI component and then attempts to move to another device, service, or resource. In this case, the attack is represented not as a single anomalous event, but as a sequence of graph structure changes. In the proposed DI graph, lateral movement appears as new paths among user, device, service, and behavior nodes that were not observed in the normal DI profile. Therefore, the proposed framework can support future attack prediction by estimating abnormal paths and candidate target nodes from the currently compromised DI state.

DI Graph Construction and Feature Extraction

In this study, DI is defined as the structural tuple shown in (58).

$$DI(u,t) = \left(h_u, d_u(t), l_u(t), s_u(t), b_u(t)\right).$$

Here, each component has the following meaning.

- h_u. user identity information,
- $d_u(t)$. device attributes,
- $l_u(t)$. location information,
- $s_u(t)$. service-access information,
- $b_u(t)$. behavioral-pattern information.

Each DI event was mapped to graph nodes, and edges were constructed based on temporal continuity, attribute similarity, and service relevance. Figure 2 illustrates the process by which DI events are transformed into a graph structure.

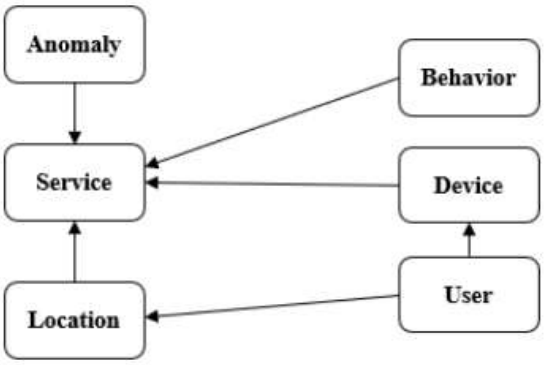

Figure 2. DI graph construction from DI events

The following graph-based features were used.

- node degree,
- clustering coefficient,
- betweenness centrality,
- temporal transition consistency,
- behavioral similarity score,
- service interaction frequency.

These features are more suitable for detecting structural anomalies in DI than simple traffic-based anomaly detection features.

Implementation Details for Reproducibility

Although Sections 2–5 present the proposed framework in a general mathematical form, the experimental implementation used concrete empirical estimators for the abstract functions. Each DI event was first converted into a fixed-length feature vector by the mapping function $\phi(\cdot)$. Categorical attributes, such as user ID, device identifier, session identifier, service type, and IP/subnet information, were encoded as categorical or hashed features. Numerical attributes, such as access frequency, session duration, request interval, and behavior-based scores, were normalized using statistics computed only from the training set. The edge-existence probability p_{ij}^t in the DI graph was estimated from the frequency of observed relationships in the training data. Specifically, for a pair of DI components (v_i, v_j), the empirical edge probability was computed as follows,

$$\hat{p}_{ij} = \frac{N(v_i, v_j) + \eta}{N(v_i) + \eta|V_j|},$$

where $N(v_i, v_j)$ is the number of times the relationship between v_i and v_j was observed in the training data, $N(v_i)$ is the number of observations involving v_i, $|V_j|$ is the number of possible target nodes of the corresponding type, and η is a Laplace smoothing parameter. In the experiments, this probability was used as the edge weight of the DI graph. The transition model T_θ was implemented as an empirical first-order transition model over discretized DI states. Each DI state was represented by a state signature consisting of user, device, location/subnet, service, session, and behavior-level features. The transition probability from the previous state s_k to the next state s_{k+1} was estimated as follows,

$$\hat{T}(s_{k+1}|s_k) = \frac{N(s_k \to s_{k+1}) + \eta}{\sum_{s'} N(s_k \to s') + \eta|\mathcal{S}|},$$

where $N(s_k \to s_{k+1})$ denotes the number of observed transitions from s_k to s_{k+1} in the training data, $\mathcal{S}$ is the set of observed DI state signatures, and η is the Laplace smoothing

parameter. The temporal transition inconsistency feature was then calculated as follows,

$$A_{\text{temp}}(u, t+1) = -\log \hat{T}(s_{t+1}|s_t).$$

Thus, an unseen or rarely observed transition receives a larger temporal anomaly score. The deviation function $\delta(v_i, v_j, t)$ used in the graph-based anomaly score was implemented as the difference between the current relationship and the historical relationship profile. For an observed edge, the deviation was calculated as follows,

$$\delta(v_i, v_j, t) = 1 - \hat{p}_{ij}.$$

Therefore, relationships that were frequently observed in the normal training data produced small deviation values, whereas newly created or rarely observed relationships produced large deviation values. The final feature vector used by the classifier consisted of graphstructural features and DI-behavioral features, including node degree, clustering coefficient, betweenness centrality, edge probability, temporal transition inconsistency, behavioral similarity score, service interaction frequency, and sessionduration-related features. The proposed model was implemented using a Random Forest classifier trained on the extracted DI graph features. Feature normalization, empirical probability estimation, and threshold selection were performed using only the training and validation data to prevent information leakage from the test set. The computational complexity of the proposed framework is mainly determined by DI graph construction, graph-feature extraction, and classification. For each incoming DI event, attribute encoding and feature mapping require approximately $O(d)$ time, where d is the feature dimension. Since each DI event updates only a bounded number of typed relationships among user, device, location, service, session, and behavior components, the graph update cost is approximately $O(r)$ per event, where r is the number of relationship types. Empirical edge-probability and transition-probability updates are implemented using count tables, so their average update cost is $O(1)$ per observed edge or transition. Local graph features such as node degree, edge weight, service-interaction frequency, and temporal consistency can be updated incrementally with $O(1)$ to $O(\deg(v))$ cost. However, global features such as exact betweenness centrality may require $O(nm)$ time for a graph with n nodes and m edges if recomputed over the entire graph. Therefore, enterprise-scale or IoT-scale deployment would require sparse graph storage, sliding-window graph maintenance, approximate centrality computation, and incremental feature updates. The present study provides a theoretical complexity discussion, but does not conduct a large-scale stress test with millions of nodes and edges.

Baseline Models and Evaluation Metrics

To evaluate the performance of the proposed model, representative unsupervised anomaly detection methods were used as baselines.

- Isolation Forest,
- One-Class Support Vector Machine (OC-SVM),
- Autoencoder.

The proposed model was implemented as a Random Forest-based anomaly detection framework that combines DI graph-based structural features with behavior-based features.

Performance was evaluated using the metrics shown in (63)–(66).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

$$\text{Precision} = \frac{TP}{TP + FP}.$$

$$\text{Recall} = \frac{TP}{TP + FN}.$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

In addition, the following metrics were used.

- ROC-AUC,
- False Positive Rate (FPR),
- detection latency.

Although recent graph neural network-based models, such as GCN, GraphSAGE, GAT, and temporal graph learning models, are important state-of-the-art approaches for graph-based anomaly detection, they were not included as direct baselines in this proof-of-concept experiment. The datasets used in this study were originally provided as flow-level or event-level cybersecurity datasets rather than as native temporal DI graphs with explicit *user-device-session-service-behavior* relationships. Therefore, applying GNN-based baselines would require an additional graph construction protocol, node/edge labeling strategy, and temporal graph split, which could introduce another source of experimental bias. For this reason, the present evaluation focuses on representative feature-level anomaly detection baselines under the same input setting. A full comparison with GCN, GraphSAGE, GAT, and temporal graph learning models will be conducted in future work using real-world DI graph datasets.

Experimental Results and Performance Analysis

To avoid overly optimistic evaluation, the dataset was split at the user/session-scenario level rather than at the individual event level. Events generated from the same user profile or the same synthetic attack scenario were not simultaneously included in both training and test sets. Feature normalization and threshold selection were performed using only the training and validation data. This setting was used to reduce the possibility of data leakage between training and test samples.

Table 2 presents the performance comparison of the anomaly detection models.

Because some DI-based attack scenarios were synthetically generated, the results in Table 2 should be interpreted carefully. Under the controlled DI-based synthetic scenario setting, the proposed framework achieved high performance. However, the additional validation results indicate that the current synthetic DI dataset contains highly separable feature patterns. In particular, several baseline models also achieved near-perfect performance in repeated runs, and the

performance difference between the proposed framework and the strongest baseline was not statistically significant under the current setting. Therefore, the reported results should not be interpreted as conclusive evidence that the proposed framework substantially outperforms all baseline methods in real-world environments. Rather, they should be interpreted as controlled feasibility evidence showing that the proposed DI graph representation can separate predefined normal and attack-related identity structures in the constructed experimental setting.

Statistical Validation and Ablation Analysis

To examine whether the performance gain of the proposed framework was caused by dataset construction bias or a particular random split, additional validation was conducted over five independent runs using different random seeds. In each run, the dataset was split at the user/session-scenario level, and the same training, validation, and test partitions were used for all compared models. Feature normalization, empirical probability estimation, and threshold selection were performed using only the training and validation data. For each metric, the mean value and 95% confidence interval were calculated across repeated runs. The confidence interval was calculated as follows:

$$CI_{95\%} = \bar{x} \pm 1.96 \frac{s}{\sqrt{n}},$$

where $\bar{x}$ denotes the mean performance over repeated runs, s denotes the standard deviation, and n denotes the number of runs.

As shown in Table 3, the proposed framework achieved stable performance across five independent runs. However, the baseline models also achieved very high performance, indicating that the current synthetic DI dataset contains strongly separable feature patterns. Therefore, the performance gap should be interpreted carefully. The result supports the feasibility of the proposed DI graph representation under controlled synthetic conditions, but it does not by itself prove general superiority in real-world DI environments. A paired statistical test was additionally conducted between the proposed framework and the strongest baseline model. In this experiment, the Autoencoder model was selected as the strongest baseline because it showed the highest overall performance among the baseline methods in the repeated-run evaluation.

Statistical significance test between the proposed framework and the strongest baseline.

Metric	Compared Models	Test Method	p-value
Accuracy	Proposed vs. Autoencoder	Paired t-test	0.3739
Precision	Proposed vs. Autoencoder	Paired t-test	0.3739
Recall	Proposed vs. Autoencoder	Paired t-test	N/A
F1-score	Proposed vs. Autoencoder	Paired t-test	0.3739
ROC-AUC	Proposed vs. Autoencoder	Paired t-test	N/A

The paired t-test results in Table 4 show that the difference between the proposed framework and the strongest baseline was not statistically significant under the current synthetic DI setting. For Recall and ROC-AUC, statistical testing was not applicable because both compared models produced identical values across repeated runs. This result suggests a ceiling effect caused by the highly separable structure of the synthetic DI dataset. Accordingly, the proposed framework should not be claimed to statistically outperform the strongest baseline in this experimental setting. To further analyze the contribution of each feature group, an ablation analysis was conducted over five independent runs. The full model was compared with reduced variants in which one major feature group was removed at a time, including relational graph features, temporal transition features, behavioral features, and service-interaction features. The ablation analysis showed that removing a single feature group did not reduce the performance in the current synthetic DI dataset. This result should not be interpreted as evidence that these feature groups are unnecessary. Rather, it indicates that the constructed synthetic DI scenarios contain multiple highly separable signals, so that the remaining feature groups can still distinguish normal and attack cases even after one component is removed. Therefore, the ablation analysis further supports the need for future validation using less separable and real-world DI datasets. Additional diagnostic analysis confirmed that several individual features were already highly discriminative in the constructed synthetic dataset. This observation further supports the interpretation that the reported performance reflects controlled feasibility rather than real-world generalization.

Use-Case-Based Experimental Analysis

This section analyzes the experimental results from the perspective of the six DI-based security use cases defined in Section 6.2. Rather than simply reporting overall classification performance metrics, the discussion focuses on how each attack scenario alters the structure of the DI graph, and what anomaly signals the proposed framework captures from these changes. This use-case-based analysis is intended to demonstrate that the proposed model can detect not only network-level anomalous behavior, but also inconsistencies in identity structure that arise after authentication.

Analysis of Credential Theft

In the credential theft scenario, since the attacker uses valid account information, the access may appear legitimate when viewed solely in terms of authentication outcome. However, the proposed framework detects this anomalous behavior by comparing the current DI graph with the user's past DI

profile. Here, the key anomaly signals emerge in the relationships among the user, session, service, and behavior nodes. That is, even with a valid user account, if it appears alongside an unfamiliar session pattern, abnormal service access, or an unusual behavioral pattern, the graph-based anomaly score naturally rises. These results demonstrate that the proposed model can effectively detect identity misuse occurring after the authentication stage—something that is difficult to capture through credential verification alone.

Analysis of Device Spoofing

In the device spoofing scenario, the user node itself remains stable, while the connected device node changes to an unfamiliar or low-trust device identifier. This change manifests in the DI graph as either the creation of a new user-device edge or a weakening of the weight of an existing edge. Experimental analysis confirmed that device-related graph features, such as user-device edge consistency and changes in device identity, substantively contribute to detecting this scenario. This demonstrates that, by analyzing the relational structure among DI components, the proposed model can effectively distinguish between device changes within a normal range and abnormal manipulation of device identity.

Analysis of Impossible Travel

In the impossible travel scenario, the location component changes abruptly within a time interval that is realistically impossible. Each individual access event, considered in isolation, may appear valid, but when examining the transition probability between consecutive DI states, the likelihood proves to be extremely low. The proposed framework detects this case through the resulting inconsistency in temporal transitions and abnormal location-related edges. These results show that the proposed model is effective at identifying anomalous behavior arising from the continuous evolution of DI states, rather than from a single isolated attribute value.

Analysis of Session Hijacking

In the session hijacking scenario, since a valid session token can be reused as-is by an attacker, the session component alone may not clearly reveal anomalous behavior in the early stages of the attack. Over time, however, abnormal relationships gradually begin to form between the session node and the service node, or between the session node and the behavior node. The proposed framework detects this scenario by comprehensively analyzing session-service consistency, session-behavior consistency, and the deviation from the user's typical service access patterns. These results clearly demonstrate that the proposed DI graph model can continuously verify the integrity of identity relationships even after authentication has been completed.

Analysis of Bot-Driven Abnormal Access

In the bot-based anomalous access scenario, the most prominent anomaly signals are observed above all in the behavior and temporal components. Repeated requests, abnormal access intervals, and excessive frequency of service interaction generate graph patterns that are clearly distinguishable from normal human behavior. In the proposed framework, these changes are reflected through behavior-based features and temporal consistency features, gradually raising the anomaly score. These results suggest that even when device or session information appears normal on the surface, the proposed model can sufficiently detect automated anomalous access.

Analysis of Lateral Movement

In the lateral movement scenario, the attack unfolds not as a single momentary event, but as a continuous sequence of abnormal relationships among DI components. The attacker first compromises a single node, such as a device or session, and then attempts to leverage this foothold to expand access to additional services or resources. In the DI graph, this movement appears as a new path leading from the compromised component to a high-risk service node, or as an abnormal edge transition. The proposed framework analyzes these changes in graph structure to estimate potential future attack directions. These results support the notion that the proposed model can be extended beyond simple anomaly detection to proactive attack path prediction.

Summary of Use-Case-Based Analysis

Overall, the use-case-based analysis shows that the proposed graph-based DI framework can effectively detect diverse types of identity-related attacks by observing structural changes in DI components. Credential theft and session hijacking are detected through abnormal session-service-behavior relationships, device spoofing through user-device inconsistency, impossible travel through location transition inconsistency, bot-based anomalous access through behavioral and temporal patterns, and lateral movement through abnormal graph path transitions. These results demonstrate that the proposed model, by identifying which DI relationships have been violated and how the anomalous identity state evolves over time, provides more interpretable anomaly detection than existing feature-based approaches.

Explainability Analysis Based on Graph Relationships

To clarify the interpretability of the proposed graph-based DI framework, this section summarizes how graph relationships provide explanatory evidence for representative attack scenarios. Unlike conventional feature-based anomaly detection methods that mainly report whether an event is abnormal, the proposed framework identifies which DI relationships are violated and how the abnormal state is formed.

Case-based explainability analysis of graph relationships in DI attack scenarios.

Attack case	Explainability through graph relationships
Credential theft	A valid login may appear normal in traditional feature-level analysis. However, the DI graph reveals abnormal user–session and session–behavior relationships after successful

Attack case	Explainability through graph relationships
	authentication. This explains that the anomaly is caused by post-authentication identity misuse rather than by login failure.
Device spoofing	Traditional analysis may only detect a new or changed device identifier. In contrast, the DI graph shows that a new or low-weight user–device edge is created between the user node and an unfamiliar device node. This indicates device identity mismatch rather than general traffic variation.
Impossible travel	A single access event may appear valid when considered independently. However, the DI graph reveals a temporally inconsistent transition between distant location nodes for the same user identity. This explains that the anomaly is caused by an impossible movement path.
Session hijacking	A valid session token may not immediately appear abnormal. However, the DI graph shows abnormal session–service or session–behavior edges that differ from the user's historical profile. This explains how a valid session is misused after authentication.
Lateral movement	Traditional feature-level analysis may observe sequential access to multiple services. The DI graph explains this behavior as a newly formed path from a compromised DI component to a high-risk service node, revealing the likely movement direction from the compromised component to the target service.

As shown in Table 5, the proposed framework provides interpretability by mapping anomalous events to violated DI relationships. For example, in credential theft, the anomaly is not explained simply as a high anomaly score, but as an abnormal combination of a legitimate user node with unfamiliar session and behavior nodes. In device spoofing, the explanation is obtained from the creation of a new or low-weight user–device edge. In impossible travel, the explanation comes from an unlikely transition between location nodes within a short time interval. Therefore, the proposed graph representation improves interpretability by showing the relational cause of an anomaly, whereas traditional feature-based methods mainly provide event-level anomaly scores without explicitly identifying the violated identity relationship.

Scope of Attack Prediction Evaluation

Although Section 5 formulates attack-state transition, future attack-direction estimation, and predicted attack paths, the present experimental evaluation does not provide a full quantitative validation of attack prediction performance. In this study, the attack prediction component is evaluated only at the feasibility and interpretation level through use-case-based analysis, especially in scenarios such as session hijacking, impossible travel, and lateral movement. Therefore, the reported experimental results should not be interpreted as quantitative evidence of attack-path prediction accuracy or forecasting performance. A quantitative evaluation of attack prediction would require ground-truth labels for future attack targets, attack paths, and prediction horizons. For example, the evaluation would need to define whether the predicted next target node $\hat{v}_{t+h}$ matches the actual future target node v_{t+h} at a given forecasting horizon h, and whether the predicted path $\widehat{Path}_a$ correctly overlaps with the actual attack path $Path_a$. However, the public datasets used in this study, including CICIDS2017 and IoT-23, are primarily flow-level or event-level cybersecurity datasets and do not provide complete DI-level ground-truth attack-path sequences. Likewise, the synthetic DI scenarios used in this study were designed mainly to evaluate structural anomaly detection and integrity violation patterns, rather than to provide fully labeled multi-step attack-path prediction benchmarks. Accordingly, this study does not claim that the proposed framework has been fully validated as a quantitative attack prediction system. Rather, the contribution of the attack prediction component is limited to the mathematical formulation of attack-state transition, candidate target estimation, and interpretable attack-path representation. Future work will construct DI datasets with explicit temporal attack-path labels and evaluate prediction performance using metrics such as top-k target prediction accuracy, attack-path success rate, path-level precision and recall, mean reciprocal rank, and horizon-wise forecasting accuracy.

External Validity and Generalization Limitations

The generalization of the proposed framework to real-world DI environments remains an important limitation of this study. At present, publicly available cybersecurity datasets rarely contain complete DI-level information, including user identity, device identity, session context, service-access history, location transition, and post-authentication behavior in a unified form. Therefore, this study combined public network-security datasets with DI-based synthetic scenarios to evaluate the feasibility of the proposed graph-based DI representation. The use of synthetic DI scenarios has both advantages and limitations. The advantage is that it enables controlled modeling of identity-related attacks, such as credential theft, device spoofing, impossible travel, session hijacking, bot-driven abnormal access, and lateral movement. These scenarios are difficult to evaluate using existing public traffic datasets alone because such datasets mainly provide network-flow-level attributes and attack labels. However, the limitation is that synthetic scenarios may not fully capture the diversity, noise, concept drift, and policy-dependent characteristics of real operational DI environments. Accordingly, the experimental results reported in this study

should be interpreted as evidence of feasibility under controlled conditions, rather than as conclusive evidence of real-world generalization. In practical deployment, the proposed framework would require calibration using organization-specific DI logs, including normal user behavior, device usage history, session patterns, access-control policies, and service-interaction records. Future work will therefore focus on validating the proposed model using real-world DI logs collected from enterprise, IoT, and medical information systems, and on evaluating robustness under cross-domain, cross-user, and temporally shifted conditions.

Integrity Verification Analysis

The objective of the DI integrity verification experiment is not merely to classify network traffic as benign or abnormal, but to verify whether the DI structure composed of user, device, location, service, and behavioral information remains consistent over time. In this study, a DI integrity violation is defined as a state in which a specific DI component is independently manipulated or the connection relationship among components becomes inconsistent with previously observed normal patterns. The experiment assumes situations in which an attacker manipulates some DI components. Specifically, the following four types of integrity violations were constructed.

- **Device identity manipulation**: the same user account accesses the system through an unfamiliar device identifier or an abnormal device fingerprint;
- **Service attribute tampering**: the access relationship suddenly changes to a sensitive or high-privilege service that is unrelated to the user's historical service-access pattern;
- **Temporal inconsistency injection**: authentication, session creation, or service-access events are inserted in an order that is inconsistent with the normal temporal sequence;
- **Identity linkage corruption**: normal connections among user, device, service, and behavior components are broken, or new abnormal connections are created by an attacker.

Figure 3 illustrates how these DI integrity violations appear in the graph structure. In a normal DI graph, a specific user node forms stable connections with a limited set of device nodes, consistent service nodes, and repeated behavioral-pattern nodes. In contrast, when an integrity violation occurs, the connectivity structure of the existing DI graph changes sharply. For example, in device identity manipulation, the user node remains the same, but the connected device node deviates from the existing set of normal devices. In service attribute tampering, a new edge is generated toward a service node that the user does not normally access. Temporal inconsistency injection creates transition edges that are impossible or highly unlikely in the normal temporal sequence. Identity linkage corruption disconnects existing user–device–service paths or creates abnormal bypass paths.

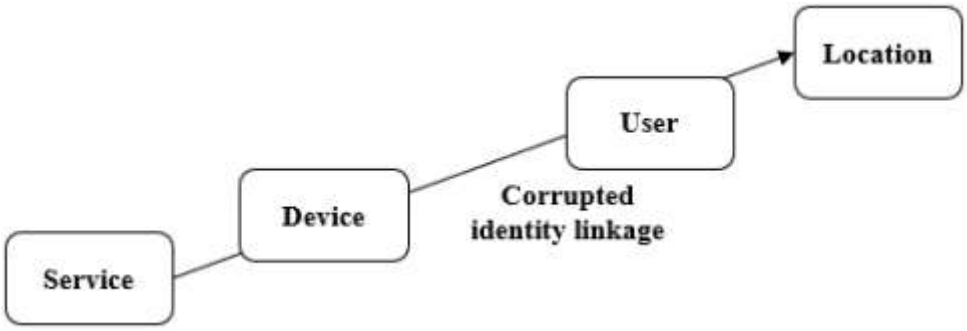

Figure 3. Integrity violation detection in the DI graph. Solid edges indicate normal and historically consistent relationships among user, device, location, service, and behavior nodes.

The detection process in Figure 3 can be interpreted in three steps. First, each DI event is decomposed into user, device, location, service, and behavioral attributes and then transformed into graph nodes and edges. Second, the current DI connectivity structure is compared with the historical normal DI profile. At this stage, the model compares not only changes in individual attribute values, but also graph-based features such as the connection strength between user and device, service-access frequency, temporal transition consistency, and behavioral similarity. Third, the degree to which the current DI graph deviates from the normal graph distribution is calculated as an integrity violation score. In this study, the DI integrity violation score $IV(u,t)$ is defined as shown in (68). In this figure, a corrupted identity linkage does not simply mean a network disconnection. It denotes a relational integrity violation in which a normal DI edge is missing, invalid, or replaced by an abnormal connection.

$$IV(u,t) = \alpha \Delta D(u,t) + \beta \Delta S(u,t) + \gamma \Delta T(u,t) + \delta \Delta L(u,t) + \eta \Delta B(u,t).$$

Here, $\Delta D(u,t)$ denotes the degree of device identity change, $\Delta S(u,t)$ denotes the degree of service-access attribute change, $\Delta T(u,t)$ denotes the degree of temporal transition inconsistency, $\Delta L(u,t)$ denotes the degree of linkage corruption among user, device, and service components, and $\Delta B(u,t)$ denotes the degree of behavioral-pattern change. The parameters α, β, γ, δ, and η are weights that control the relative importance of each integrity factor.

The integrity violation decision is made according to (69).

$$I_{\text{viol}}(u,t) = \begin{cases} 1, & \text{if } IV(u,t) > \tau_I, \\ 0, & \text{otherwise.} \end{cases}$$

Here, τ_I is the threshold for integrity violation detection. If a specific DI event shows only a weak change in a single attribute, it may be regarded as a normal variation. However, if device change, abnormal service access, temporal inconsistency, and behavioral change occur simultaneously, $IV(u,t)$ increases and the event is detected as an integrity violation. The result illustrated in Figure 3 shows that the proposed model detects integrity violations based on the relational consistency among DI components rather than on simple attribute-level changes. In particular, device identity manipulation appeared as the creation of an abnormal edge connected to a new device node, and service attribute tampering was observed as a connection to a node outside the existing service-access subgraph. Temporal inconsistency injection sharply reduced the edge weight based on the normal temporal order, while identity linkage corruption decreased the structural cohesion of the user-centered ego graph. Therefore, even events that may appear normal at the individual log level can be detected as complex integrity violations by jointly analyzing the connectivity structure and temporal transition patterns of the DI graph. These results imply that integrity verification in DI-based security models should be continuously performed even after successful authentication. Conventional authentication-oriented models tend to regard a user as legitimate once valid account credentials are provided. In contrast, the proposed approach continuously verifies whether the relationships among device, location, service, and behavior remain consistent with the existing DI profile after authentication. Therefore, it provides higher explainability for attacks that are difficult to detect using authentication information alone, such as account takeover, session hijacking, and device spoofing.

CONCLUSION

The primary achievements of this study are summarized as follows. First, this study formalized DI as a dynamic structural state composed of human, device, location, service, and behavioral components, rather than as a static authentication credential. Second, it proposed a graph-based DI representation that explicitly models relationships among heterogeneous identity components. Third, it defined anomaly detection and integrity verification scores by jointly considering state rarity, relational inconsistency, temporal transition abnormality, and linkage corruption. Fourth, it demonstrated through controlled experimental scenarios that DI-based graph features can provide interpretable signals for detecting credential theft, device spoofing, impossible travel, session hijacking, bot-driven access, and lateral movement. Finally, the proposed framework extends identity analysis from post-event detection toward future attack direction estimation based on graph-structural changes and temporal correlation. This study proposed a graph-based framework for the mathematical modeling, anomaly detection, integrity verification, and future attack-direction estimation of DI in dynamic cyber environments. Unlike conventional DI management approaches that mainly treat user identity as a static object verified through one-time authentication, this study redefined DI as a multidimensional and dynamic structural object composed of human, device, location, service, and behavioral attributes. This formulation enables continuous integrity verification and context-aware trust evaluation beyond simple authentication success or failure. The proposed framework represents the relationships among DI components as a graph structure, allowing abnormal identity transitions, unauthorized behavioral changes, and structural integrity violations to be detected from a relational perspective. Rather than focusing only on event-level anomalies, the proposed approach analyzes connection patterns and interactions among identity components, thereby enabling the identification of security threats that may not be evident from individual attributes alone. In addition, by incorporating temporal signal correlation analysis, the framework provides a mathematical basis for estimating possible future attack directions from graph-structural changes, rather than a fully validated forecasting system for future attacks. The experimental evaluation combined public cybersecurity datasets with DI-based synthetic attack scenarios to assess the feasibility of the proposed method. The results showed that the graph-based DI model can effectively distinguish normal identity behavior from abnormal behavior under controlled experimental conditions. Compared with conventional behavior-based anomaly detection methods, the proposed approach provided interpretable structural signals by incorporating relationship information among DI components. These findings suggest that DI can be utilized not merely as authentication information, but as a continuously observable and analyzable security entity. Nevertheless, this study has several limitations. First, due to the lack of public datasets specifically designed for DI security analysis, some experiments relied on synthetic scenarios, which may limit generalizability to real-world environments. Second, although the proposed model effectively reflects structural relationships among DI attributes, its adaptability to concept drift in highly dynamic environments has not been sufficiently considered. Third, although this study formulates future attack-direction estimation based on attack-state transition and correlation-based path inference, quantitative evaluation of attack prediction accuracy, attack-path prediction success rate, path-level precision and recall, and forecasting horizon was not conducted due to the absence of DI-level ground-truth attack-path sequences. Therefore, the attack prediction component should be interpreted as a mathematical formulation and feasibility-level analysis rather than as a fully validated predictive model. Future work will expand empirical validation using real-world data collected from various DI ecosystems, including enterprise environments, IoT environments, and medical information

systems. In addition, Graph Neural Networks (GNNs) and temporal graph learning techniques will be investigated to improve adaptive anomaly detection and to quantitatively evaluate attack-direction estimation in dynamic environments. Future studies will also construct temporally labeled DI attack-path datasets and evaluate prediction performance using metrics such as top-k target prediction accuracy, attack-path success rate, path-level precision and recall, mean reciprocal rank, and horizon-wise forecasting accuracy. Furthermore, the proposed framework will be extended toward next-generation DI security by integrating it with Decentralized Identity (DID) and Zero Trust security architectures. In conclusion, this study reinterprets DI as an active security object that enables continuous integrity verification and anomaly detection, while providing a mathematical foundation for future attack-direction estimation, rather than as a static authentication mechanism. Therefore, the main contribution of this study is not only the use of graph features for anomaly detection, but also the establishment of a mathematical foundation for treating DI as an active, continuously verifiable, and predictive security object. In this context, the term predictive refers to the proposed formulation for estimating possible future attack directions, and not to a fully validated operational forecasting capability.

Author Contributions

H.Ko. conceptualization of this study, methodology, writing.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Data Availability Statement

The public datasets used in this study are available from their original sources. The CICIDS2017 dataset is publicly available from the Canadian Institute for Cybersecurity, and the IoT-23 dataset is publicly available from the Stratosphere Laboratory. The DI-based synthetic attack scenarios used in this study were generated by the authors to simulate identity-related attacks, including credential theft, device spoofing, impossible travel, session hijacking, bot-driven abnormal access, and lateral movement. The generated synthetic scenario data and preprocessing scripts are available from the corresponding author upon reasonable request.

ACKNOWLEDGMENTS

Conflicts of Interest

The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript.

DI	**Digital Identity**
DF	Digital Fingerprint
GNN	Graph Neural Network
IoT	Internet of Things
IoMT	Internet of Medical Things
DDoS	Distributed Denial of Service
OC-SVM	One-Class Support Vector Machine
FPR	False Positive Rate
ROC-AUC	Receiver Operating Characteristic–Area Under the Curve

REFERENCES

1. Goel, A.; Rahulamathavan, Y. A Comparative Survey of Centralised and Decentralised Identity Management Systems. Analysing Scalability, Security, and Feasibility. *Future Internet* **2025**, *17*(1), 1. https://doi.org/10.3390/fi17010001.
2. Fang, J.; Zhang, X.; Liu, Y.; Wang, H. Privacy-Enhanced Distributed Revocable Identity Management Scheme Based on Self-Sovereign Identity. *Journal of Cloud Computing* **2024**, *13*, 67. https://doi.org/10.1186/s13677-024-00715-8.
3. Böttinger, K.; Wiesmaier, A.; Krombholz, K. Analyzing the Threats to Blockchain-Based Self-Sovereign Identities by Conducting a Literature Survey. *Applied Sciences* **2024**, *14*(1), 139. https://doi.org/10.3390/app14010139.
4. Kim, T.; Park, J.; Lee, H. A Comprehensive Approach to User Delegation and Anonymity within Decentralized Identifiers for IoT. *Sensors* **2024**, *24*(7), 2215. https://doi.org/10.3390/s24072215.
5. Wessel, L.; von Solms, B.; Furnell, S. Blockchain-Driven Decentralized Identity Management. An Interdisciplinary Review and Research Agenda. *Information & Management* **2024**, *61*(6), 104005. https://doi.org/10.1016/j.im.2024.104005.
6. Suzuki, S.; Nakamura, T.; Yamamoto, K. Current Status of Decentralized Identifiers and Verifiable Credentials. *IEICE Fundamentals Review* **2024**, *18*(1), 42–55. https://doi.org/10.1587/essfr.18.42.
7. Ekle, O.A.; Eberle, W. Anomaly Detection in Dynamic Graphs. A Comprehensive Survey. *ACM Computing Surveys* **2024**, *56*(8), 1–38. https://doi.org/10.1145/3641234.
8. M. Zhong, M. Lin, C. Zhang, and Z. Xu, A survey on graph neural networks for intrusion detection systems. Methods, trends and challenges, *Computers & Security*, vol. 141, article 103821, 2024, https://doi.org/10.1016/j.cose.2024.103821.
9. Zhang, Y.; Li, K.; Xu, H.; Sun, R. Temporal Graph Learning for Dynamic Network Anomaly Detection.

Expert Systems with Applications **2023**, *225*, 120145. https://doi.org/10.1016/j.eswa.2023.120145.

10. Almseidin, M.; Alzubi, A.; Alkasassbeh, M. Artificial Intelligence-Based Cyber Attack Detection and Prediction. A Survey. *Electronics* **2023**, *12*(11), 2440. https://doi.org/10.3390/electronics12112440.
11. Rashid, M.; Khan, S.; Ahmed, R. Machine Learning-Based Predictive Cybersecurity Frameworks. A Comprehensive Review. *Computers & Security* **2024**, *138*, 103640. https://doi.org/10.1016/j.cose.2024.103640.
12. Z. Wang, M. Fei, Y. Xiong, and A. Wang, Dynamic Attack Path Prediction and Visualization for Industrial Cyber-Physical Systems Under Cyber Attacks, in *Proceedings of the 2024 43rd Chinese Control Conference (CCC)*, Kunming, China, 2024, pp. 9005–9010, https://doi.org/10.23919/CCC63176.2024.10662270.
13. S. Rose, O. Borchert, S. Mitchell, and S. Connelly, *Zero Trust Architecture*, NIST Special Publication 800-207, National Institute of Standards and Technology, 2020, https://doi.org/10.6028/NIST.SP.800-207.
14. Singh, A.; Kumar, V.; Sharma, P. Identity-Centric Zero Trust Security Framework for Enterprise Systems. *Future Generation Computer Systems* **2024**, *151*, 233–247. https://doi.org/10.1016/j.future.2024.01.015.
15. S. K. Jena, R. C. Barik, and R. Priyadarshini, A systematic state-of-art review on DI challenges with solutions using conjugation of IOT and blockchain in healthcare, *Internet of Things*, vol. 25, article 101111, 2024, https://doi.org/10.1016/j.iot.2024.101111.