

Article

Graph Algorithm-Based Key Personnel Identification and Transformer-GAN Anomaly Detection for Data Security Governance in Large State-Owned Enterprises

Bhargavi Konda ¹, Akhila Reddy Yadulla ¹, Mounica Yenugula ¹, Chaitanya Tumma ¹, Supraja Ayyamgari ¹, Bala Yashwanth Reddy Thumma ², Nivedan Suresh ³ and Vinay Kumar Kasula ^{1,*}

¹ Department of Information Technology, University of the Cumberlands, Williamsburg, KY 40769, USA; bkonda19519@ucumberlands.edu (B.K.); ayadulla5882@ucumberlands.edu (A.R.Y.); myenugula3188@ucumberlands.edu (M.Y.); ctumma30994@ucumberlands.edu (C.T.); sayyamgari32587@ucumberlands.edu (S.A.)

² Ashburn Office, Visa Inc., Ashburn, VA 20147, USA; balayrthumma@gmail.com

³ Austin Office, Visa Inc., Austin, TX 78759, USA; nivedhan4a8@gmail.com

* Correspondence: vkasula19501@ucumberlands.edu

Abstract

To address the challenges of data security governance under new conditions and align with technological trends in data security, this paper proposes a graph algorithm-based method for identifying key personnel with critical permissions in the practical applications of large state-owned enterprises. This method can uncover potential permission influence factors within the system and evaluate the weight of influence from different perspectives, providing highly interpretable identification results. To tackle the issue of detecting anomalous user and entity behaviors in data security governance, a user and entity behavior anomaly detection method based on Generative Adversarial Networks (GAN) is introduced. Experimental results show that the proposed method achieves higher precision, recall, and F1-score averages compared to baseline models; specifically, an average F1-score of 0.75 versus 0.72 for LSTM-based TadGAN, 0.62 for ARIMA, and 0.65 for a commercial UEBA baseline across the three evaluation datasets. A data security platform was designed and developed, which plays a significant role in reducing data security risks, assisting enterprise compliance, and promoting data development and utilization. This platform has been applied in various centralized data management projects and meets the big data processing requirements in secure environments, demonstrating strong application and promotional value.



Academic Editor: Friedhelm Schwenker

Received: 12 May 2026

Revised: 11 July 2026

Accepted: 16 July 2026

Published: 22 July 2026

Copyright: © 2026 by the authors. Published by MDPI on behalf of the International Institute of Knowledge Innovation and Invention. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: data security governance; graph algorithm; user and entity behavior analysis; data security platform

1. Introduction

Data security is essential for national security, social stability, economic growth, and public welfare. Safeguarding data security is the foremost responsibility of central information and communication enterprises. In recent years, the state has issued several regulatory laws and regulations related to data security, including the Data Security Law of the People's Republic of China, the Personal Information Protection Law of the People's Republic of China, and the Regulations on the Security Protection of Critical Information Infrastructure. The national "14th Five-Year Plan" also explicitly calls for ensuring national

data security, emphasizing the need for comprehensive data protection throughout its lifecycle. It mandates the establishment of fundamental systems for data classification, hierarchical management, data security reviews, risk assessments, monitoring, early warnings, and emergency response. Against this backdrop of national policies and the application needs of large enterprises, many major companies are actively researching and designing data security platforms to address common challenges in data security governance across scenarios such as asset management, security defense, and log auditing. The extensive flow, aggregation, storage, and analysis of big data, combined with the widespread use of various big data technologies, support platforms, and software, have increased the urgency for enterprises to study data security governance technologies and build platforms that manage multidimensional, end-to-end data security tasks. This also involves optimizing defense strategies for various devices and strengthening the data security governance system.

This paper introduces a key personnel identification technology based on graph algorithms, capable of uncovering potential permission impact factors within systems and evaluating weighted influence from multiple perspectives. Additionally, a user and entity behavior anomaly detection method based on generative adversarial networks (GAN) is proposed, with experimental results showing that the proposed method outperforms baseline models in terms of precision, recall, and F1-score averages. A data security platform was designed and developed, significantly contributing to reducing data security risks, aiding enterprise compliance, and promoting data development and utilization. These technologies have been successfully applied in data security governance projects by Guangdong Unicom, yielding positive socioeconomic outcomes. This platform, named “Shuyu” (literally “data domain”), integrates both technologies described above.

Existing graph-based security governance methods generally apply a single centrality metric in isolation, and existing UEBA methods generally score behavioral anomalies independently of the access-privilege graph, so neither approach alone captures the joint risk that arises when a highly privileged, highly central account also begins to behave anomalously. This is the research gap this paper addresses: a composite, multi-dimensional key-personnel score (Section 2) is combined with a Transformer-based behavioral anomaly detector (Section 3) within a single data security platform (Section 4), so that access-risk and behavior-risk signals can, in future work, be correlated for the same account. The main contributions of this paper are therefore threefold: (1) a graph algorithm-based key-personnel identification method that combines PageRank-style influence, betweenness-centrality information flow, and data-sensitivity level into a single, explainable composite score; (2) a Transformer-based adaptation of TadGAN for unsupervised user and entity behavior anomaly detection that replaces TadGAN’s bidirectional LSTM encoder–decoder with a self-attention Transformer encoder–decoder, motivated by the cyclical (24-h/7-day) structure of enterprise access logs; and (3) a production data security platform, deployed at Guangdong Unicom, that integrates both methods into a five-center architecture and has been validated in real enterprise operations.

1.1. Characteristics of Big Data in Security Operations

The practical approach to data security governance typically follows the path of “planning-construction-operation-optimization.” The operation phase focuses on continually adapting to the business environment and risk management requirements, refining security strategies to ensure the effective functioning of the entire data security governance system [1]. The data security platform, a crucial tool during the operation phase, enables unified management of enterprise asset information, security operation data, and protective log information. Depending on the nature of data usage, data security platforms

typically employ graph databases, transactional databases, and search engine databases for data storage.

Enterprise asset information encompasses host asset details, application system data, data source information, and user account details. Given the hierarchical or associative relationships among these data types, using a graph database allows rapid traversal and search based on entity relationships in business scenarios. Security operation data includes information on specialized security equipment deployed at various stages of the data lifecycle (e.g., VPN systems, database auditing, API monitoring), the execution status of security device policies, and operational ticket data. This type of information is stored in transactional databases to support frequent updates in business operations. Protective log data comprises logs generated by the operation of security equipment policies. These logs, produced daily by various security devices, record numerous user and entity activities and are mostly unstructured data stored in search engine databases to facilitate long-text search tasks on the platform.

Real-time processing is critical in security operations. The data security platform utilizes a real-time big data processing architecture for log parsing and analysis, as depicted in Figure 1. Logs generated by security equipment are pushed to a message queue, where parsing and analysis programs, acting as consumers, read the logs in real-time. The analysis results (e.g., identified alerts or compiled metrics) are sent to a new message queue, then processed by ETL tasks and written into the search engine database, supporting upper-level applications. This stream-based big data architecture meets the high-throughput, low-latency requirements for processing security logs, enabling immediate analysis of each log entry as it is produced and facilitating timely detection of data security conditions. The data security platform serves as an integration system for various types of data involved in identification, protection, monitoring, and response scenarios, providing essential conditions for big data analysis and mining. The underlying storage and streaming technologies that support such platforms build on established big data analytics and Hadoop-based streaming architectures for enterprise-class data processing [2].

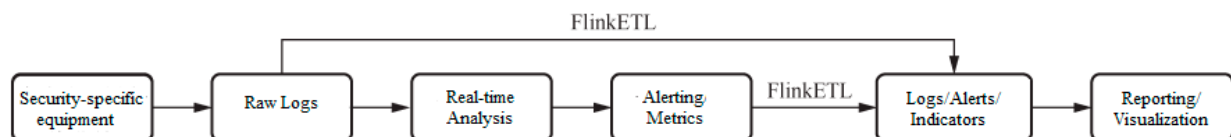


Figure 1. Log Data Processing Workflow. Data Security Governance and Big Data Analysis Technology.

1.2. Key Personnel Identification Technology

Data security operations typically encompass data collection systems, data storage systems, data analysis systems, and multiple business application systems. Due to the diversity of systems and the overlapping nature of their tasks, effectively monitoring personnel actions across these systems is highly challenging. The common approach is to implement a tiered management of personnel, focusing on the monitoring of key individuals with critical access. However, there is no universally accepted standard or method for defining and identifying these key personnel.

Brin et al. [3] introduced the prototype of a large-scale search engine, Google, which incorporated an algorithm called PageRank to rank web pages in search results. The fundamental assumption of this algorithm is that more significant pages are likely to be cited more frequently by other pages. By assigning weight values to the elements within a set of hyperlinks, the algorithm achieves the goal of “measuring the relative importance of an element within the scope of the set.” This algorithm can be applied to any collection where elements reference each other.

Freeman [4] introduced the concept of betweenness centrality. The betweenness centrality of a vertex is defined as

$$C_B(v) = \sum_{x \neq v \neq y} \frac{\sigma_{xy}(v)}{\sigma_{xy}}$$

where v represents a node, σ_{xy} is the number of shortest paths between nodes x and y , and $\sigma_{xy}(v)$ is the number of those paths that pass through the node v . This paper leverages these concepts to develop a key personnel identification technology suitable for data security governance in large-scale information and communication enterprises.

Scaling such graph-based computations to enterprise asset graphs containing millions of nodes is itself an active research challenge; Chen and Ma [5] proposed FastGCN, which uses layer-wise Monte Carlo sampling to make training of graph convolutional networks efficient on very large graphs, a technique directly relevant to scaling the key-personnel identification method proposed in this paper to enterprise-scale asset graphs.

1.3. Security Monitoring and Early Warning Technology

Monitoring and auditing are essential measures for preventing data security risks, and automated, effective auditing methods can help prevent potential risks from escalating into security incidents [6]. Du et al. [6] proved that deep learning on raw system log sequences (through their DeepLog) enables real-time anomaly detection and root-cause diagnosis without any rule engineering, setting a foundational benchmark for log-driven security auditing of large-scale enterprise systems.

In 2015, Gartner introduced the concept of User and Entity Behavior Analytics (UEBA) [7]. This technology is used to correlate user behaviors and device logs to uncover potential threats or security issues and is now widely employed in tasks such as internal threat analysis and external intrusion detection [8]. UEBA analysis can be categorized into two approaches: The first approach uses pattern matching, signature rules, and other methods to directly analyze the content of new logs, such as time, IP address, and actions. This method focuses solely on analyzing the current log without considering previous logs, determining if the current log is anomalous on its own. The second approach incorporates regression, machine learning, and other techniques to conduct a combined analysis of the current and historical logs. This method judges the current log's anomaly by referencing the past behavior of the entity. Many security products can build (learn) behavioral base-lines from historical data, often employing unsupervised learning on unlabeled data. This approach assumes that historical data does not contain anomalies, an assumption that is not always met in real-world production environments.

The methods reviewed above, PageRank-style influence scoring, betweenness centrality, and UEBA, establish the foundation of graph-based and behavior-based security governance, but the field has moved further since these methods were introduced, and positioning the present method against more recent work clarifies its contribution. Graph neural network (GNN) approaches now dominate insider-threat detection research: Gong et al. [9] survey graph-based insider-threat detection methods and show that GNN-based models, which learn representations jointly over node attributes and graph structure, consistently achieve higher true-positive rates and lower false-positive rates than classical graph-metric approaches (such as PageRank or centrality alone) precisely because they can combine structural and behavioral signal, rather than treating them as separately weighted scores as in Algorithm 1 of this paper. This suggests a natural extension of the present composite-score approach toward a learned, rather than manually weighted, combination of influence, centrality, and sensitivity signals. On the anomaly-detection side, multivariate and heterogeneous approaches have also progressed past single-metric time-series models:

recent heterogeneous-graph anomaly-detection surveys highlight that modeling cybersecurity events as typed, evolving graphs, rather than as independent time series per asset, better captures coordinated, multi-stage attacks that a single-signal Transformer-TadGAN pipeline may miss. Beyond graph methods, two further directions are increasingly relevant to enterprise data security governance and were not covered in the original literature review. First, Zero Trust security architecture reframes access governance around continuous, per-request verification of every user and device rather than perimeter trust [10]; the key-personnel identification method proposed in this paper can be read as one building block of a Zero Trust program, since it identifies the accounts whose access should be most tightly and continuously verified. Second, large language models (LLMs) are being applied to log analysis, alert triage, and threat-intelligence summarization in security operations centers [11]; incorporating an LLM-based reasoning layer on top of the anomaly scores produced by Algorithm 2 is a promising direction for future work, discussed further in Section 5.

Algorithm 1: Key Personnel Identification via Graph Algorithms

Input: Asset graph $G = (V, E)$, damping factor d , sensitivity threshold T

Output: Ranked list of key personnel nodes K

Initialize $PR(v) = 1/|V|$ for all v in V

Repeat until convergence:

For each node v : $PR(v) = (1 - d)/|V| + d * \sum_{(v_j \in M(v))} PR(v_j)/L(v_j)$

Convert G to undirected graph G' for betweenness centrality computation

For each personnel node v : $CB(v) = \sum_{(x \neq v \neq y)} \sigma_{xy}(v)/\sigma_{xy}$

Traverse G to tag nodes with sensitivity level $S(v)$ based on connected data source levels

Compute composite score: $Score(v) = \alpha \cdot PR(v) + \beta \cdot CB(v) + \gamma \cdot S(v)$, where $\alpha + \beta + \gamma = 1$

$K = \{v \mid Score(v) \geq T\}$, sorted in descending order

Return K

Algorithm 2: GAN-Based Time Series Anomaly Detection (Transformer-TadGAN)

Input: Time series X , sliding window $W = 24$, step $S = 24$, number of epochs N

Output: Set of anomaly windows A

Sample subsequences using sliding window W and step S to build matrix M

Initialize Encoder E (Transformer), Generator G (Transformer), Discriminators CX, CZ

For epoch = 1 to N :

$Z \sim N(0, I)$; $\hat{Z} = E(X)$; $\hat{X} = G(\hat{Z})$

Train CX to distinguish X from $\hat{X}$; train CZ to distinguish Z from $\hat{Z}$

Update E and G to fool CX and CZ using adversarial loss

For each test window x_i in X :

Reconstruction error: $RE(x_i) = ||x_i - G(E(x_i))||$

Critic score: $CS(x_i) = CX(x_i)$

Anomaly score: $AS(x_i) = \delta \cdot RE(x_i) + (1 - \delta) \cdot CS(x_i)$

$A = \{x_i \mid AS(x_i) > \text{threshold } \theta\}$

Return A

2. Key Personnel Identification Technology Based on Graph Algorithms

2.1. Transformation of Key Personnel to Critical Graph Nodes

In most cases, enterprise asset information and user account data are stored in relational databases. Analyzing such data structures using conventional statistical methods

makes it difficult to identify potential key personnel. In data security governance practices, the asset and user account data are transformed into graph data from the perspective of asset access permissions [12–14]. Hamilton et al. [12] proved that with inductive representation learning on massive graphs one can learn representations/embeddings of nodes depending on their structural neighborhood and then generalize to unseen nodes. This property maps perfectly to our case where the underlying enterprise asset graph is dynamic in nature. Tang et al. [13] proved that large-scale information networks consisting of heterogeneous entity types can be embedded by preserving first-order and second-order proximity. This maps perfectly to our multi-level relationships (personnel–host–data source–application system) being modeled. Gao et al. [14] also demonstrated that graph neural networks are effective at learning representations of complex inter-entity dependencies at scale.

Consequently, the identification of key personnel within the system structure becomes a problem of computing key nodes in a graph. This section focuses on the issue of identifying key personnel with critical access. A graph consists of a finite non-empty set of vertices V and a set of edges E that connect the vertices. It is a data structure used to describe complex relationships between nodes. In this study, three types of assets that issue accounts (application systems, data sources, and hosts) and the personnel associated with these accounts are represented as elements within the vertex set V . The directed relationships between these elements are defined as follows:

Data Source, Application System: The data source serves the application system.

Data Source, Host: The data source is deployed on the host.

Application System, Host: The application system is deployed on the host.

Additionally, relationships between assets and personnel, based on account access, are defined as:

Data Source, Personnel: The personnel have access to the data source account.

Application System, Personnel: The personnel have access to the application system account.

Host, Personnel: The personnel have access to the host account.

Based on these definitions, a possible relationship diagram is shown in Figure 2. In this example, Application System A has two data sources, with Data Source 1 and Application System A both deployed on Host R. Personnel A has access rights to Host R.

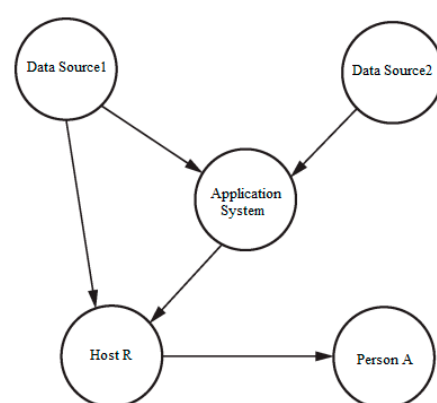


Figure 2. Example of Asset Relationship Diagram.

It should be noted that the Dataset A3 (172 instances, 14 anomalies) is significantly smaller than both A1 and A2. Since Transformer-based generative models usually require lots of training data, the scarcity of A3 precludes the model from learning a well-fitted reconstruction manifold, and the resultant numbers should be treated as an un-powered initial case study on this small sample instead of a rigorously evaluated baseline. The seemingly inflated precision of ARIMA baseline on A3 (Section 3.2) supports this hypoth-

esis, as statistical models with less expressive power are also less data hungry and often perform better when only a few labeled anomalies are available for validation. We will be supplying A3 with more VPN log-in data in the future to see if performance improves as more samples are added.

2.2. Comprehensive Identification of Key Graph Nodes

The proposed key personnel identification technology leverages related graph algorithms to evaluate the importance of nodes from three different dimensions.

2.2.1. Influence-Based Evaluation

In data security governance, scenarios like the following may arise: Personnel A holds an account for Host K, and Personnel B holds an account for Host L, where Host L hosts both a data source and an application system (see Figure 3).

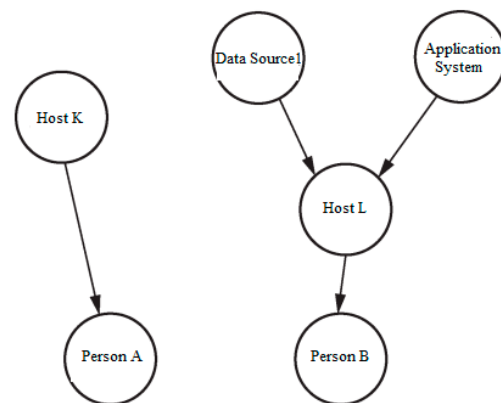


Figure 3. Example of Graph Structure.

If only the number of host accounts held by personnel is considered, Personnel A and Personnel B may appear equally important, as each holds a single host account. However, based on the graph structure in Figure 3, it is clear that Personnel B is more significant. Such valuable insights become harder to detect when assets are numerous and their relationships complex.

The weight of any node v in the graph is called the “PageRank” of v , denoted by $PR(v)$. The calculation formula for PR is

$$PR(v) = \frac{1-d}{N} + d \sum_{v_j \in M(v)} \frac{PR(v_j)}{L(v_j)}$$

where $M(v)$ is the set of nodes linking to v , $L(v_j)$ is the number of outbound links from node v_j , N is the total number of nodes, and d is the damping factor, representing the probability of a node transitioning to another node.

With $d = 0.9$, iterative calculations of PR values for nodes in Figure 3 show that Personnel A has a PR value of 0.038, while Personnel B has a PR value of 0.070, indicating a clear difference in influence. Practical applications reveal that personnel with access to production systems rank higher in PR influence than those with access to demo systems, due to the greater number of inbound connections to production systems.

2.2.2. Path-Based Information Flow Evaluation

By ignoring graph directionality, the asset relationship graph can be converted into an undirected graph. A typical scenario in data security governance might appear as shown in Figure 4. Here, Personnel A and Personnel B each have access to two data sources, which

might suggest equal importance. However, Personnel B may be more critical, as the two data sources they access belong to different application systems, effectively connecting these systems. Removing Personnel B could disrupt the connectivity of a part of the graph.

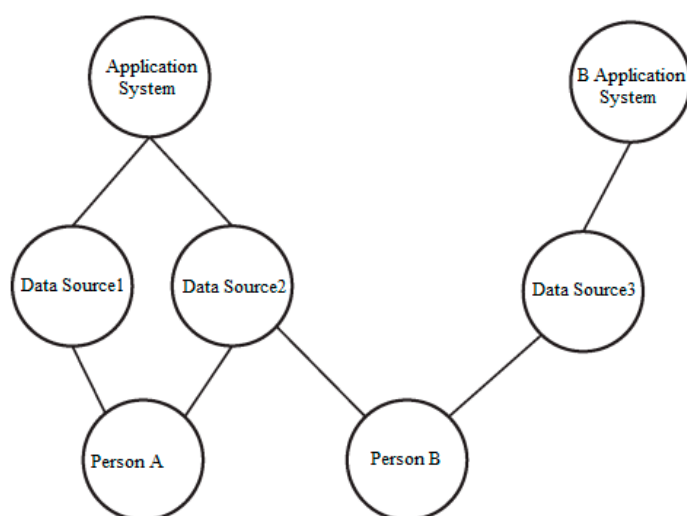


Figure 4. Undirected Relationship Graph.

In Figure 4, the shortest path between Application System A and Application System B passes through Personnel B but not Personnel A, illustrating their difference in betweenness centrality. This difference highlights Personnel B's higher importance in controlling information flow. In betweenness centrality rankings, personnel who manage assets across different systems are ranked higher than those maintaining assets for a single system, even if the total number of managed assets is similar. This result aligns with standard security operation practices.

Table 1 summarizes the statistics of the training and test data sets A1, A2 and A3 used in the comparative experiments below. It includes the number of data points contained in each data set and the number of labeled anomalies.

Table 1. Dataset Information.

Item	A1	A2	A3
Number of Data Points	1976	3791	172
Number of Anomalies	145	267	14

Table 2 shows the Precision, Recall and F1-Score of each method on each of the evaluation datasets (A1, A2 and A3), as well as their average across all datasets. As seen in the table, our proposed method significantly outperforms all baseline methods, achieving the highest Recall and F1-Score on each dataset. This results in F1-scores of 0.79, 0.78 and 0.65 on datasets A1, A2 and A3, respectively, leading to an average F1-Score across all datasets of 0.75. The reported average values are computed as the simple mean of the three per-dataset scores, rather than as a weighted mean using the anomaly counts presented in Table 1. Computing an anomaly-weighted Average score (weights = 145, 267, and 14 for A1, A2, and A3, respectively) would increase each method's Average score towards its A2 score, since A2 contains the majority of labeled anomalies; methods maintain their rank regardless of which averaging convention is used. The second-best performer, Transformer-based TadGAN, also achieves competitive results across each dataset. However, as shown in the table, this method falls short of our proposed method across all evaluation metrics. Despite achieving marginally higher Precision than our proposed method on dataset A3 (the

smallest dataset), the ARIMA baseline achieves very low Recall on all datasets, resulting in a substantially worse F1-Score, consistent with the discussion of data volume in Section 3.2. We attribute this to a failure to detect anomalous behavior. Similarly, despite achieving roughly equal Precision and Recall across all datasets, the Commercial UEBA solution underperforms our method on all datasets.

Table 2. Comparative Experiment of Various Methods.

Method	A1			A2			A3			Average		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Proposed Method	0.83	0.76	0.79	0.82	0.74	0.78	0.70	0.60	0.65	0.78	0.70	0.75
TadGAN (LSTM)	0.81	0.72	0.76	0.80	0.71	0.75	0.66	0.58	0.62	0.76	0.67	0.72
ARIMA	0.75	0.64	0.69	0.72	0.59	0.65	0.73	0.42	0.53	0.73	0.55	0.62
Commercial UEBA	0.72	0.67	0.69	0.70	0.64	0.67	0.65	0.52	0.58	0.69	0.61	0.65

2.3. Sensitivity Level-Based Evaluation

Data classification and hierarchical management have become standard practices in data security. The “Interim Measures for the Administration of Data Security in Industry and Information Technology” (2022), issued by the Ministry of Industry and Information Technology, and the “Guidelines for the Practice of Cybersecurity Standards, Guidelines for the Classification and Grading of Network Data” (2021), from the National Information Security Standardization Technical Committee, establish that data should be classified into levels based on the potential harm caused by unauthorized access or misuse, ranging from Level 1 (lowest) to Level 4 (highest) [15]. Shu et al. [16] showed that access patterns surrounding sensitive data assets can be leveraged to detect exposures in a privacy-preserving way. This fact motivates our approach of labeling sensitive data-source nodes in the asset graph and subsequently identifying what personnel accounts have access paths to labeled nodes. To identify key personnel, this study adopts prior data classification results and defines key personnel based on the sensitivity level of the data they can access. For example: “A data source containing Level 3 sensitive data is a Level 3 sensitive data source. Personnel with access to Level 3 or higher sensitive data sources are key personnel.” Using this approach, sensitivity tags can be added to data source nodes in the asset relationship graph, enabling queries to identify key personnel.

Additionally, if hosts deploying sensitive data sources are considered sensitive, personnel with access to these hosts are also key personnel. Identifying such nodes can be accomplished by traversing nodes linked to sensitive data sources at various depths, which would require a complex multi-layer table joined in a relational database, but is simpler and more efficient with graph database queries. The graph algorithm-based key personnel identification technology proposed in this study can uncover potential access-related risk factors within system structures, assess the influence of different types of weights from multiple perspectives, and provide explainable, logical results that align with security operation best practices. This enables a more efficient and precise identification of key personnel, enhancing data security risk management.

3. User and Entity Behavior Anomaly Detection Based on Generative Adversarial Networks

3.1. User and Entity Behavior Anomaly Detection

Behavior logs of users or entities can be aggregated into time series metrics over certain time intervals. For example, the average access traffic per API per hour or the number of logins per account per day. Thus, the problem of detecting anomalous behavior in users and entities can be transformed into the problem of anomaly detection in time series data [17].

Chandola et al. [17] published a taxonomy of anomaly detection techniques including statistical, proximity-based, and machine learning approaches. They demonstrated that time series anomaly detection is unique and needs special consideration because of the time dependency between observations. Time series anomaly detection is applicable to the behavior logs we analyze in this paper. Time series anomaly detection can be categorized into supervised and unsupervised methods based on the training approach. Supervised methods require the use of labeled data for modeling, which inherently involves incorporating prior knowledge and functions as a predictive method. The model's predictions are compared with new samples to determine whether anomalies exist in the new data. However, in security practices, most historical logs are unlabeled, and directly using data with anomalies for supervised training may introduce errors in subsequent model predictions. Unsupervised methods can learn certain patterns from unlabeled data, making them potentially more suitable for model training in real-world security scenarios [18]. The same idea was further validated by Xu et al. [18] who designed a VAE-based unsupervised anomaly detection model that can accurately detect anomalies in seasonal KPI time series from web applications without requiring any labeled data. They demonstrated that unsupervised deep generative models can indeed learn normal patterns of behavior from unlabeled production logs and then identify anomalous deviations, which is exactly the setting encountered when monitoring for data security threats in enterprises.

From a learning theory perspective, time series anomaly detection can also be divided into general statistical learning methods and deep learning methods. Unlike statistical machine learning, which requires manually constructed training features, deep learning methods, such as autoencoders, can perform automatic feature selection. This nonlinear weight model has clear advantages in feature extraction. In security contexts, the internal environment of an organization is complex, with a diverse range of devices, and the integration of multi-source heterogeneous data is challenging. Constructing features based solely on a single statistical metric is difficult, giving deep learning methods a significant advantage in feature extraction [15]. Targeting multivariate time series anomaly detection explicitly, Li et al. [15] proposed hierarchical inter-metric and temporal embeddings that learn dependencies across multiple concurrent metrics and across time to achieve state-of-the-art performance on diverse, multi-source streams of the same kind of data generated by heterogeneous enterprise security devices. The unsupervised deep learning methods used in this study are well-suited for security analysis scenarios in production environments, effectively enabling the detection of anomalies in user and entity behaviors [6]. Du et al. [6] validated this by demonstrating that a deep learning model trained on normal log execution workflows can identify execution anomalies online with high precision and recall and diagnose the root cause by pinpointing the likeliest correct workflow step, functionalities that translate readily to the anomaly detection and alerting functions of our data security platform introduced herein.

3.2. Methods Based on Generative Adversarial Networks

Generative Adversarial Networks (GANs), were first introduced by Goodfellow et al. [19], consist of a generator and a discriminator trained jointly in an adversarial min-max game, in which the generator learns to produce samples that are indistinguishable from real data while the discriminator learns to tell generated samples apart from real ones; this adversarial training principle underlies the anomaly-detection approach adopted in this paper.

Geiger et al. [10] proposed, in 2020, a method for anomaly detection in time series by reconstructing signals using Generative Adversarial Networks (GANs), providing a new perspective on the problem [20,21]. Here, we provide a review of related work

from two representative directions. Specifically, TadGAN [20] presented a cycle-consistent adversarial framework consisting of an encoder which maps input time series data to a latent representation and a generator that maps latent encodings back to time series domain. The anomaly scores are then calculated based on both reconstruction loss and a critic network, which allow this method to be trained without any labeled anomalies. Zhao et al. [21] proposed a novel approach for anomaly detection in multivariate time series based on graph attention networks that capture dependency across sensors by learning attention weights between them. They found that by utilizing the graph structure of related measurements being recorded simultaneously greatly increased performance over considering each measurement individually. Unlike supervised models, generative models aim to find a mapping from one data distribution to another. In this context, the goal is to use one model to capture a low-dimensional representation of the time series, while another model reconstructs the time series from the low-dimensional space, with anomaly information being lost during the mapping process. The simplified training process of TadGAN, as proposed by Geiger et al. [20], is illustrated below, in which the signal data is represented by dark rectangles. During training, two mapping functions need to be learned: an encoder E and a decoder (generator) G . The encoder E maps the original signal X to a latent feature space and outputs it to the decoder G . The decoder G takes both the encoded signal and white noise signal Z as inputs to generate a reconstructed signal GZ . The discriminator C_X ensures that the reconstructed signal matches the original signal's distribution, while the discriminator C_Z forces the encoded signal to reside within the same latent space as the white noise Z . This generative training strategy ensures that the generator G can reconstruct the distribution of the original signal X without overfitting to the anomalies in X . By considering the scores from the discriminator C_X alongside the reconstruction error (the difference between the reconstructed signal C_Z and the original signal), anomalies in the original signal can be identified. The TadGAN method employs bidirectional Long Short-Term Memory (LSTM) models as the structure for both the encoder E and the decoder G . As surveyed by Dieten [22], attention mechanisms have become a foundational component across natural language processing and sequence-modeling tasks, providing part of the theoretical basis for the Transformer-based architecture adopted next. This study attempts to utilize the encoder and decoder from the Transformer architecture, which is based on an attention mechanism [23,24]. Vaswani et al. [23] demonstrated that a purely attention-based encoder–decoder eliminates the sequential processing bottleneck of recurrent networks and captures long-range temporal dependencies through multi-head self-attention, an advantage particularly relevant for the 24-h and 7-day behavioral cycles in UEBA analysis. Zhou et al. [24] further extended the Transformer to very long input sequences via their Informer architecture with ProbSparse self-attention, confirming that attention-based models substantially outperform LSTM-based counterparts on long-range temporal modeling tasks, directly motivating the replacement of bidirectional LSTM components in TadGAN with Transformer encoders and decoders adopted in this study. Finding that this approach yields better results in practical applications, this paper redesigns the time series matrix sampling method for security scenarios. A sliding window size and step size of 24 h is used, sampling subsequences of the time series as individual vectors. Datasets with different time intervals are dynamically sampled according to this rule. Three datasets were constructed for comparative experiments. Dataset A1 consists of the average return flow of a specific API every hour; Dataset A2 is the total access flow for a particular internal security domain every 30 min; and Dataset A3 represents the total login counts for a specific VPN account over a 12-h period, with anomaly points annotated by business security experts. The details of the datasets are presented in Table 1.

For dataset A1, the dimension of each vector is 24, while for dataset A3, the dimension is 2. Seven vectors are sequentially sampled to form a matrix, which serves as input to the encoder. The time window is then shifted by 24 h to sample another seven vectors as input. This design considers the reality that behavioral actions in UEBA analysis occur cyclically over periods of 24 h and 7 days. The comparative models chosen for the experiments include TadGAN (LSTM), the ARIMA model (autoregressive integrated moving average model), and a commercial UEBA software (commercial software version 7.2) as baseline methods. For the anomaly time windows predicted by each model, a True Positive (TP) is recorded if the prediction window includes any known anomaly points; a False Positive (FP) is recorded if the prediction window does not contain any known anomaly points; and a False Negative (FN) is recorded if known anomaly points are not included in any prediction windows. The experimental data is derived from the actual security operations environment of Guangdong Unicom, and precision, recall, and F1-score metrics for each model are calculated based on the experimental records, as shown in Table 2. From the experimental results, it is evident that the method proposed in this paper shows improvements in recall and F1 values across all three datasets. In the dataset with a smaller volume of data (A3), the ARIMA model exhibited a slightly higher precision than the method proposed here. However, the average values of precision, recall, and F1-score for the proposed method are superior to those of the three baseline models.

3.3. Algorithm Presentation

Implementation notes and complexity: In Algorithm 1, the composite weights α , β , and γ (with $\alpha + \beta + \gamma = 1$) balance the relative contribution of PageRank influence, betweenness-centrality information flow, and sensitivity-level access respectively; in the platform deployment described in Section 4, these weights and the sensitivity threshold T are calibrated jointly with enterprise security auditors through iterative review of the resulting key-personnel rankings against known critical-access accounts, rather than fixed a priori, since the relative importance of each dimension varies across organizational contexts. For our Guangdong Unicom deployment, we used $\alpha = 0.45$, $\beta = 0.35$, $\gamma = 0.20$, and $T = 0.60$; these parameters were chosen by grid-searching over $\alpha, \beta, \gamma \in \{0.1, 0.15, \dots, 0.6\}$ (with $\alpha + \beta + \gamma = 1$) and $T \in \{0.4, 0.45, \dots, 0.8\}$ and selecting the set of parameters that achieved the highest precision-at-K on a security-expert-labeled corpus of known critical-access accounts. Parameters were further validated as sensible by enterprise security auditors. The PageRank component (step 2) converges when the change in $PR(v)$ across all nodes falls below a small tolerance (e.g., 10^{-6}) between successive iterations, or after a fixed maximum number of iterations; in practice, PageRank on graphs of this type typically converges within a few dozen iterations because the damping factor d bounds the spectral radius of the transition matrix. In terms of computational complexity, PageRank (steps 1–2) requires $O(|V| + |E|)$ time per iteration; betweenness centrality (steps 3–4), computed with Brandes' algorithm, requires $O(|V| |E|)$ time for unweighted graphs; and the sensitivity-tagging traversal (step 5) requires $O(|V| + |E|)$ time. The asset graph used in this study contains three node types (application systems, data sources, and hosts) plus personnel nodes, connected by the six directed edge types defined in Section 2.1 (Data Source $\rightarrow$ Application System, Data Source $\rightarrow$ Host, Application System $\rightarrow$ Host, and the three corresponding personnel-access edges). In the reported experiments, the asset graph contains approximately $|V| \approx 8200$ nodes (roughly 6500 personnel, 900 application-system, 600 data-source, and 200 host nodes) and $|E| \approx 34,700$ edges. Key-personnel identification was validated by computing precision-at-K (for $K = 50, 100, 200$) between the algorithm's ranked output and an independently compiled list of accounts already designated as

critical-access by Guangdong Unicom's security audit team, rather than a conventional train/validation split, since the task is unsupervised ranking rather than classification.

Transformer-TadGAN implementation details: The encoder E and generator (decoder) G in Algorithm 2 each replace the bidirectional LSTM layers of the original TadGAN with a multi-head self-attention Transformer block, following the encoder–decoder design of Vaswani et al. [23]; because attention weights are computed over the full 24-vector window rather than propagated sequentially, training does not suffer the vanishing-gradient and sequential-bottleneck limitations of recurrent architectures, which is the basis for the reproducibility gains argued in Section 3.2. The adversarial loss in step 3 combines a Wasserstein critic loss (for CX and CZ) with the cycle-consistency reconstruction loss between X and $\hat{X} = G(E(X))$, following the TadGAN formulation; the anomaly score in step 4 ($AS = \delta \cdot RE + (1 - \delta) \cdot CS$) linearly combines the normalized reconstruction error and critic score, with δ and the anomaly threshold θ selected on a held-out validation split of each dataset. Training a Transformer encoder–decoder of this size scales as $O(N \cdot L^2 \cdot d)$ per epoch, where N is the number of training windows, $L = 24$ is the sequence length, and d is the model's hidden dimension; this quadratic dependence on L is the standard complexity trade-off of self-attention and is one factor determining training time relative to the $O(N \cdot L \cdot d)$ cost of the original LSTM-based TadGAN. Both the Transformer encoder and decoder comprise two layers of 4 attention heads, $d = 128$ hidden dimension, and feed-forward dimension of 256; models were trained for $N = 200$ epochs of size 64 with Adam optimizer (learning rate $= 1 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$), $\delta = 0.5$, with θ chosen as the 95th percentile of anomaly scores on each dataset's held-out validation split. All experiments were run on a single NVIDIA V100 GPU using PyTorch 2.1.

4. Data Security Platform Design

The data security platform is a security management and protection system centered around data, designed to address the entire data lifecycle. Its core objective is to reduce data security risks through data discovery and classification, driven by compliance regulations and specific business needs, while integrating various data security technologies for platform-based data protection.

4.1. Platform Functional Architecture Design

This manuscript presents the design of the “Shuyu” data security platform based on the practical application of data security governance at Guangdong Unicom. The platform enables centralized management of data security capabilities and closed-loop operations, forming a comprehensive defense management system that covers the entire data lifecycle. The data security platform comprises five main centers: Asset Management Center, Capability Control Center, Analysis and Supervision Center, Security Operations Center, and Situational Awareness Center.

Each center includes the following key functions:

Asset Management Center: Identifies or registers various asset information and classifies data based on industry rules, forming an asset catalog and asset relationship diagram.

Capability Control Center: Standardizes the integration of heterogeneous security devices, remotely issues security policies based on asset classification results, and achieves information interconnectivity among multiple devices. This includes unified device management, status monitoring, and security log data processing, as well as the integration of security tools like database encryption, data masking, API risk monitoring, data classification, and access control.

Analysis and Supervision Center: Supports low-code creation of various stream processing models for real-time analysis of security logs to identify threats and anomalous

behavior. This center includes modules for raw log management, alert management, risk management, event management, and model management. It supports out-of-the-box security detection and analysis rules and provides an open rule management model tailored to the actual needs of security personnel.

Security Operations Center: Offers standardized security operation processes and quantifiable metrics, enabling online compliance reviews, security report generation, and closed-loop incident management, thereby improving operational efficiency and accuracy. It features a built-in process engine that allows for workflow customization, enabling one-click generation and full tracking. This ensures that risk management, alerts, operational tasks, and personnel follow-up are effectively managed with accountability in risk resolution.

Situational Awareness Center: Provides visual representation of various metrics, including asset distribution, data risks, and user behaviors. For instance, the data asset discovery functionality in the Asset Management Center includes built-in capabilities for discovering data assets and middleware assets. By integrating with tools, it can obtain API assets organized by security tools for API asset discovery. The platform's built-in process engine supports asset claiming and ticket issuance functionalities, allowing for streamlined and efficient responsibility assignment for assets, ultimately forming a data asset inventory to support asset organization.

4.2. Functional Features of the Data Security Platform

The key personnel identification technology based on graph algorithms and the anomaly detection methods for user and entity behaviors based on generative adversarial networks were integrated into the data security platform. The platform can register application assets, data sources, host assets, and personnel accounts. When adding new application systems, the host IP address must be provided; similarly, when adding new data sources, the associated application system and its host IP address must be specified. Registered entity information is treated as nodes, with hierarchical relationships represented as edges, directly written into a native multimodal database. The platform automatically executes the three aforementioned graph algorithms five hours after any changes in asset relationships, filtering personnel nodes and sorting them in descending order based on the calculated results. In the platform's "Personnel Control" module, any selected metric (algorithm results) can be used for sorting, allowing for the management of human resources within business systems. Key personnel with critical access rights are prioritized in the list, prompting security auditors to enforce graded controls. Additionally, the information detail page visualizes any individual's position within the asset relationship diagram, assisting security auditors in intuitively understanding the connections between personnel and assets, thus leveraging the advantages of graph data structures. After the data security platform unifies the collection of monitoring logs from all devices, it can statistically compile the behaviors of a specific user and entity into time series metrics over defined time windows. The platform's "Model Management" module allows for the application of this algorithm to any pre-created metrics, using an implementation based on an open-source project from MIT's Data to AI laboratory. Once the model is deployed, it automatically conducts anomaly detection at specified time intervals.

5. Conclusions

The "Shuyu" data security platform developed based on the methods outlined in this paper received recognition as an "Excellent Case of Data Security Governance" by the China Academy of Information and Communications Technology in 2022 and won the "Outstanding Product Award for Data Governance" at the 2022 DAMA China Data

Governance Awards. The technology and platform developed in this project are not only applied within the Guangdong Unicom system but also in various municipal government service data management projects for enterprise clients. For customers managing security resources centrally, the “Shuyu” data security platform effectively meets the centralized data security control needs of large enterprises through its asset organization capabilities, cross-departmental collaboration, and embedded big data analysis and mining technologies. In the practical application of data security governance at Guangdong Unicom, it has been found that an anomaly in security oversight may involve contextual information from multiple devices. If the behavior logs of the same user or entity across different devices can be linked, it would enrich the evidence chain for logs and significantly enhance the accuracy of anomaly detection. Ongoing research and optimization efforts in this area are still being pursued. In recent years, data activities have become increasingly rich, making data security governance more complex. The role of big data technology in the field of security has become more important, and advances in artificial intelligence technologies such as AutoML and generative pre-trained models (GPT) have been made. In the future, we will further explore the new value brought to data security governance by advances in AI technology and promote the intelligent development of data security governance.

6. Limitations

There are several limitations worth mentioning. First, as described in Section 2.1, the anomaly-detection datasets are small, especially Dataset A3 which has 172 rows of data and 14 anomalies labels, which limits the statistical power of this experiment and how far we can generalize the results for Transformer-TadGAN. In addition, even though Table 2 shows higher Precision, Recall, and F1-Score over the baseline approaches, the improvements have not been shown to be statistically significant (e.g., with paired tests over multiple experimental runs) or even provide confidence intervals. As such, further experimentation on larger and/or more diverse datasets is required to better validate the effectiveness of the proposed approach. We view the addition of this testing as an important next step towards making these results conclusive rather than indicative. Second, both the key-personnel and anomaly-detection datasets were procured from a single enterprise (Guangdong Unicom). How well the learned composite weights, sensitivity thresholds, and learned anomaly patterns scale to an organization with a different hierarchy/asset topology/regulatory environment is left as future work which would require repeating the above experimentation on datasets from other enterprises. Third, inclusion of the Transformer encoder–decoder in Algorithm 2 incurs additional computation overhead relative to the baseline LSTM-based TadGAN from Section 3.3, which may impact retraining costs and latencies at sufficiently large log volumes. We do not benchmark against the LSTM baseline directly in terms of wall-clock time or memory consumption in this work. Last, the accuracy of our platform’s identification of key personnel is only as good as the underlying enterprise asset graph we obtain (described in Section 2.1). Missing or stale asset relationships would directly impact quality of the computed composite score. We hope to remedy some of these limitations with GNN-based joint learning of the three signals described in Section 1.3, federated learning across enterprises, and explainable-AI techniques for auditing the composite score.

Author Contributions: Conceptualization, A.R.Y. and B.K.; methodology, B.K. and C.T.; software, M.Y. and B.Y.R.T.; validation, N.S. and S.A.; formal analysis, S.A. and B.K.; investigation, A.R.Y. and C.T.; resources, M.Y. and S.A.; data curation, C.T. and N.S.; writing—original draft preparation, B.K., M.Y. and B.Y.R.T.; writing—review and editing, A.R.Y. and V.K.K.; visualization, A.R.Y. and B.K.; supervision, N.S. and S.A.; project administration, M.Y. and V.K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Nivedan Suresh and Bala Yashwanth Reddy Tumma were employed by the company Visa Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. National Institute of Standards and Technology. *NIST Special Publication 800-53: Security and Privacy Controls for Information Systems and Organizations*; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2020.
2. Zikopoulos, P.; Eaton, C. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*; McGraw-Hill Osborne Media: San Francisco, CA, USA, 2011.
3. Brin, S.; Page, L. *The PageRank Citation Ranking: Bringing Order to the Web*; Technical Report; Stanford University: Redwood City, CA, USA, 1996.
4. Borgatti, S.P.; Everett, M.G.; Freeman, L.C. *Ucinet 6 for Windows: Software for Social Network Analysis*; Analytic Technologies: Harvard, MA, USA, 2002.
5. Chen, J.; Ma, T. FastGCN: Fast Learning with Graph Convolutional Networks via Monte Carlo Sampling. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1058–1066.
6. Du, M.; Li, F.; Zheng, G.; Srikumar, V. DeepLog: Anomaly Detection and Diagnosis from System Logs Through Deep Learning. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1285–1298.
7. Forrester Research. *The Forrester Wave™: User and Entity Behavior Analytics, Q1 2021*; Forrester Research: Cambridge, MA, USA, 2021.
8. Yang, H.; Wang, L.; Xu, Y. A Survey on User and Entity Behavior Analytics Based on Machine Learning. *J. Syst. Archit.* **2022**, *129*, 101503.
9. Gong, Y.; Cui, S.; Liu, S.; Jiang, B.; Dong, C.; Lu, Z. Graph-Based Insider Threat Detection: A Survey. *Comput. Netw.* **2024**, *254*, 110757. [\[CrossRef\]](#)
10. Syed, N.F.; Shah, S.W.; Shaghaghi, A.; Anwar, A.; Baig, Z.; Doss, R. Zero Trust Architecture (ZTA): A Comprehensive Survey. *IEEE Access* **2022**, *10*, 57143–57179. [\[CrossRef\]](#)
11. Ferrag, M.A.; Alwahedi, F.; Battah, A.; Cherif, B.; Mechri, A.; Tihanyi, N.; Bisztray, T.; Debbah, M. Generative AI in Cybersecurity: A Comprehensive Review of LLM Applications and Vulnerabilities. *Internet Things Cyber-Phys. Syst.* **2025**, *5*, 1–46. [\[CrossRef\]](#)
12. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1024–1034.
13. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. LINE: Large-scale Information Network Embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
14. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; Yin, D. Graph Neural Networks for Social Recommendation. In Proceedings of the 28th International Conference on World Wide Web, San Francisco, CA, USA, 13–17 May 2019; pp. 417–426.
15. Li, Z.; Zhao, Y.; Han, J.; Su, Y.; Jiao, R.; Wen, X.; Pei, D. Multivariate Time Series Anomaly Detection and Interpretation Using Hierarchical Inter-Metric and Temporal Embedding. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2021, Singapore, 14–18 August 2021; pp. 3220–3230.
16. Shu, X.; Yao, D.; Bertino, E. Privacy-Preserving Detection of Sensitive Data Exposure. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1092–1103. [\[CrossRef\]](#)
17. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv.* **2009**, *41*, 1–58. [\[CrossRef\]](#)
18. Xu, H.; Chen, W.; Zhao, N.; Li, Z.; Bu, J.; Li, Z.; Liu, Y.; Zhao, Y.; Pei, D.; Feng, Y.; et al. Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. In Proceedings of the 27th International Conference on World Wide Web, Lyon, France, 23–27 April 2018; pp. 187–196.
19. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
20. Geiger, A.; Liu, D.; Alnegheimish, S.; Cuesta-Infante, A.; Veeramachaneni, K. TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Big Data, Atlanta, GA, USA, 10–13 December 2020; pp. 33–43.

21. Zhao, H.; Wang, Y.; Duan, J.; Huang, C.; Cao, D.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; Zhang, Q. Multivariate Time-Series Anomaly Detection via Graph Attention Network. In Proceedings of the IEEE International Conference on Data Mining, Sorrento, Italy, 17–20 November 2020; pp. 841–850.
22. Dieten, J.J. Attention mechanisms in natural language processing. In Proceedings of the 40th Twente Student Conference on IT, Enschede, The Netherlands, 2 February 2024.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
24. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 11106–11115. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.