The ISC Int'l Journal of Information Security

ISeCure

January 2026, Volume 18, Number 1 (pp. 99–120)

http://www.isecure-journal.org

Harnessing Deep Learning for Anomaly Detection in Log Data: A Comprehensive study

Kamiya Pithode [1,*], and Pushpinder Singh Patheja [1]

[1] *Department of SCSE, VIT Bhopal, M.P, India.*

ARTICLE INFO.

Article history:
Received: July 30, 2024
Revised: December 06, 2024
Accepted: October 24, 2025
Published Online: October 26, 2025

Keywords:
Multilayered neural network, Anomaly detection, Deep neural networks, System log, Anomaly detection, log analysis

Type: Review Article

doi: 10.22042/isecure.2025.470715.1155

ABSTRACT

With the increasing prevalence of online services, big data systems, and Internet of Things (IoT) devices, detecting anomalies in large system logs has become a significant concern. This study presents a systematic literature review of automated log analysis for anomaly detection from January 2017 to October 2024. The study's primary objective is to classify existing approaches into five types: hybrid, supervised, unsupervised, semi-supervised, and self-supervised, and to provide a comprehensive analysis of each technique based on its assumptions, benefits, limitations, computational complexity, and performance in practical applications. Additionally, it addresses the challenges and concerns associated with developing anomaly detection systems for real-life applications using deep neural networks. The survey's goal is not to perform a statistical analysis of the published methodologies but to classify them, highlight the key features of various deployed architectures, and focus on unresolved issues that require further investigation in this domain. The study offers valuable direction for researchers, emphasising the need for scalable, robust, and interpretable anomaly detection systems. This survey advances the understanding of current capabilities and highlights future directions for enhancing the reliability of complex systems.

1 Introduction

Large-scale systems need robust anomaly detection to swiftly identify unusual activities, minimise failures, and resolve issues to ensure safety and optimal performance. Log anomaly identification is challenging due to the vast amount of irrelevant log data, making human analysis impractical and necessitating automated approaches [113]. Machine learning algorithms analyse loglines to find patterns and notify operators of abnormalities [64]. Real-time monitoring solutions using deep neural networks are being developed to identify abnormalities without human involvement [20]. Various methods have been proposed to address log anomaly detection, including process mining, log clustering, temporal analysis, SVM, DT, and PCA [99]. Traditional machine learning algorithms are being replaced by supervised, unsupervised, and semi-supervised deep neural networks [4]. Supervised models need extensive training data, while semi-supervised and unsupervised models rely on the uniqueness of anomalies. Hybrid and self-learning approaches are also used to enhance anomaly detection in log data [31, 104]. This demonstrates the shift from traditional methods to advanced deep learning sys-

* Corresponding author.
Email addresses: kamiya.pithode2019@vit.ac.in, pspatheja@gmail.com
ISSN: 2008-2045

tems in detecting log data anomalies, emphasising adaptability, transparency, and real-time capability.

1.1 Evolution of log-based anomaly detection and handling approaches

Since its inception in 2003, automation in log analysis has continuously evolved [14]. Figure 1 illustrates the progress of anomaly detection technology. Initially, researchers used classic statistical techniques like Gaussian mixture models and distance-based approaches to find anomalies in log files

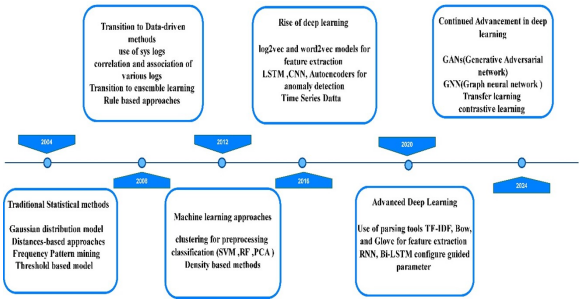

Figure 1. Evolution of deep neural network-based anomaly detection

Frequency pattern analysis tools explored connections between different logs. Rule-based systems with manually constructed rules were widely used for anomaly detection. Since 2010, clustering techniques have become popular for pre-processing, and machine learning algorithms such as Random Forest, Gaussian NB, Naive Bayes, and one-class SVM are commonly used for data classification and prediction capabilities. In 2016, researchers began using NLP algorithms like Word2Vec, TF-IDF, and Glove to extract features. With the increasing volume of log data, models were trained using deep neural networks like RNN, CNN, LSTM, and Bi-LSTM. Until the 2020s, research shifted to more complex deep learning structures, such as transformers, GAN, GNN, transfer learning, and pre-trained models, to identify intricate linkages and dependencies in log data. The emphasis on real-time detection led to the development of models suitable for dynamic and time-sensitive applications.

1.2 Understanding and Evaluating Surveys of existing log analysis approaches

There is a gap between academic research and industrial applications in log data anomaly detection due to a lack of survey papers on current methods. The survey of existing log analysis approaches is understood in Table 1. [65] compare five advanced models for automated anomaly detection, highlighting factors like early detection and noise affecting model efficacy. [49] uses log analysis to find anomalies in large-scale computing setups. [61] provides a toolkit for developers, evaluating six advanced anomaly detection methods and exploring self-learning technologies. Their review covers model topologies, data preprocessing, anomaly detection, and evaluations. [14] shows that log semantics improve model robustness against noise.

A gap exists between academic research and industrial applications in deep learning log data anomaly detection, primarily due to the lack of comprehensive survey papers on current methods. An outline of the existing Survey on log analysis approaches is presented in Table 1. [65] compare five innovative models for automated anomaly detection. [49] utilises log analysis to identify anomalies in large-scale computing settings. [61] bids a toolkit for developers by evaluating six advanced anomaly detection methods and exploring the possibility of self-learning technologies. [14] proves that including log semantics can augment model robustness, mainly in noise. [134] analyses various anomaly detection approaches.

1.3 Our Contribution

Deep neural network enhances log data anomaly detection by improving accuracy, flexibility, automation, and scalability, leading to safer systems. [61] categorises model training methods into supervised, semi-supervised, and unsupervised, emphasising the importance of feature extraction and suitable model architectures for log anomaly detection. However, a noted gap exists in classifying multilayered neural networks by training data labels. This research aims to fill that gap by classifying models based on data labelling and examining training methodologies, deep neural frameworks, and performance metrics. We identify five training methods: supervised, semi-supervised, unsupervised, self-supervised, and hybrid. Our insights help scholars and organisations understand deep learning techniques, aiding in developing practical training and monitoring strategies. Figure 2 shows the complete framework of this paper.

The study addresses the research issues listed below:

- **RQ1:** How does log data anomaly detection performance vary across neural network structures?
- **RQ2:** How do neural network models depend on labelled data, and what are their classifications: supervised, semi-supervised, unsupervised, self-supervised, and hybrid?
- **RQ3:** What are the constraints of current literature and guidelines that might inform future research directions?
- **RQ4:** How do various model architectures and hyperparameters affect the time delay of

Table 1. Summary of existing log analysis survey papers

Survey	Author finds	Performance	Classification	Metrics	Data set	Challenge	Future scope
This Survey (2024)	The survey focuses on log anomaly detection and classifies deep neural network approaches into supervised, unsupervised, semi-supervised, self-supervised, and hybrid categories.	No	Yes		Yes	Yes	Yes
[61]	Appropriate pre-processing enhances deep model performance, deep learning improves anomaly detection, and model performance varies across supervised, unsupervised, and semi-supervised approaches, suggesting careful model selection.	No	Yes	No	Yes	Yes	Yes
[65]	Performance is influenced by training data selection, dataset characteristics, and early detection capability.	No	No	No	Yes	No	Yes
[14]	A comprehensive evaluation of five popular neural networks and six state-of-the-art methods, providing insights into their performance and applicability.	Yes	Yes	Yes	Yes	No	Yes
[64]	The study categorises log data analysis approaches by clustering techniques and outlines objectives like overview, parsing, outlier detection, and dynamic anomaly detection.	No	No	No	Yes	Yes	Yes
[134]	The study reviews recent research on using Deep Neural Networks for anomaly detection to identify system behaviour.	No	No	No	Yes	Yes	Yes
[8]	The study discusses basic techniques, variants, advantages, limitations, and computational complexities and highlights open research issues and adoption challenges in deep learning.	No	No	No	Yes	Yes	Yes
[49]	Reviews the challenge of manual log inspection in large-scale distributed systems and evaluates six advanced log-based anomaly detection methods.	Yes	No	Yes	Yes	Yes	Yes

anomaly detection in real-world systems?

- **RQ5:** What is the Explainability and Availability of deep anomaly detection models?

1.4 Structure of paper

The remaining modules of this article are structured as follows: Section 2 describes the survey technique, which includes finding relevant studies and statistically analysing articles. Supervised, semi-supervised, unsupervised, self-supervised, or hybrid (Section 2.2). Section 2.3 analyses different neural network approaches. Section 3 details the log dataset's assessment characteristics, including evaluation criteria and benchmark models for comparison with past publications. we reviewed the survey and answered our research questions with detailed answers and future routes. Section 4 concludes this article.

Figure 2. Framework of paper

2 Survey Methodology

The literature review approach consists of three main steps: an initial search, refining results, and a recursive reference search by forming a search string. A thorough search was done for research papers from 2017 to October 2024. It used specific terms related to log data, anomaly detection, and deep learning in prominent online databases and digital libraries, as

mentioned in Figure 3. The second stage focused on refining results by the selection criteria for studies on log data anomaly detection using deep neural networks, including relevance, focusing on studies using deep learning specifically for log anomaly detection. Only recent papers (2017 to early 2024) are included to capture current advancements, and methodological rigour is required, with detailed descriptions of neural network techniques. Benchmarking or comparative analysis with existing methods is essential, as is accessibility, limiting the review to studies published in English. Lastly, studies must use adequately sized, standard log datasets relevant to future research, excluding those with insufficient data detail. The third step involved a recursive search to find references in selected studies and locate the original research on specific themes.

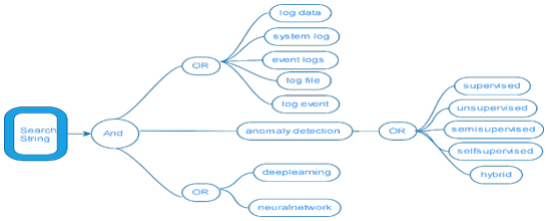

Figure 3. Formation of Search string

2.1 Survey Statistics

This segment provides survey statistics on anomalies detected in log data via multi-layered neural networks. It includes citation counts and annual publication numbers from 2017 to October 2024. The results show a growing scholarly focus, with most literature published in the past five years. Figure 4 shows that 109 of 124 articles were published from 2019 onwards. The number of publications is expected to continue rising beyond 2024. Figure 5 shows the distribution of 124 evaluated publications across different neural structures, with a strong focus on classical supervised learning (41 publications), Semi-supervised (30 publications), unsupervised learning (29 publications), and hybrid learning (18 publications), self-supervised learning (7 articles). This distribution showcases a varied research environment, emphasising supervised learning and a growing interest in data-efficient and adaptive methods like self-supervised and hybrid models

2.2 Classification of multilayered neural networks

Using advanced deep-learning techniques, multilayered neural network models detect uncommon patterns or outliers in data, which is particularly effective where anomalies are rare and underrepresented in training. [8] highlighted the growing significance of these anomalies. This article explores various deep neural network-based anomaly detection models categorised by training data type or label availability: supervised, semi-supervised, unsupervised, self-supervised, and hybrid. It analyses training assumptions, architecture complexity, computational demands, and practical applications of these approaches.

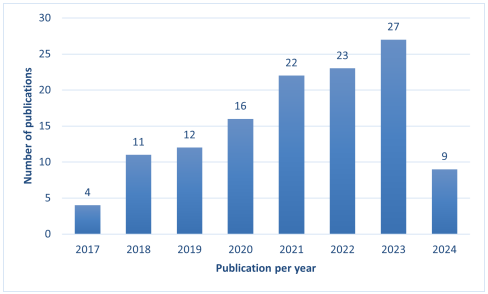

Figure 4. Number of publications per year

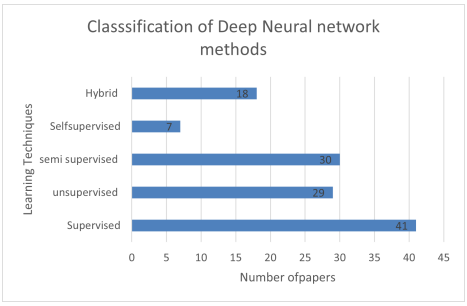

Figure 5. Classification of existing multilayered neural network methods

2.2.1 Supervised multilayered neural network

Supervised neural networks for anomaly detection in system logs require labelled data to extricate standard data from anomalous [83]. The accuracy of these models relies on the consistency and accuracy of the labels, and they assume that the data distribution remains even over time. However, noteworthy changes in log patterns may need regular model updates. A key constraint is a dependence on labelled data, which can be expensive and limited [49]. Moreover, these models suffer from data drift, where varying data features lead to performance degradation. They also face difficulties in generalising to new types of anomalies [134]. Training these models is computationally costly, requiring substantial resources, and inference can present potential latency, especially in real-time applications. Additionally, supervised models request significant memory, which can be challenging in resource-limited settings. To address these issues, multilayered architectures, such as CNNs, RNNs, LSTMs, and

Transformers, are often deployed, with hybrid models offering improved performance by leveraging the strengths of each architecture. Table 2 outlines various designs and approaches of supervised deep neural networks for anomaly detection.

2.2.2 Semi-supervised multilayered neural network

Semi-supervised anomaly detection methods for system logs assume that only a tiny portion of the data is labelled, and they depend on the assumption that similar distributions have been provided to labelled and unlabeled data [31]. These methods also assume occasional anomalies, with regular instances being more dominant. However, their usefulness relies on the quality of the labelled data, as inappropriate labels can deceive the learning procedure. Semi-supervised models are more complex to train, as they must balance labelled and unlabeled data [65]. They also suffer to identify new anomalies not represented in the labelled data.

These models require High computational resources, mainly when using pseudo-labeling or consistency regularisation [9]. Additionally, hyperparameter tuning is often needed to achieve optimal performance. Commonly used architectures in these models include Generative Adversarial Networks (GANs), autoencoders, Graph Neural Networks (GNNs), and self-training techniques, all of which help improve the detection of anomalies in log data. They are employed when unsupervised methods lack accuracy and obtaining sufficient labelled data for supervised methods is impractical [12]. These solutions classify anomalies based on specific data concepts. Table 3 describes various semi-supervised deep neural network structures for log anomaly detection.

2.2.3 Unsupervised Multilayered Neural Network

Unsupervised anomaly detection methods for system logs assume that no labelled data is available for training, and they rely on finding inherent patterns in the data to detect anomalies [124]. These methods also assume that most data represents normal behaviour, allowing them to identify deviations from this pattern. They depend on the ability to extract useful features from raw log data to differentiate normal from anomalous behaviour. Unsupervised methods frequently have higher false favourable rates and are difficult to evaluate since no labelled data exists for comparison. They may also struggle if the data distribution differs from the assumed pattern. Computationally, these methods can be resource-intensive and involve complex architectures, like autoencoders or clustering techniques, increasing training time and making model tuning more challenging. However, they have difficulty recognising patterns in complex, high-dimensional spaces. Optimising dimensionality reduction through autoencoders necessitates careful tuning, and these methods may struggle with complex or evolving anomalies [116]. Popular architectures in unsupervised anomaly detection include autoencoders, clustering methods, generative models like VAEs and GANs, and Graph Neural Networks (GNNs), all of which help identify anomalies in log data by modelling normal behaviour and detecting deviations. Table 4 describes various unsupervised deep neural network structures for log anomaly detection.

2.2.4 Self-Supervised Multilayered Neural Network

Self-supervised methods for anomaly detection assume that large amounts of unlabeled data can be used to generate supervisory signals without manual labelling. These methods rely on pretext tasks, such as predicting parts of data or reconstructing input, to help the model learn valuable representations of the log data. The goal is for these learned representations to generalise well to tasks like anomaly detection, even without labelled data. However, the effectiveness of self-supervised methods depends on how well these pretext tasks are designed. Poorly designed tasks may lead to ineffective models. There is also a risk of overfitting to the pretext tasks, where the model performs well on these tasks but fails at anomaly detection. Evaluating the performance of these models can be challenging, especially when assessing how well the learned representations apply to anomaly detection. Computationally, self-supervised learning can be resource-intensive and involve complex model architectures, such as Transformers or generative models like VAEs and GANs. Finding the optimal configuration often requires extensive hyperparameter tuning. Popular self-supervised techniques include contrastive learning, generative models, Transformers, and multi-task learning frameworks, which help improve anomaly detection capabilities. The deep self-supervised architectures employed in anomaly detection are detailed in Table 5.

2.2.5 Hybrid Multilayered Neural Network

Hybrid deep neural network anomaly detection combines Supervised, unsupervised, and self-supervised learning techniques to improve performance. These models depend on the idea that different approaches can complement each other, allowing for more accurate and scalable anomaly detection, especially in large datasets. However, hybrid models are more com-

Table 2. Supervised deep neural approaches

References	Deep neural Approaches	Description in
[83], Onelog [45], [100]	CNN	Section 2.3.2
Lightlog [121]	TCNN	
Logspy [69]	CNN, AM	
Sentilog [145], Skdlog [41]	RNN	Section 2.3.3
Allcontext [105], LogNADS [80]	RNN,AM	
Logtransfer [12], Bertlog [13], Logsayer [152], [93], [116], [27], Conanomaly [84], LogLR [146]	LSTM	
Swisslog [70]	BI-LSTM	
LogRobust [149]	BI-LSTM, AM	
[103]	GRU	
[44], [78]	AE, AM	Section 2.3.4
[17]	CAE	
[56]	VAE	
Adanomaly [96]	BI-GAN	
Neurolog [66], Hitanomaly [52], Tranlog [35]	TF	Section 2.3.5
[21]	TF, AM	
LogGPT [42]	BERT	
LogGD [129]	GTNN	Section 2.3.6
CSCIlog [15, 23]	GNN	
[135]	GAN, GCN	
[18]	EGNN	

Table 3. Semi-supervised deep neural approaches

References	Deep neural Approaches	Description
[4], DeepAnt [87]	CNN	Section 2.3.2
ETCNLOG [10]	TCN	
Loganomaly [86]	RNN,AM	Section 2.3.3
Sialog [46]	RNN	
LTAnomaly [40],LogTAD [43],[112]	LSTM	
[91]	AE	Section 2.3.4
LogAttn [147], [88]	AE,AM	
LogBASA [76]	TF	Section 2.3.6
LogUAD [115]	TF,VAE	
Lanobert [68], Bert [19], Rapid [89]	BERT	
[110],[117]	BERT CL	
GenGlad [114], HiLog [54],[57], [74]	GNN	Section 2.3.7
[25]	MLP	Section 2.3.1

plex and require more computational resources during training and deployment. Selecting and combining the proper techniques can also be challenging. These systems handle large datasets efficiently using linear or nonlinear kernel models on reduced input dimensions. However, they may not optimise representation learning in hidden layers due to generic loss functions rather than tailored anomaly detection objectives [153]. The computational efficiency of hybrid anomaly detection models depends on the architecture, ensemble techniques, data size, algorithm choice, feature engineering, preprocessing, hyperparameter tuning, inference speed, memory requirements, and parallelisation. Empirical evaluations of specific hardware and software configurations are crucial for understanding their computational demands. Integrating CNNs and LSTMs in hybrid architectures balances computational complexities, addressing weaknesses of individual methods while offering adaptability to diverse datasets. Drawbacks include complexity in model design and training. Table 6 outlines various deep hybrid neural network architectures for log

Table 4. Unsupervised deep neural approaches

References	Deep neural Approaches	Description
[11],[107],SSDlog [82]	CNN	Section 2.3.2
Deeplog [20],[3] ,[139], Boostlog [106], MADDC [119]	LSTM	Section 2.3.3
NLSAlog [137], [125]	LSTM,AM	
LogST [148]	BI-LSTM	
[34], [74]	GRU	
ATT-GRU [128],PLELog [136]	GRU,AM	
Autolog [7],[101],[102],Logformer [38]	DAE	Section 2.3.4
[92], LogGAN [126]	CVAE	
[123], LSADNET [142]	TF	Section 2.3.6
[88]	TF,AM	
LogLG [37], GLAD-PAW [113], [133]	GNN	
DeepTralog [143]	GGNN	

Table 5. Self Supervised deep neural network approaches

References	Deep neural Approaches	Description
Log Pal [104], LogAttention [21], Logbert [36]	TF,AM	Section 2.3.6
LogFit [1]	BERT	
LogLG [37]	GNN	Section 2.3.7
LogEncoder [94]	AE,AM,CL	Section 2.3.6
LogSD [130]		

anomaly detection.

2.3 Multilayered neural network architectures for locating anomalies

This section outlines the features and objectives of multilayered neural network models employed in the studied articles.

2.3.1 Deep learning models

Several deep neural network models are utilised for anomaly detection in log data [134]. The Multi-Layer Perceptron (MLP) is a fundamental structure in these models, featuring fully connected layers where each node connects to every node in adjacent layers with weighted connections. MLPs excel in tasks like regression, classification, and clustering due to their simplicity.

However, they often exhibit lower classification accuracies than models explicitly designed for sequential data properties. Consequently, they are less frequently highlighted in literature and are typically used with other models or as supplementary attention methods [41], [3].In recent literature, ADLI Log by [4] presents a novel method for consistently and realistically detecting IT system abnormalities using AIOps. This system employs a two-phase learning technique that integrates log commands with intended system data, creating a multilayered neural network model that outperforms others.

2.3.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) excel in identifying detailed features from complex, high-dimensional data [44]. Combining convolutional and max pooling layers in Multi-Layer Perceptron (MLP) architectures enhances pattern detection and feature abstraction. [121] used a CNN model with max-pooling, logkey2vec embeddings, 1D convolutional layers, and a dropout layer for anomaly detection in large-scale system records. When paired with a lightweight temporal convolutional network (TCN), this approach effectively detects log abnormalities on resource-limited devices. [70] surpassed CNN in identifying irregularities in parallel systems by integrating natural language processing, clustering, and attention mechanisms to extract log templates in distributed systems. The ETCN-Log model by [10] combines efficient channel attention and temporal convolutional networks to understand longer sequence logs by merging semantic and temporal information, enhancing the capability of neural networks in detecting anomalies in log data.

2.3.3 Recurrent Neural Network

RNNs are extensively used, with 42 out of 124 analysed approaches employing them to identify abnormal-

Table 6. Hybrid deep neural network approaches

References	Deep neural Approaches	Description
[127], [138], [23], [133]	CNN,LSTM	
Fuzzy CNN [31]	CNN,AE	
[95]	CNN,SPN	
[51]	LSTM,AE	
[26]	BI-LSTM,GRU	
[6]	LSTM,VAE	Section 2.3
[112]	CAE,VAE	
Trine [150], [122]	TF,GAN	
[55]	BILSTM,BERT	
Glad [72]	GNN,TF	
[48]	GNN,GAN	

ities [105], [5], and [96]. RNNs capture patterns of sequential occurrences over time but struggle with long-term dependencies, leading to LSTM and GRU systems [4]. SentiLog [145] The first supervised method for identifying parallel file system issues using RNN-based emotional natural language models. . SKD-Log [41] use recurrent neural networks and attention strategies to distil self-knowledge for unpredictable log data. AllContext [105] utilises attention-RNNs to create region-specific and meaningful vectors from log events. Siamese Networks [46] use RNNs to detect log abnormalities.OC4Seq [120] Identifies discrete event anomalies using a multi-scale, one-class RNN. DeepLog [20] Employs LSTM to locate faults in logs and discover repeating log patterns. LogTransfer [12] Uses transfer learning, Glove log template encoding, and fully connected networks—BertLog [13] Classifies large-scale system abnormalities using bidirectional encoder representation transformers. Log Time Assessment [3] uses LSTM to discover and fix information system performance issues. LogC [139] Performs component-aware analysis to find log message-related abnormalities. LogNL [153] Utilizes NLP and LSTM to detect cloud platform log abnormalities. LogSayer [152] Uses LSTM networks for accurate cloud log abnormality detection. [40, 42] : Combines transformer and LSTM to find log abnormalities focusing on feature information. BoostLog [106] Uses LSTM classifiers for complex 5G network diagnostics.

LSTM-LRP [93]. Addresses log file anomaly detection interpretability. LogTAD [43] Uses LSTM and adversarial domain adaptation to identify system abnormalities. Cloud Log Message Analysis [27] use LSTM for cloud log anomaly detection. NLSALog, [137] Detects ITS security anomalies using layered LSTM and self-attention. Attention-based LSTM [125] for intelligent catastrophe recovery. ConAnomaly [84] uses a multi-layer LSTM and log2vec encoder.LogLR [146] Constructs logical connections between log sequences with LSTM.Unsupervised Deep Learning [111] for real-time network insider threat detection. SwissLog [70, 71] Finds abnormalities in dynamic software system logs using deep learning. LogRobust [149] Detects software-intensive system abnormalities using bi-LSTM and attention. Bi-SSGRU-Ga-Attention [34] Quickly and accurately finds anomalous data. Sentiment Analysis and GRU [103] Detects unusual OS log behaviours—attention-based Neural Network [128] Discovers log patterns during regular operation. PLELog [136] Combines supervised and unsupervised learning using GRU and attention. These techniques highlight the diverse applications and improvements in neural networks for log anomaly detection. [74] HiparaLog is a novel log anomaly detection method that converts raw log messages into semantic vectors, combining parameter and template features. It uses a self-attention-enhanced GRU model to capture global dependencies and contextual information, enabling multi-dimensional anomaly detection.

2.3.4 Autoencoders

Neural methods, particularly autoencoders and their variants, have shown significant promise in anomaly detection. Autoencoders, which use an encoder-decoder architecture to learn efficient codings of input data, detect anomalies based on reconstruction errors. Studies such as those by [91] have demonstrated their effectiveness in detecting anomalous log events and reducing false negatives. Semi-supervised approaches, as implemented by [7] , and deep autoencoders for forensic anomaly detection, as employed by [101], further illustrate their adaptability and effectiveness. Variational Autoencoders (VAEs) add a probabilistic element, enhancing robustness, as [141] shows in parsing and classifying log files with minimal human intervention. Attention-based models like LogAttn by [147] improve anomaly detection by focusing on important input data parts. Hybrid models, such

as those combining autoencoders with ant colony optimisation [17] or support vector machines [56], capture temporal connections and enhance detection capabilities.

Compared to traditional methods like PCA, autoencoders have shown superior performance in maintaining up-to-date system logs and detecting anomalies without extensive preprocessing, as evidenced by [92]. These findings underscore the flexibility and efficacy of autoencoder architectures in various anomaly detection applications, making them a feasible and promising solution.

2.3.5 Generative Adversarial Networks

Generative Adversarial Networks (GANs) provide a unique and promising approach to anomaly detection within the context of neural network techniques. GANs, composed of a generator and a discriminator in a competitive setup, enable unsupervised deep learning. [135] highlights their utility, though relatively few researchers have employed GANs specifically for anomaly detection. One notable example is Adanomaly by [96], which uses a Bidirectional GAN (BiGAN) model for feature extraction combined with an ensemble technique for detecting system anomalies and addressing class imbalance. This approach has been shown to improve both memory efficiency and accuracy. Another significant contribution is LogGAN by [126], an LSTM-based GAN designed for system log anomaly detection. LogGAN effectively handles out-of-order log issues through permutation event modelling. By bridging the gap between normal and abnormal instances, GANs enhance the detection of anomalies in complex datasets.

These studies support the feasibility of using GANs for anomaly detection, demonstrating their ability to improve accuracy and handle complex log data issues. GANs offer a flexible and powerful tool for identifying anomalies in various contexts, making them a valuable addition to the repertoire of neural network techniques for anomaly detection.

2.3.6 Transformer

Self-attention approaches introduced by transformers have significantly advanced deep learning, particularly in anomaly detection. These methods bring similar vector space components closer together, enhancing the performance of models like RNNs and transformers with attention mechanisms [135]. Several transformer-based models have been developed for log anomaly detection, demonstrating notable feasibility and effectiveness in this domain. Models like Neural Log, introduced by [66], are transformer-based classification models designed explicitly for log anomaly detection. Similarly, HitAnomaly by [52] uses a hierarchical transformer architecture to store log template sequences and parameter values, improving anomaly detection accuracy.

[35] proposed TRANSLOG, a unified transformer-based approach that excels in anomaly detection with fewer parameters and lower training costs. LogBASA, developed by [76], employs a self-attention encoder-decoder transformer model and system log knowledge graphs to identify anomalies in an unsupervised manner. LSAD-NET by [147] utilises a multi-layer convolutional method and a globally sparse transformer model for unsupervised log data anomaly detection.

UMFLog, created by [47], combines BERT for semantic characteristics and VAE for statistical aspects to detect log abnormalities without supervision. LogPal, developed by [104], classifies anomalies in large, heterogeneous log databases.

BERT has been effectively used for accurate system log analysis without templates, as demonstrated by [68], [1] LogFiT leverages a BERT-based language model for self-supervised training to fix log discrepancies. Pre-trained language models also reduce training time for RAPID, developed by [89], enabling real-time log data anomaly detection.

CLDTLog by [110] uses BERT models and contrastive learning to detect dual-objective log anomalies without log parsing. [118] (Wang X,2022) improved log anomaly detection using contrastive learning and a multi-scale MASS model.

The feasibility of transformer-based neural methods in anomaly detection is well-supported by these studies, demonstrating their ability to handle complex log data, reduce training costs, and improve detection accuracy. These models offer powerful tools for identifying anomalies in various log data contexts, making them highly viable for practical applications.

2.3.7 Graph Neural Network

Graph-based deep neural networks have been demonstrated to be highly effective for log anomaly detection, leveraging their capability to model complex relationships in log data. Graph Neural Networks (GNNs) have gained power due to their ability to represent log-event relations as graph structures. [37] proved the practicability of GNNs by building log-event graphs to spot irregularities in unlabeled data, using event dependencies and contextual patterns that conventional methods may overlook. Another technique, EdgeTorrent by [73], considers attribution graphs to analyse event flows, providing a robust way to find anomalies. [74] Logs2Graphs is an unsuper-

vised method for log anomaly detection that converts event logs into attributed, directed, and weighted graphs using the OCDiGCN model for effective graph-level anomaly detection. [133] The IST-GCN model integrates temporal and spatial perspectives using graph neural networks to enhance anomaly detection in system logs. It improves interpretability and performance, surpassing existing methods by increasing Average Precision and ROC AUC, demonstrating its effectiveness across multiple datasets. These developments in GNNs and associated neural methods show the flexibility and usefulness of such models in handling complex log events, optimising detection accuracy, and reducing training costs. The flexibility of neural methods confirms their significant role in advancing log anomaly detection across diverse domains. [114] GenGLAD is a graph-based framework for detecting log anomalies, designed to capture complex log associations more effectively than traditional methods. It generates graphs representing log data, using random walk and word2vec for node embeddings and employing clustering for unsupervised anomaly detection.

2.3.8 Hybrid Models

Hybrid neural network methods have demonstrated significant feasibility in improving log anomaly detection across various applications. [127] proposes a hybrid CNN-LSTM model for detecting data centre abnormalities, while [31] enhance cybersecurity using a convolutional autoencoder with fuzzy clustering. SpikeLog [95] combines recurrent neural networks and spiking neurons to achieve high accuracy and interpretability in poorly supervised settings. [108] introduced CausalConvLSTM, utilising CNNs and LSTMs to reduce industrial system defects. [140] detects insider attacks with an LSTM-CNN model, and [33] uses a hybrid LSTM Neural Network and Autoencoder for anomaly detection. [51] employ an LSTM autoencoder for smart device development, while [26] utilise auto-B/LSTM and auto-GRU models for log message anomaly detection.

[112] discovered unsupervised anomalies using a hybrid CAE-VAE model. [6] analysed computer system logs with an autoencoder model incorporating LSTM units. [150] uses a generative adversarial network (SeqGAN) to generate log messages and detect abnormalities. [55] suggest identifying HTTP request threats without log processing using BERT and BiLSTM models. [113] finds log sequence anomalies using contrastive adversarial training and dual feature extraction with BERT and VAE. [73] describes GLAD, a graph neural network (GNN) and transformer approach for log data anomaly detection. [50] combine GANs and log graph representation in AdvGraLog to discover log data abnormalities. [53] introduces deep ensemble models combining RNNs, LSTMs, CNNs, Transformers, and GNNs for improved anomaly detection in high-dimensional time series data. [122] proposes a fusion model integrating Isolation Forest, GAN, and Transformer for improved network anomaly detection and log analysis. Isolation Forest identifies anomalies, GAN generates synthetic data for training augmentation, and Transformer extracts time-series context, resulting in more accurate and robust detection with lower false alarms. Experimental results demonstrate enhanced anomaly detection and log analysis performance, improving system stability and network security. [133] This paper proposes a multi-source data anomaly detection method incorporating temporal and spatial characteristics. It uses a Transformer encoder for parsing log templates, an attention-based CNN for spatial features, and a Bi-LSTM network for capturing temporal features.

2.4 Analysis and Comparison Between Different Neural Network Approaches

Table 7 provides a clear overview of the strengths and weaknesses of each neural network approach in the context of log anomaly detection. Each approach has specific advantages that make it suitable for different scenarios, depending on dataset characteristics, anomaly types, and available computational resources. Analyse and compare different Neural Network approaches, highlighting their strengths and weaknesses in the context of anomaly detection.

3 Evaluation

This section summarises evaluations in research publications by integrating publicly available datasets and discussing frequently used evaluation criteria and benchmark procedures for comparing with the suggested methodology.

3.1 Data sets

Multilayer neural networks evaluate log data anomaly detection methods using LogHub [48] datasets. Key datasets include BGL, HDFS, Thunderbird, and OpenStack. HDFS is notable for its affordability and fault tolerance, requiring over 200 Amazon EC2 nodes for map-reduction. Annotated Hadoop logs provide a solid foundation for evaluation [132], [77] highlights system issues in a five-machine Hadoop cluster. CUHK labs enhance Apache server logs with Spark and ZooKeeper record OpenStack logs, including timeouts and failed injection errors [45]. The BGL dataset has four million log events from the BlueGene/L supercomputer over 200 days, with reliable anomaly labels [90]. Thunderbird and Spirit datasets

Table 7. Comparison Between Different Neural Network Approaches

Neural Network Approach	Strengths	Weaknesses	Application (References)
Multi-Layer Perceptron (MLP)	Simplicity and ease of implementation. Versatility in handling various machine learning tasks.	Lower accuracy compared to models designed for sequential data. Limited temporal understanding without additional mechanisms.	[134], [3], [41]
Convolutional Neural Networks (CNN)	Exceptional at capturing spatial dependencies. Effective feature extraction from log sequences.	Limited ability to capture long-term temporal dependencies in sequential data.	[44], [134], [121], [70], [10]
Recurrent Neural Networks (RNN)	Effective at capturing temporal dependencies. Stateful memory for retaining sequence context.	Prone to vanishing/exploding gradients. Complexity in handling large-scale data effectively.	[105], [5], [96], [41], [20], [12], [13], [139], [153], [152], [46], [106], [93], [27], [137], [125], [84], [148], [111], [70, 71], [34], [103], [128], [136], [43]
Autoencoders	Anomaly detection based on reconstruction errors. Unsupervised learning capability.	Complexity in architecture and hyperparameter tuning. Interpretability of anomalies based solely on reconstruction errors.	[91, 92], [7], [101], [17], [56]
Generative Adversarial Networks (GANs)	Unsupervised learning with feature extraction. Effective in handling complex log data issues.	Training instability (e.g., mode collapse). High computational requirements.	[135], [96], [126]
Transformer	Efficient at capturing long-range dependencies. Scalable to large datasets.	High computational cost, especially for large models. Interpretability challenges.	[135], [66], [52], [35], [38], [76], [147], [47], [104], [89], [68], [1], [110], [119], [72]
Hybrid Models	Combines strengths of different models for improved accuracy. Adaptable to diverse data types.	Increased model complexity and tuning requirements. Integration challenges.	[127], [31], [95], [108], [140], [33], [51], [26], [112], [6], [150], [55], [120], [47]

from Sandia National Labs provide extensive log data, with Spirit containing 172 million aberrant log messages [90]. The HPC dataset by [151] is recorded from Los Alamos National Laboratories' System 20 cluster. Windows 7 lab PC logs span 226.7 days and nearly 27 terabytes [29], and over 263.9 days, Linux logs were collected from /var/log/messages [119], [62].

The Dig Corpora log collection includes logs from various operating systems, network devices, web servers, and databases, capturing digital activities, system events, and security data for analysis [62], [102], [103] emphasise the need for secure log dataset evaluation methods. [30] developed AIT (v1.1) for evaluating anomaly-based intrusion detection systems.

A multilayered neural network-based log data anomaly detection system is evaluated using standard metrics. True positives (TP) are correctly predicted dangerous log entries, while false positives (FP) are routine entries misidentified as aberrant. True negatives (TN) are correctly predicted regular entries and false negatives (FN) are malicious entries that are misclassified as usual. These metrics assess the system's performance and reliability.

Precision measures the proportion of predicted abnormalities that are actual, calculated as:

$$P = \frac{TP}{TP + FP}$$

Recall, sensitivity, or actual positive rate is the proportion of anomalies the model appropriately recognised. A high recall indicates that the model can recognise most data problems. F1-Score balances recall and precision by being the harmonic mean of the two. Its utility is especially prominent in situations involving unbalanced datasets:

$$F1 = \frac{2 \times P \times R}{P + R}$$

The AUC-ROC curve (Area Under the Receiver Operating Characteristic Curve) measures the model's capacity to differentiate between normal and anomalous cases at different probability thresholds. A higher AUC-ROC indicates superior ability to distinguish between regular and abnormal cases.

Recall is given as:

$$R = \frac{TP}{TP + FN}$$

Precision-Recall Curve Area Under the Curve (AUC-PR) finds the area under the precision-recall curve. PR works well when most of the cases are normal. The false positive rate (FPR) quantifies the portion of typical occurrences indirectly identified as anomalies.

Accuracy measures the proportion of accurately predicted cases out of the total number of examples in the dataset:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

Numerous studies use Precision, Recall, and F1 scores to evaluate anomaly detection algorithms [4], [121], [46], [103], [76] , [104], The balance between Precision, Recall, and F1 scores is important. [10], [68] emphasise the F1 score, a weighted average of Precision and Recall, when determining the model's threshold value. AOC-RUC and AUC-PR curves, independent of the threshold value, are widely used in method assessments using receiver operator characteristic (ROC) [51] and precision-recall curves.

Accuracy, False Positive Rate (FPR), and other unusual assessment metrics are considered. Researchers calculate average assessment measures for all classes [105] in multiclass classification. Complex neural structures should be assessed using model parameters and training/detection time. Some studies examine how training affects datasets and how well their algorithms adapt to log pattern changes: [46], [52], and [147]. Assessments include FLOPS, model size, and detection time [115]. Deep anomaly detection models for log data must be evaluated using metrics that meet the application's needs and grasp the practical ramifications of false positives and negatives. .

3.2 Benchmark models

Many papers compare metrics to benchmark methodologies to demonstrate improvements over current methods. DeepLog by [20], an LSTM-based neural network for log anomaly detection, is the most frequently used benchmark, cited in 45 124 articles. [131] introduced PCA for anomaly detection by reducing the counting matrix and identifying unusual patterns. [86] developed Log Anomaly using template2vec and log templates, while [149] created Log Robust for handling erroneous log events with semantic vectors.Log Cluster, based on SVM methods [98], groups log sequence vectors by similarity. [81] used invariant mining to find linear correlations in log events. LogBERT by [36] uses MLM and BERT for anomaly detection. Isolation Forest identifies anomalies by partitioning data into isolated points [79]. CNN-based methods [32] and logistic regression [103] also classify event sequences.HitAnomaly by [52] uses transformer-based learning and custom parsers, while OC4Seq by [121] (Wang,2021) employs a multiscale RNN for detecting abnormalities. [24] uses an autoencoder with SeqGAN for feature extraction and anomaly detection. Swisslog by [70] and LogSy by [88] leverage deep learning for log format adjustment and data merging. LogGAN by [126] uses an LSTM-based generative adversarial network for log anomaly detection.

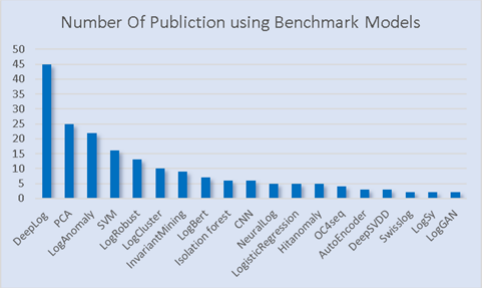

Figure 6. Publication using benchmark models for comparison

3.3 Discussion and Future Directions

In the following module, we will summarise these findings, address unresolved issues in detecting anomalies using multilayered neural structures, and suggest possible paths for further study as we address the research topics outlined in Section 1.

RQ1: How does log data anomaly detection performance vary across neural network structures?

The efficiency of log data anomaly detection is impacted by different neural network structures, which should be selected based on log data features, kinds of errors, availability of data labelling, and interpretability requirements. Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks, are often used to detect sequential patterns in log data to capture temporal linkages [3], [34]. Gated Recurrent Units (GRU) are a kind of recurrent neural network (RNN) that control the flow of information in a sequence of inputs [34]. Convolutional neural networks (CNNs) recognise event relationships proficiently, offering a feasible substitute for RNNs [32]. Autoencoders, such as Variational Autoencoders (VAEs), capture intricate patterns and nonlinear relationships in log data to represent many normal behaviours [56]. Generative Adversarial Networks (GANs) use adversarial training to generate typical log data and identify anomalies, while Convolution Autoencoders (CAEs) are proficient in capturing spatial patterns [17].

Table 8. Benchmark public data sets

Data set	Description	Time	Number of Messages	Data size	Labeled	Used in evaluation
HDFS [132]	Hadoop-distributed file system log	38.7 hours	11,175,629	1.47 GB	yes	[4] ,[83], [45], [87], [11], [114], [41], [105], [117], [12], [13], [139], [153], [151], [40] , [106], [93], [27], [137], [68], [69], [143], [65], [34], [136], [112], [6], [150] , [126], [66], [52], [76], [47], [36], [122], [37], [113], [129], [72], [88]
Hadoop [77]	Hadoop map-reduce job log	N.A.	394,308	48.61 MB	Yes	[45], [87], [41], [46], [12], [103], [7], [37], [15]
Spark [48]	Spark job log	N.A.	33,236,604	2.75 GB	No	[103]
Zookeeper [48]	ZooKeeper service log	26.7 days	74,380	9.95 MB	No	[103]
OpenStack [20]	OpenStack infrastructure log	N.A.	207,820	58.61 MB	Yes	[45], [82], [80], [13], [152], [43], [71], [51], [150], [96], [24], [52], [123], [55], [57]
BlueGen/L (BGL) [90]	Blue Gene/L supercomputer log	214.7 days	4,747,963	708.76 MB	Yes	[45],[87], [114], [10], [31], [105], [113], [80], [13], [40], [43], [27], [137], [125], , [70], [71], [34], [126], [66], [26], [7], [24], [123], [76], [47], [68], [1], [110], [37], [72], [88]
HPC RAS [151]	High-performance cluster log	N.A.	433,489	32.00 MB	No	[88]
Thunderbird [90]	Thunder bird supercomputer log	244 days	211,212,192	29.60 GB	Yes	[139], [27], [111], [24], [66], [52], [35], [76], [47], [68], [1], [37], [129], [72], [124], [88]
Spirit [90]	Spirit supercomputing system	558 days	272,298,969	30.289 GB	yes	[129], [72], [124], [88]
Digital corpora [29]	Windows logs	-	-	-	No	[102, 103]
DFRWS [22]	Linux logs	-	-	-	-	[103, 103]
Honeynet [2], [85]	OS event logs	-	-	-	-	[102, 103]
Linux [16]	Linux system log	263.9 days	25,567	2.25 MB	No	[102, 103]
Android [48]	Android log	N.A.	1,555,005	183.37 MB	No	[70, 71]

Utilising hybrid strategies or attention mechanisms, which merge many designs, may improve the model's performance. Temporal convolutional networks (TCNs) successfully manage sequences of varying lengths and temporal dependencies [10]. Transformers, well-known for their use in unsupervised learning, are becoming more popular. Graph Neural Networks (GNNs) are becoming more often used due to their ability to effectively model intricate relationships within graph-structured data [72]. Self-supervised learning techniques, such as contrastive learning, decrease reliance on annotated data. The research determines that log data features, anomalies, and model interpretability significantly influence the performance of deep neural networks. It is crucial to thoroughly evaluate and compare benchmark datasets with real-world log data to comprehend the capabilities and constraints of each architecture for detecting anomalies in log data.

RQ2: How do neural network models depend on labelled data, and what are their classifications: supervised, semi-supervised, unsupervised, self-supervised, and hybrid?

Neural network models significantly relate to labelled data, influencing their learning methods and uses. Supervised learning models rely on labelled datasets to learn patterns and correlations for tasks such as image classification and natural language processing. Semi-supervised learning combines labelled and unlabeled data, using labelled data for supervised learn-

ing with unlabeled data to improve overall data comprehension. Unsupervised learning functions without explicit output labels, recognising intrinsic data patterns for tasks like clustering. Self-supervised learning creates labels from incoming data via surrogate tasks, efficiently using large quantities of unlabeled data. Hybrid learning combines elements from several models, demonstrating flexibility in many learning situations. We found 40 supervised, 28 semi-supervised, 28 unsupervised, 15 hybrid, and six self-supervised methods for anomaly detection tasks in our analysis, showcasing the adaptability of neural network models in various scenarios.

RQ3: What are existing literature limits and recommendations that might guide future research?

During our systematic review, we examined how well current methods address the challenges outlined in Section X. Data instability emerged as a key issue, primarily tackled by representing logs as semantic vectors for comparison [7], [43], [66], [70], [84], [108], [125], [136], [149]. Techniques like numeric vector generation and context-aware embedding address the challenges of unstructured data and imbalanced datasets [24], [69], [105], [14], [126] . Some methods optimise lightweight algorithms by leveraging low-dimensional vectors and convolutional neural networks [11], [36], [121]. Adaptive learning and transfer learning offer solutions for dynamically adjusting models and utilising data from different domains [12], [35]. However, challenges persist, with many methods overlooking diverse anomaly artefacts and lacking explainability. Future efforts should focus on enhancing model interpretability for more effective system operation. Hybrid models that integrate deep learning and machine learning techniques have the potential to improve accuracy and resilience [72]. Optimising and parallel processing are essential for managing substantial log data streams. Unsupervised and self-supervised learning are crucial when limited labelled data is available. Specialised models tailored to specific fields, the ability to explain anomalies, and human involvement may enhance the system's dependability. These suggestions support the advancement of log-based anomaly detection through deep neural networks.

RQ4: How do model architectures and hyperparameters affect real-world system anomaly detection time delays?

Hyperparameter tuning is crucial in optimising the performance of machine learning models for real-world applications, particularly in anomaly detection systems. [70, 71] emphasises the significance of hyperparameter tuning for optimal performance in anomaly detection by introducing a unified attention-based BiLSTM model. [31] stresses the importance of efficient hyperparameter adjustments in CNNs to enhance algorithm efficiency, highlighting the quadratic relationship between method running time and data quantity. Additionally, [15] advocates optimising hyperparameters in anomaly detection, such as sliding window length and threshold value, to improve accuracy and reduce processing costs, showcasing the critical role of hyperparameter settings in enhancing algorithm performance and resource utilisation. Balancing model complexity, processing efficiency, and hyperparameters is essential to address latency issues and improve anomaly detection responsiveness in real-world systems.

RQ5: What is the Explainability and Availability of deep anomaly detection models?

Our review explores the fundamental considerations of explainability and availability in the practical adoption of deep anomaly detection models. The growing complexity of these models, mainly neural network-based ones, emphasises the need for explainability to generate confidence and comprehension. Researchers have used activation maximisation, attention mechanisms [105], Layer-wise Relevance Propagation (LRP), SHAP (Shapley Additive explanations), and model-specific methods like LSTMs to address this difficulty. Activation maximisation visualises neurons, while attention processes emphasise key input data to help understand model choices. SHAP values give important feature significance ratings for LRP pre-dictions.

Interpretability is fundamental in model topologies like LSTMs. Researchers and open-source contributors make pre-trained models and code implementations available on public repositories like GitHub. TensorFlow, PyTorch, and Keras are popular deep-learning frameworks and libraries. Blogs, tutorials, and courses help develop and refine deep anomaly detection models. Businesses selling deep learning-based anomaly detection technologies enhance the landscape and promote accessibility. Despite these advances, balancing complexity and interpretability remains difficult. This study is essential for improving and integrating deep anomaly detection algorithms into real-world applications. This review summarises the field's progress in explainability and availability and suggests further research and improvement.

4 Conclusion

This article reviews 124 multilayered neural network anomaly detection approaches using log data. Despite

the considerable progress made, challenges and opportunities persist in the field of log-based anomaly detection through the utilisation of multilayered neural networks. Subsequent studies should prioritise the enhancement of reproducibility through the use of standardised benchmarks and open datasets, the curation of diverse evaluation datasets, and the improvement of model explainability for practical implementation. It is imperative to employ methodologies that facilitate a better understanding of classification outcomes and to investigate innovative architectures, such as hybrid models. The development of real-time models that are continuously updated to adapt to emerging threats, along with the integration of domain expertise, can bolster reliability. Using cross-domain applications and transfer learning can mitigate the necessity for extensive labelled datasets. Collaboration between academia and industry and investment in interdisciplinary research will cultivate more efficient anomaly detection solutions, thus fortifying systems and ensuring their reliability in real-world scenarios.

Funding

The authors state that this work has not received any funding.

Acknowledgement

We thank all the persons in the department who helped to complete the research, especially the HOD and guide.

References

[1] C. Almodovar, F. Sabrina, S. Karimi, S. Azad (2024), Log Fit: Log Anomaly Detection using Fine Tuned Language Models. IEEE Transactions on Network and Service Management.DOI: 10.1109/TNSM.2024.3358730.

[2] G. Arcas, H. Gonzales, J. Cheng (2011), Challenge 7 of the Honeynet Project Forensic Challenge 2011-Forensic analysis of a compromised server. Retrieved August, 21, 2017.

[3] X. Baril, O. Coustié, J. Mothe, *et al.*, Application performance anomaly detection with LSTM on temporal irregularities in logs, Proc. 29th ACM Int. Conf. on Information & Knowledge Management, pp. 1961–1964, 2020, doi: 10.1145/3340531.341215.

[4] J. Bogatinovski, G. Madjarov, S. Nedelkoski, et al., Leveraging Log Instructions in Log-based Anomaly Detection, 2022 IEEE Int. Conf. on Services Computing (SCC), 2019, pp. 321–326. [Online]. Available: https://arxiv.org/pdf/2207.03206.pdf

[5] A. Brown, A. Tuor, B. Hutchinson, and N. Nichols, Recurrent neural network attention mechanisms for interpretable system log anomaly detection, in Proc. 1st Workshop on Machine Learning for Computing Systems, pp. 1–8, 2018. [Online]. Available: https://arxiv.org/pdf/1803.04967.pdf

[6] S. Bursic, V. Cuculo, and A. Amelio, Anomaly detection from log files using unsupervised deep learning, in Int. Symp. on Formal Methods, pp. 200–207, 2019, doi: 10.1007/978-3-030-54994-7-15.

[7] M. Catillo, A. Pecchia, and U. Villano, AutoLog: Anomaly detection by deep autoencoding of system logs, Expert Systems with Applications, 2022, doi: 10.1016/j.eswa.2021.116263.

[8] R. Chalapathy and S. Chawla, Deep learning for anomaly detection: A survey, arXiv preprint arXiv:1901.03407, 2019.

[9] V. Chandola, A. Banerjee, and V. Kumar, Anomaly detection: A survey, ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, 2009. [Online]. Available: https://conservancy.umn.edu/bitstream/handle/11299/215731/07-017.pdf?sequence=1

[10] Y. Chang, N. Luktarhan, J. Liu, and Q. Chen, ETCNLog: A System Log Anomaly Detection Method Based on Efficient Channel Attention and Temporal Convolutional Network, Electronics, vol. 12, no. 8, 2023, doi: 10.3390/electronics1208187.

[11] P. Cheansunan and P. Phunchongharn, Detecting anomalous events on distributed systems using convolutional neural networks, in 10th Int. Conf. on Awareness Science and Technology, pp. 1–5, 2019, doi: 10.1109/ICAwST.2019.8923357.

[12] R. Chen, S. Zhang, D. Li, et al., Log transfer: Cross-system log anomaly detection for software systems with transfer learning, in IEEE 31st Int. Symp. on Software Reliability Engineering, pp. 37–47, 2020, doi: 10.1109/ISSRE5003.2020.00013.

[13] S. Chen and H. Liao, Bert-log: Anomaly detection for system logs based on a pre-trained language model, Applied Artificial Intelligence, 2022, doi: 10.1080/08839514.2022.2145642.

[14] Z. Chen, J. Liu, W. Gu, Y. Su, and M. R. Lyu, Experience report: Deep learning-based system log analysis for anomaly detection, 2021, doi: 10.1145/1122445.1122456.

[15] L. Chen, C. Song, X. Wang, D. Fu, and F. Li, CSCLog: A Component Subsequence Correlation-Aware Log Anomaly Detection Method, arXiv preprint arXiv:2307.03359, 2023.

[16] A. Chuvakin, Public security log-sharing site, 2010. [Online]. Available: https://log-sharing.dreamhosters.

[17] Y. Cui, Y. Sun, J. Hu, et al., A convolutional

auto-encoder method for anomaly detection on system logs, in Proc. IEEE SMC, pp. 3057–3062, 2018, doi: 10.1109/SMC.2018.00519.

[18] L. Decker , D. Leite , F. Viola F, *et al.* (2020), Comparison of evolving granular classifiers applied to anomaly detection for predictive maintenance in computing centres. IEEE Conference on evolving and adaptive Intelligence, DOI: 10.1109/EAIS48028.2020.9122779.

[19] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv:1810.04805, 2018, doi: 10.48550/arXiv.1810.04805.

[20] M. Du, F. Li, G. Zheng, *et al.*, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning, in ACM SIGSAC Conf. on Computer and Communications Security, pp. 1285–1298, 2017, doi: 10.1145/3133956.3134015.

[21] Q. Du, L. Zhao, J. Xu, *et al.*, Log-based anomaly detection with a multi-head scaled dot-product attention mechanism, in Int. Conf. on Database and Expert Systems, 2021, doi: 10.1007/978-3-030-86472-9-31.

[22] C. Eoghan and G. R. I. Golden, URL: http://old.dfrws.org/2009/challenge/index.shtml, 2009.

[23] Y. Fang , Z. Zhao , Y. Xu , *et al.* (2023), Log Anomaly Detection Based on Hierarchical Graph Neural Network and Label Contrastive Coding. Computers, Materials & Continua 2023(2):74–74, 10.32604/cmc.2023.033124.

[24] A. Farzad, Log message anomaly detection with oversampling, Int. J. Artificial Intelligence and Applications (IJAIA), vol. 4, pp. 11–11, 2020, doi: 10.5121/ijaia.2020.11405.

[25] A. Farzad and T. Gulliver, A Log message anomaly detection with fuzzy C-means and MLP, Applied Intelligence, 2022, pp. 17708–17717, doi: 10.1007/s10489-022-03300-1.

[26] A. Farzad and T. Gulliver, Log message anomaly detection and classification using auto-B/LSTM and auto-GRU, arXiv preprint arXiv:1911.0874, 2019.

[27] A. Farzad and T. A. Gulliver, Two class pruned log message anomaly detection, SN Computer Science, vol. 2, no. 5, pp. 1–18, 2021, doi: 10.1007/s42979-021-00772-9.

[28] G. Li, J. Mo, G. Zhou, and C. Li, HiparaLog: Improving Log-based Anomaly Detection through Parameter Feature Integration, in 2024 Int. Joint Conf. on Neural Networks (IJCNN), Yokohama, Japan, pp. 1–8, 2024, doi: 10.1109/IJCNN60899.2024.10650376.

[29] S. Garfinkel, P. Farrell, V. Roussev, et al., Bringing science to digital forensics with standardised forensic corpora, Digital Investigation, vol. 6, pp. 2–11, 2009.

[30] I. Giurgiu and Crosby, URL: https://github.com/microservices-demo, 2017.

[31] O. Gorokhov, M. Petrovskiy, I. Mashechkin, et al., Fuzzy CNN Autoencoder for Unsupervised Anomaly Detection in Log Data, Mathematics, vol. 18, 3995, 2023, doi:10.3390/math11183995.

[32] O. Gorokhov, M. Petrovskiy, I. Mashechkin, Convolutional neural networks for unsupervised anomaly detection in text data, in Int. Conf. on Intelligent Data Engineering and Automated Learning, pp. 500–507, 2017, doi:10.1007/978-3-319-68935-7-54.

[33] A. Grover, Anomaly detection for application log data, 2018, doi:10.31979/etd.znsb-bw.

[34] S. Gu, Y. Chu, W. Zhang, et al., Research on system log anomaly detection combining two-way slice GRU and GA-attention mechanism, in 4th Int. Conf. on Artificial Intelligence and Big Data, pp. 577–583, 2021, doi:10.1109/ICAIBD51990.2021.9459087.

[35] H. Guo, X. Lin, J. Yang, et al., Translog: A unified transformer-based framework for log anomaly detection, CoRR, 2022, doi:10.48550/arXiv.2201.00016.

[36] H. Guo, S. Yuan, X. Wu, Logbert: Log anomaly detection via Bert, in Int. Joint Conf. on Neural Networks, pp. 1–8, 2021, doi:10.1109/IJCNN52387.2021.9534113.

[37] H. Guo , Y. Guo , J. Yang , *et al.* (2023), LogLG: Weakly Supervised Log Anomaly Detection via Log-Event Graph Construction. In: International Conference on Database Systems for Advanced Applications. Springer, pp 490–501, doi:10.1007/978-3-031-30678-5-36.

[38] H. Guo, J. Liu, W. Gu, Y. Su, M. R. Lyu, LogFormer: A Pre-train and Tuning Pipeline for Log Anomaly Detection, 2024, doi:10.48550/arXiv.2401.04749.

[39] Y. Guo , Y. Wu , Y. Zhu , *et al.* (2021), Anomaly detection using distributed log data: A lightweight federated learning approach. 2021 international joint conference on neural networks 2021, doi:10.1109/IJCNN52387.2021.9533294.

[40] D. Han, M. Sun, M. Li, *et al.*,(2023), LTAnomaly: A Transformer Variant for Syslog Anomaly Detection Based on Multi-Scale Representation and Long Sequence Capture, Applied Sciences, vol. 13, 2023, doi:10.3390/app13137668.

[41] N. Han, S. Lu, D. Wang, *et al.*, Skdlog: self-knowledge distillation-based CNN for abnormal log detection, in 19th IEEE

Int. Conf. on Ubiquitous Intelligence and Computing, 2022, doi:10.1109/SmartWorld-UIC-ATC-ScalComDigitalTwin-PriCompMetaverse56740.2022.00122.

[42] X. Han , S. Yuan , M. Trabelsi , *et al.* (2023), Log Anomaly Detection via GPT. 2023 IEEE International Conference on Big Data (BigData) pp 2023–2023, doi: 10.1109/BigData59044.2023.10386543 .

[43] X. Han , S. Yuan , LogTAD (2021), Unsupervised cross-system log anomaly detection via domain adaptation. Proceedings of the 30th ACM international conference on information & knowledge management pp 3068–3072, doi:10.1145/3459637.3482209.

[44] M. Hariharan, A. Mishra, S. Ravi, A. Sharma, A. Tanwar, K. Sundaresan, R. Karthik, (2023), Detecting log anomaly using subword attention encoder and probabilistic feature selection. Applied Intelligence, 1-16.

[45] S. Hashemi, M. Mäntylä, OneLog: Towards end-to-end training in software log anomaly detection, 2021, doi:10.48550/arXiv.2104.07324.

[46] S. Hashemi, M. Mäntylä, SiaLog: detecting anomalies in software execution logs using the siamese network, Automated Software Engineering, vol. 29, no. 2, pp. 61–61, 2022, doi:10.1007/s10515-022-00365-7.

[47] S. He, T. Deng , B. Chen, *et al.* (2023), Unsupervised Log Anomaly Detection Method Based on Multi-Feature. Computers, Materials & Continua 2023(1):76–76, https://doi.org/10.48550/arXiv.2008.06448.

[48] S. He, J. Zhu, P. He, M. R. Lyu, Loghub: An extensive collection of system log datasets towards automated log analytics, 2020, doi:10.48550/arXiv.2008.06448.

[49] S. He, J. Zhu, P. He, *et al.*, Experience report: System log analysis for anomaly detection, in 2016 IEEE 27th Int. Symp. on Software Reliability Engineering (ISSRE), pp. 207–218, 2016, doi:10.1109/ISSRE.2016.21.

[50] Z. He, Y. Tang, K. Zhao, J. Liu and W.c. Chen, Graph-Based Log Anomaly Detection via Adversarial Training. Dependable Software Engineering. Theories, Tools, and Applications. SETTA 2023, 14464 https://doi.org/10.1007/978-981-99-8664-4-4.

[51] R. Hirakawa, K. Tominaga, Y. Nakatoh, Software log anomaly detection through one transformer encoder representation clustering class, in Int. Conf. on Human-Computer Interaction, 2020.

[52] S. Huang, Y. Liu, C. Fung, *et al.*, Hitanomaly: Hierarchical transformers for anomaly detection in the system log, IEEE Trans. on Network and Service Management, vol. 17, no. 4, pp. 2064–2076, 2020, doi:10.1109/TNSM.2020.3034647.

[53] A. Iqbal, R. Amin, F. S. Alsubaei, A. Alzahrani, Anomaly detection in multivariate time series data using deep ensemble models, PLoS ONE, vol. 19, no. 6, e0303890, 2024, doi:10.1371/journal.pone.0303890.

[54] T. Jia, Y. Li, Y. Yang, *et al.*, Augmenting Log-based Anomaly Detection Models to Reduce False Anomalies with Human Feedback, in 28th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, pp. 3081–3089, 2022, doi:10.1145/3534678.3539106.

[55] L. S. Ramos Júnior, D. Macêdo, A. L. Oliveira, C. Zanchettin, (2022), Detecting Malicious HTTP Requests Without Log Parser Using RequestBERT-BiLSTM. In Brazilian Conference on Intelligent Systems (pp. 328-342). Cham: Springer International Publishing.Kan D, Fang X (2023) ,doi:10.1007/s00530-023-01199-3.

[56] Y. Kawachi, Y. Koizumi, N. Harada, Complementary set variational autoencoder for supervised anomaly detection, in 2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 2366–2370, 2018, doi:10.1109/ICASSP.2018.8462181.

[57] S. Kong, J. Ai, M. Lu, *et al.*, GRAND: GAN-based software runtime anomaly detection method using trace information, Neural Networks, vol. 169, pp. 365–377, 2024, doi:10.1016/j.neunet.2023.10.036.

[58] L. G. Korzeniowski, K. Goczyla, Landscape of automated log analysis: a systematic literature review and mapping study, IEEE Access, vol. 10, pp. 21892–21913, 2022, doi:10.1109/ACCESS.2022.3152549.

[59] C. Kruegel, G. Vigna, Anomaly detection of web-based attacks, in 10th ACM Conf. on Computer and Communications Security, pp. 251–261, 2003, doi:10.1145/948109.948144.

[60] D. Kwon, H. Kim, J.Kim, S. C. Suh, I. Kim, , and K. J. Kim, A survey of deep learning-based network anomaly detection, Cluster Computing, vol. 22, no. 1, pp. 949–961, 2019. doi: 10.1145/948109.948144

[61] M. Landauer , S. Onder , F. Skopik , *et al.*, Deep learning for anomaly detection in log data: A survey, Machine Learning with Applications, 2023. doi: 10.1016/j.mlwa.2023.100470

[62] M. Landauer , F. Skopik , M. Wurzenberger , *et al.*, Have it your way: Generating customised log datasets with a model-driven simulation testbed, IEEE Transactions on Reliability, 2021. doi: 10.1109/TR.2020.303131

[63] M. Landauer , M.Wurzenberger , F. Skopik ,

et al., Dynamic log file analysis: An unsupervised cluster evolution approach for anomaly detection, Computers & Security, vol. 79, pp. 94–116, 2018. doi: 10.1016/j.cose.2018.08.009

[64] M. Landauer , F.Skopik , M.Wurzenberger , *et al.*, System log clustering approaches for cyber security applications: A survey, Computers & Security, 2020, pp. 92–92. doi: 10.1016/j.cose.2020.101739

[65] VH. Le , H. Zhang , Log-based anomaly detection with deep learning: How far are we? Proceedings of the 44th International Conference on Software Engineering, pp. 1356–1367, 2022. doi: 10.1145/3510003.3510155

[66] VH. Le , H. Zhang , Neural log. Log-based anomaly detection without log parsing, 2021 36th IEEE/ACM International Conference on Automated Software Engineering, pp. 492–504. doi: 10.1109/ASE51524.2021.9678773

[67] Y. Lecun , Y. Bengio , G. Hinton , Deep learning, Nature, vol. 521, no. 7553, pp. 436–444, 2015. doi: 10.1038/nature14539

[68] Y. Lee, J. Kim, P. Kang, P., and Lanobert, 2023, System log anomaly detection based on the Bert masked language model, Applied Soft Computing, vol. 146, pp. 110689. doi: 10.1016/j.asoc.2023.110689

[69] H. Li, and Y. Li, (2020), Logspy: System log anomaly detection for distributed systems, International Conference on Artificial Intelligence and Computer Engineering, pages 347–352, doi: 10.1109/ICAICE51518.2020.00073.

[70] X. Li , P. Chen , L. Jing , *et al.* (2020), Swisslog: Robust and unified deep learning-based log anomaly detection for diverse faults, IEEE 31st International Symposium on Software Reliability Engineering (ISSRE) pp 92–103, doi: 10.1109/ISSRE5003.2020.00018.

[71] X. Li , P. Chen , L. Jing , *et al.*, Swiss Log: Robust anomaly detection and localisation for interleaved unstructured logs," IEEE Transactions on Dependable and Secure Computing, 2022, doi: 10.1109/TDSC.2022.3162857

[72] Y. Li, Y. Liu, H. Wang, Z.Chen, W. Cheng, Y. Chen, ... & C. Liu, Glad: Content-aware dynamic graphs for log anomaly detection. In 2023 IEEE International Conference on Knowledge Graph (ICKG) (pp. 9-18). IEEE, doi: 10.1109/ICKG59574.2023.0.

[73] Z. Li, J. Shi, M. van Leeuwen, (2023), Graph Neural Network-based Log Anomaly Detection and Explanation, doi:10.48550/arXiv.2307.00527.

[74] Li, Zhong, J. Shi, and M. Van Leeuwen, Graph Neural Networks based Log Anomaly Detection and Explanation, Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings, 2024.

[75] HJ. Liao , CHR. Lin , YC. Lin , *et al.*, Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, 2013. doi: 10.1016/j.jnca.2012.09.00.

[76] L. Liao , K. Zhu , J. Luo , *et al.*, Log Anomaly Detection Based on System Behavior Analysis and Global Semantic Awareness, International Journal of Intelligent Systems, 2023. doi: 10.1155/2023/3777826.

[77] Q. Lin , H. Zhang , JG. Lou , *et al.*, Log clustering-based problem identification for online service systems, Proceedings of the 38th International Conference on Software Engineering Companion, pp. 102–111, 2016. doi: 10.1145/2889160.2889232.

[78] C. Liu, M. Liang, J. Hou, J. Gu, Z.Wang, (2022), LogCAD: An Efficient and Robust Model for Log-Based Conformal Anomaly Detection. Security and Communication Networks, 2022, doi:10.1155/2022/5822124.

[79] FT. Liu , KM. Ting , ZH.Zhou , Isolation forest, Eighth IEEE International Conference on Data Mining, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17

[80] X. Liu , W. Liu , X. Di , *et al.* (2021), LogNADS: Network anomaly detection scheme based on log semantics representation Future Generation Computer Systems 124:390–405, doi:10.1016/j.future.2021.05.024.

[81] J. G. Lou, Q. Fu, S.Yang, Y. Xu, & J. Li, Mining invariants from console logs for system problem detection, In 2010 USENIX Annual Technical Conference (USENIX ATC 10), 2010. [Online]. Available: https://www.usenix.org/legacy/event/atc10/tech/full_papers/Lou.pdf

[82] S. Lu, N. Han, M. Wang, X. Wei, Z. Lin, and D. Wang, SSDLog: a semi-supervised dual branch model for log anomaly detection, World Wide Web, pp. 1–17, 2023. doi: 10.1007/s11280-023-01174-y.

[83] S. Lu, X. Wei, Y. Li, and L. Wang, Detecting anomalies in big data system logs using convolutional neural network. 2018 IEEE 16th Intl Conf on dependable, autonomic and secure computing, 16th Intl Conf on pervasive intelligence and computing, 4th Intl Conf on considerable data intelligence and computing and Cyber Science and Technology Congress, pages 151–158, doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.000.

[84] D. Lv , N. Luktarhan , Y. Chen , ConAnomaly: Content-based anomaly detection for system

logs, Sensors, vol. 21, no. 18, pp. 6125, 2021. doi: 10.3390/s2118612

[85] R. Marty, A. Chuvakin, & S. Tricaud, The Honeynet Project 2010 challenge 5 – log mysteries, 2022. [Online]. Available: https://www.honeynet.org/challenges/forensic-challenge-7-analysisof-a-compromised-server/

[86] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, and Y. Liu, Unsupervised detection of sequential and quantitative anomalies in unstructured logs, IJCAI, pp. 4739–4745, 2019. [Online]. Available: https://nkcs.iops.ai/wp-content/uploads/2019/06/paper-IJCAI19-LogAnomaly.pdf

[87] M. Munir, S. A. Siddiqui, A. Dengel, & S. Ahmed, DeepAnT: A deep learning approach for unsupervised anomaly detection in time series, IEEE Access, vol. 7, pp. 1991–2005, 2018. doi: 10.1109/ACCESS.2018.2886457

[88] S. Nedelkoski, J. Bogatinovski, A. Acker, J. Cardoso, O. Kao, and Logsy, Self-attentive classification-based anomaly detection in unstructured logs, 2020 IEEE International Conference on Data Mining, pp. 1196–1201. doi: 10.1109/ICDM50108.2020.001

[89] G. No, Y. Lee, H. Kang, & P. Kang, RAPID: Training-free Retrieval-based Log Anomaly Detection with PLM considering Token-level information, 2023. doi: 10.48550/arXiv.2311.0516

[90] A. Oliner and J. Stearley, What supercomputers say: A study of five system logs, in 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2007, pp. 575–584, doi:10.1109/DSN.2007.103.

[91] K. Otomo, S. Kobayashi, K. Fukuda, and H. Esaki, Finding anomalies in network system logs with latent variables, Proc. Workshop Big Data Anal, 2018, pp. 8–14, doi:10.1145/3229607.3229608.

[92] K. Otomo, S. Kobayashi, K. Fukuda, and H. Esaki, Latent variable-based anomaly detection in network system logs, IEICE Transactions on Information and Systems, vol. 102, no. 9, pp. 1644–1652, 2019, doi:10.1587/transinf.2018OFP0007.

[93] A. Patil, A. Wadekar, T. Gupta, R. Vijan, and F. Kazi, Explainable LSTM model for anomaly detection in HDFS log file using layerwise relevance propagation, IEEE Bombay Section Signature Conference, 2019, pp. 1–6, doi:10.1109/IBSSC47189.2019.8973044.

[94] J. Qi, Z. Luan, S. Huang, C. Fung, H. Yang, and D. Qian, LogEncoder: Log-based Contrastive Representation Learning for anomaly detection, IEEE Transactions on Network and Service Management, 2023, doi:10.1109/TNSM.2023.3239522.

[95] J. Qi, Z. Luan, S. Huang, C. Fung, H. Yang, and D. Qian, SpikeLog: Log-based anomaly detection via Potential-assisted Spiking Neuron Network, IEEE Transactions on Knowledge and Data Engineering, 2023, doi:10.1109/TKDE.2023.3347695.

[96] J. Qi, Z. Luan, S. Huang, Y. Wang, C. Fung, H. Yang, and D. Qian, Ad anomaly: adaptive anomaly detection for system logs with adversarial learning, NOMS 2022- 2022 IEEE/IFIP Network Operations and Management Symposium, 2022, pp. 1–5, doi:10.1109/NOMS54207.2022.9789917.

[97] R. Xu and Y. Li, Interpretable Spatial–Temporal Graph Convolutional Network for System Log Anomaly Detection, Advanced Engineering Informatics, vol. 62, Part C, 102803, 2024, doi:10.1016/j.aei.2024.102803.

[98] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, *et al.*, Estimating the support of a high-dimensional distribution, Neural Computation, vol. 13, no. 7, pp. 1443–1471, 2001, doi:10.1162/089976601750264965.

[99] E. Serkani, H. G. Garakani, and N. Mohammadzadeh, Anomaly detection uses SVM as a classifier and decision tree to optimise feature vectors, The ISC International Journal of Information Security, 2019, doi:10.22042/isecure.2019.164980.448.

[100] R. Sinha, R. Sur, and R. Sharma, Anomaly Detection Using System Logs: A Deep Learning Approach, International Journal of Information Security and Privacy (IJISP), vol. 2022, no. 1, doi:10.4018/IJISP.285584.

[101] H. Song, Z. Jiang, A. Men, and B. Yang, A hybrid semi-supervised anomaly detection model for high-dimensional data, Computational Intelligence and Neuroscience, 2017, doi:10.1155/2017/8501683.

[102] H. Studiawan and F. Sohel, Anomaly detection in a forensic timeline with deep autoencoders, Journal of Information Security and Applications, vol. 63, 103002, 2021, doi:10.1016/j.jisa.2021.103002.

[103] H. Studiawan, F. Sohel, C. Payne, *et al.*, Anomaly detection in operating system logs with deep learning-based sentiment analysis, IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 5, 2020, doi:10.1109/TDSC.2020.3037903.

[104] L. Sun and X. Xu, LogPal: A Generic Anomaly Detection Scheme of Heterogeneous Logs for Network Systems, Security and Communication Networks, 2023, doi:10.1155/2023/2803139.

[105] P. Sun, E. Yuepeng, T. Li, *et al.*, Context-aware learning for anomaly detection with imbalanced log data, 2020 IEEE 22nd International Conference on High Performance Computing and Communications, 2020, pp. 449–456, doi:10.1109/HPCC-SmartCity-DSS50907.2020.00055.

[106] T. Sundqvist, M. H. Bhuyan, J. Forsman, *et al.*, Boosted ensemble learning for anomaly detection in 5G RAN, IFIP International Conference on Artificial Intelligence Applications and Innovations, 2020, pp. 15–30, doi:10.1007/978-3-030-49161-1-2.

[107] T. Sutthipanyo, T. Lamsan, W. Thawornsusin, *et al.*, Log-Based Anomaly Detection Using CNN Model with Parameter Entity Labeling for Improving Log Preprocessing Approach, TENCON 2023, pp. 914–919, doi:10.1109/TENCON58879.2023.10322478.

[108] S. Syngal, S. Verma, K. Karthik, *et al.*, Server-language processing: A semi-supervised approach to server failure detection, 2021 2nd International Conference on Computing, Networks and the Internet of Things, pp. 1–7, doi:10.1109/TKDE.2023.3347695.

[109] D. M. Tax and R. P. Duin, Support vector data description, Machine Learning, vol. 54, pp. 45–66, 2004, doi:10.1023/B:MACH.0000008084.60811.49.

[110] G. Tian, N. Luktarhan, H. Wu, *et al.*, CLDT-Log: System Log Anomaly Detection Method Based on Contrastive Learning and Dual Objective Tasks, Sensors, vol. 23, no. 11, 5042, 2023, doi:10.3390/s23115042.

[111] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, S. Robinson, Y. Xie, H. Zhang, and M. A. Babar, LogGD: Detecting Anomalies from System Logs with Graph Neural Networks, in 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS), pp. 299–310, 2022, doi:10.1109/QRS57517.2022.00039.

[112] A. Wadekar, T. Gupta, R. Vijan, *et al.*, Hybrid CAE-VAE for unsupervised anomaly detection in log file systems, 2019 10th International Conference on Computing, Communication, and Networking Technologies, pp. 1–7, 2019, doi:10.1109/ICCCNT45670.2019.8944863.

[113] Y. Wan, Y. Liu, D. Wang, *et al.*, GLAD-PAW: Graph-based log anomaly detection by position-aware weighted graph attention network, Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 66–77, 2021, doi:10.1007/978-3-030-75762-5-6.

[114] H. Wang , Y. Chen , C. Zhang , *et al.* (2022), GenGLAD: A Generated Graph-Based Log Anomaly Detection Framework. In: International Conference on Smart Computing and Communication. Springer Nature Switzerland, pp 11–22, doi:10.1007/978-3031-28124-2-2.

[115] J. Wang , C. Zhao , S. He , *et al.* (2022), LogUAD: log unsupervised anomaly detection based on Word2Vec. Computer Systems Science and Engineering 2022, doi:10.32604/csse.2022.022365.

[116] M. Wang, L. Xu, and L. Guo, Anomaly detection of system logs based on natural language processing and deep learning, in 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), pp. 140–144, 2018, doi:10.1109/ICFSP.2018.8552075.

[117] Q. Wang, X. Zhang, X. Wang, *et al.*, Log sequence anomaly detection method based on contrastive adversarial training and dual feature extraction, Entropy, vol. 1, 24, 2021, doi:10.3390/e24010069.

[118] X. Wang , Q. Cao , Q. Wang , *et al.* (2022), Robust log anomaly detection based on contrastive learning and multi-scale MASS. J Supercomputer 2022:17491–17512, doi:10.1007/s11227-022-04508-1.

[119] X. Wang, L. Yang, D. Li, *et al.*, MADDC: Multi-Scale Anomaly Detection, Diagnosis and Correction for Discrete Event Logs, in Proceedings of the 38th Annual Computer Security Applications Conference, 2022, pp. 769–784, doi:10.1145/3564625.3567972.

[120] Z. Wang , Z. Chen , J. Ni , *et al.* (2021), Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection. Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining pp 3726–3734, doi:10.1145/3447548.3467125.

[121] Z. Wang , J. Tian , H. Fang , *et al.* (2022), LightLog: A lightweight temporal convolutional network for log anomaly detection on edge. Computer Networks 203, doi:10.1016/j.comnet.2021.108616.

[122] S. Wang, *et al.*, Deep learning-based anomaly detection and log analysis for computer networks, arXiv preprint arXiv:2407.05639, 2024.

[123] S. R. Wibisono and A. I. Kistijantoro, Log anomaly detection using an adaptive universal transformer, in 2019 International Conference of Advanced Informatics: Concepts, Theory, and Applications, pp. 1–6, 2019, doi:10.1109/ICAICTA.2019.8904299.

[124] T. Wittkopp, A. Acker, S. Nedelkoski, J. Bogatinovski, D. Scheinert, W. Fan, and O. Kao, A2log: attentive augmented log anomaly detection, arXiv preprint arXiv:2109.09537, 2021, doi:10.48550/arXiv.2109.09537.

[125] L. Xi, Y. Xin, S. Luo, *et al.*, Anomaly detection mechanism based on hierarchical weights through large-scale log data, in 2021 International Conference on Computer Communication and Artificial Intelligence, pp. 106–115, 2021, doi:10.1109/CCAI50917.2021.9447458.

[126] B. Xia, Y. Bai, J. Yin, Y. Li, and J. Xu, LogGAN: A log-level generative adversarial network for anomaly detection using permutation event modelling, 2021, doi:10.1007/s10796-020-10026-3.

[127] C. Xiao, J. Huang, and W. Wu, Detecting anomalies in cluster system using a hybrid deep learning model, International Symposium on Parallel Architectures, Algorithms and Programming, pp. 393–404, 2019, doi:10.1007/978-981-15-2767-8-35.

[128] Y. Xie, L. Ji, X. Cheng, *et al.*, An attention-based GRU network for anomaly detection from system logs, IEICE Transactions on Information and Systems, 2020(8):1916–1919, 2020, doi:10.1587/transinf.2020EDL8016.

[129] Y. Xie, H. Zhang, and M. A. Babar, LogGD: Detecting Anomalies from System Logs with Graph Neural Networks, in 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS), pp. 299–310, 2022, doi:10.1109/QRS57517.2022.00039.

[130] Y. Xie, H. Zhang, and M. A. Babar, LogSD: Detecting Anomalies from System Logs through Self-Supervised Learning and Frequency-Based Masking, Proceedings of the ACM on Software Engineering, vol. 1, FSE, pp. 2098–2120, 2024.

[131] W. Xu, L. Huang, A. Fox, *et al.*, Large-scale system problem detection by mining console logs, Proceedings of SOSP'09, 2009.

[132] W. Xu, L. Huang, A. Fox, *et al.*, Detecting large-scale system problems by mining console logs, Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles, pp. 117–132, 2009, doi:10.1145/1629575.1629587.

[133] P. Xu, *et al.*, Multi-source data based anomaly detection through temporal and spatial characteristics, Expert Systems with Applications, vol. 237, 121675, 2024.

[134] R. B. Yadav, P. S. Kumar, and S. V. Dhavale, A survey on log anomaly detection using deep learning, in 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), pp. 1215–1220, 2020, doi:10.1109/ICRITO48877.2020.9197818.

[135] L. Yan, C. Luo, and R. Shao, Discrete log anomaly detection: A novel time-aware graph-based link prediction approach, Information Sciences, vol. 647, 119576, 2023, doi:10.1016/j.ins.2023.119576.

[136] L. Yang, J. Chen, Z. Wang, *et al.*, Semi-supervised log-based anomaly detection via probabilistic label estimation, in 2021 IEEE/ACM 43rd International Conference on Software Engineering, pp. 1448–1460, 2021, doi:10.1109/ICSE43902.2021.00130.

[137] R. Yang, D. Qu, Y. Gao, *et al.*, NLSA-Log: An anomaly detection framework for log sequence in security management, IEEE Access, vol. 7, pp. 181152–181164, 2019, doi:10.48550/arXiv.1710.00811.

[138] S. Yen, M. Moh, T. S. Moh, *et al.*, Semi-supervised log anomaly detection through sequence modelling, 18th IEEE International Conference on Machine Learning and Applications, pp. 1334–1341, 2019, doi:10.1109/ICMLA.2019.00217.

[139] K. Yin, M. Yan, L. Xu, *et al.*, Improving log-based anomaly detection with component-aware analysis, 2020 IEEE International Conference on Software Maintenance and Evolution, pp. 667–671, 2020, doi:10.1109/ICSME46990.2020.00069.

[140] F. Yuan, Y. Cao, Y. Shang, J. Tan, and B. Fang, Insider threat detection with deep neural network," in International Conference on Computational Science, pp. 43–54, 2018, doi:10.1007/978-3-319-93698-7-4.

[141] W. Yuan, S. Ying, X. Duan, *et al.*, PVE: A log parsing method based on VAE using embedding vectors, Information Processing & Management, vol. 60, no. 5, 2023, doi:10.1016/j.ipm.2023.103476.

[142] C. Zhang, X. Wang, H. Zhang, *et al.*, Log sequence anomaly detection based on local information extraction and globally sparse transformer model, IEEE Transactions on Network and Service Management, vol. 18, no. 4, 2021, doi:10.1109/TNSM.2021.3125967.

[143] C. Zhang, X. Peng, C. Sha, K. Zhang, Z. Fu, X. Wu, ... & D. Zhang, (2022), Deeptralog: Trace-log combined microservice anomaly detection through graph-based deep learning. In Proceedings of the 44th international conference on software engineering (pp. 623-634)..

[144] D. Zhang, Y. Zheng, Y. Wen, *et al.*, Role-based log analysis applying deep learning for insider threat detection, in Proceedings of the 1st Workshop on Security-Oriented Designs of Computer Architectures and Processors, pp. 18–20, 2018, doi:10.1145/3267494.3267495.

[145] D. Zhang, D. Dai, R. Han, *et al.*, SentiLog: Anomaly detecting on parallel file systems via log-based sentiment analysis, in Proceedings of the 13th ACM Workshop, pp. 86–93, 2021,

doi:10.1145/3465332.3470873.

[146] K. Zhang , X. Di , X. Liu , *et al.* (2022), LogLR A Log Anomaly Detection Method Based on Logical Reasoning. International Conference on Wireless Algorithms, Systems, and Applications 2022 pp 489–500, httpdoi:10.1007/978-3-031-19214-2-41.

[147] L. Zhang, W. Li, Z. Zhang, *et al.*, LogAttn: Unsupervised log anomaly detection with an AutoEncoder-based attention mechanism, in International Conference on Knowledge Science, Engineering and Management, pp. 222–235, 2021, doi:10.1007/978-3-030-82153-1-19.

[148] M. Zhang , J. Chen , J. Liu , *et al.* (2022), LogST: Log Semi-supervised Anomaly Detection Based on Sentence-BERT. 2022 7th International Conference on Signal and Image Processing (ICSIP) pp 356–361, doi:10.3390/electronics12173580.

[149] X. Zhang, Y. Xu, Q. Lin, *et al.*, logRobust log-based anomaly detection on unstable log data, in Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 807–817, 2019, doi:10.1145/3338906.3338931.

[150] Z. Zhao, W. Niu, X. Zhang, *et al.*, Trine: Syslog anomaly detection with three transformer encoders in one generative adversarial network, Applied Intelligence, 2021–2022, 2021, doi:10.1007/s10489-021-02863-9.

[151] Z. Zheng, L. Yu, W. Tang, *et al.*, Coanalysis of RAS log and job log on Blue Gene/P, in 2011 IEEE International Parallel & Distributed Processing Symposium, pp. 840–851, 2011, doi:10.1109/IPDPS.2011.83.

[152] P. Zhou, Y. Wang, Z. Li, *et al.*, Logsayer: Log pattern-driven cloud component anomaly diagnosis with machine learning, in 2020 IEEE/ACM 28th International Symposium on Quality-of-Service, pp. 1–10, 2020, doi:10.1109/IWQoS49365.2020.9212954.

[153] B. Zhu, J. Li, R. Gu, *et al.*, An approach to cloud platform log anomaly detection based on natural language processing and LSTM, in 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence, pp. 1–7, 2020, doi:10.1145/3446132.3446415.

Kamiya Pithode received the BTech and MTech computer science and engineering degrees from RGPV, Bhopal, India. She is a Research scholar at VIT University, Bhopal, India. Her research includes machine learning, deep learning, cyber security, and log data analysis.

Pushpinder Singh Patheja received a PhD in Computer science and engineering from MANIT, Bhopal, India. He is a Senior Associate Professor and Division Head of Cyber Security and Digital Forensics at the VIT University, Bhopal, India. He received a recognised prestigious EC-Council Cybersecurity Mentor award in November 2023. His primary research interests include Cyber Security, Digital forensics, Networking and cloud computing.