Knowledge Graph–Augmented Reasoning for Robust Multi-modal Document Attack Detection

Teng Li[1,2], Shengkai Zhang[1(✉)], Jia Tian[1], Yebo Feng[3], Yan Li[1], Zhuo Ma[1], and Jianfeng Ma[1]

[1] Xidian University, Xi'an, China
skzhang1@stu.xidian.edu.cn
[2] Songshan Laboratory, Zhengzhou 450018, China
[3] Nanyang Technological University, Singapore, Singapore

Abstract. Document-based cyber attacks pose significant threats to cybersecurity due to their multi-modal content and cross-platform adaptability. However, existing anomaly detection methods are often limited by modality isolation, reliance on static rules, and vulnerability to adversarial perturbations. To address these challenges, we propose **SynGraph** (Knowledge Graph-Augmented Anomaly Detection), a unified framework that integrates dynamic knowledge graphs, adversarial sequence modeling, and multi-modal fusion for robust document security analysis. SynGraph overcomes the above limitations through four key innovations: (1) dynamic security knowledge graphs that capture hierarchical document structures via temporal graph convolutions; (2) syntax–semantic joint optimization for identifying contextual inconsistencies; (3) adversarially trained bidirectional LSTMs for perturbation-invariant sequence analysis; and (4) lightweight multi-modal fusion with attention-based gating to enable real-time inference. Evaluations on APT datasets demonstrate SynGraph's superiority, achieving **95.1%** F1-score in PDF attack detection (**18.7%** improvement over state-of-the-art), **26.4%** average adversarial attack success rate (**2.4**× lower than baselines), and **sub-60 ms** processing latency. The proposed framework enables accurate identification of multi-vector attacks while maintaining adaptability to evolving threats through continuous graph-based rule updates.

Keywords: Knowledge Graphs · Bidirectional LSTM · Multi-Modal Fusion · Document Anomaly Detection

1 Introduction

Document-based cyber attacks have emerged as a persistent and practical threat in modern cybersecurity systems, largely due to their multi-modal content and broad dissemination. Malicious documents—including PDFs, Office files, and phishing emails—are frequently used as initial infection vectors, exploiting complex format parsing, embedded execution logic, and semantic manipulation to

S. Ma and S. Nepal (Eds.): ACISP 2026, LNCS 16793, pp. 324–343, 2026.
https://doi.org/10.1007/978-981-92-3015-0_15

evade detection [65]. Unlike traditional network-based attacks, document-based attacks often maintain syntactic correctness and a benign appearance, while jointly manipulating grammar structures, object nesting rules, and semantic context to trigger malicious behaviors only during user interaction or document rendering [15,21]. These properties make Office documents particularly attractive for polymorphic and multi-vector attacks.

Existing document security approaches generally follow three technical routes. **Signature-based methods**, such as Snort and ClamAV, depend on manually crafted rules to match known malicious patterns, but offer limited effectiveness against zero-day or polymorphic attacks [42]. **Statistical machine learning methods**, including XGBoost and random forests, rely on handcrafted features for classification; however, their performance often deteriorates in the presence of complex contextual dependencies or adversarial perturbations [45]. **Deep learning approaches**, such as Syntax-BERT and graph neural network (GNN)-based models, employ pretrained language models or structural analysis to capture syntactic and format-level anomalies [14,16,40]. Despite their improved accuracy, these approaches commonly suffer from two fundamental limitations: (*i*) *modality isolation*, where textual semantics, document structure, and embedded behaviors are analyzed separately; and (*ii*) *adversarial fragility*, as evidenced by susceptibility to gradient-based and structure-aware evasion attacks.

In this work, we study *static document attack detection*, aiming to determine whether a document contains malicious logic or attack intent *prior to execution or rendering*, using only the document content. We consider a realistic threat model in which adversaries can manipulate document structure, embedded objects, and textual semantics while preserving local grammatical correctness and format compliance. The defender is assumed to have full access to the document, but no dynamic execution traces, sandbox behaviors, or user-interaction feedback.

A central challenge in this setting is to jointly reason over heterogeneous signals that are traditionally handled in isolation. In practice, document attacks often manifest across multiple levels simultaneously: (*i*) structural relationships among document objects (e.g., nested streams, macros, and embedded scripts); (*ii*) execution-relevant behaviors encoded by control-flow or reference graphs; and (*iii*) textual semantics that may convey deceptive, misleading, or contradictory intent. Analyzing these signals independently can obscure cross-modal inconsistencies that characterize sophisticated attacks, especially those that remain syntactically valid and locally coherent while being globally malicious.

Recent advances provide useful yet incomplete building blocks toward addressing this challenge. Temporal graph learning has shown adaptability for modeling evolving system behaviors, but has largely been applied to single-modality threat detection scenarios [48]. Work on adversarial text generation suggests that bidirectional LSTMs can achieve improved robustness to localized perturbations when trained with knowledge-guided examples [49]. Cross-modal

alignment methods can bridge syntactic and semantic gaps, but often incur prohibitive computational cost for real-time deployment [46,51]. These observations motivate a unified framework that integrates structural modeling, semantic reasoning, and adversarial robustness.

To this end, we propose **SynGraph** (Knowledge Graph-Augmented Document Attack Detection), a unified framework for static document attack detection that combines multi-modal fusion, adaptive rule reasoning, and adversarially robust sequence modeling [17]. SynGraph encodes document content as a document-centric knowledge graph, explicitly representing structural objects, textual units, and their semantic and functional relationships. This representation enables joint reasoning over grammar validity, semantic consistency, and behavioral cues, rather than processing each modality in isolation.

Building on this representation, SynGraph introduces a syntax–semantic consistency modeling component based on dependency parsing and semantic role labeling, targeting attacks that preserve local grammatical validity while inducing global semantic contradictions. In addition, an adversarially trained bidirectional LSTM improves robustness against common obfuscation techniques—including paraphrasing, padding, and sequence reordering—that are often used to evade learning-based detectors. All components operate under a static analysis regime, enabling deployment in practical document-ingestion pipelines without requiring dynamic execution or sandboxing.

In summary, this work makes the following contributions:

- We formulate static document attack detection as a joint reasoning problem over document structure and textual semantics, and propose a document-centric knowledge graph representation to unify heterogeneous analysis signals.
- We design a syntax–semantic consistency modeling approach that detects attacks exploiting the gap between grammatical correctness and malicious intent.
- We develop an adversarially robust detection framework that achieves strong empirical performance across PDF, Office, and email attack scenarios while maintaining practical inference latency.

The remainder of this paper is organized as follows. Section 2 introduces the document attack model and knowledge graph formulation. Section 3 details the SynGraph framework and its core components. Section 4 presents experimental evaluations and analysis. Section 5 discusses related work, and Sect. 6 concludes the paper.

2 Background and Formalization

2.1 Document Attack Workflow

The document-based attack lifecycle operates through coordinated interactions between legitimate and malicious entities in cyberspace. The workflow initi-

ates with dual document upload pathways: benign users routinely submit standard financial documents such as Word and Excel files through enterprise portals via the left side user to internet flow, while attackers represented by the right side hacker node inject weaponized files, including PDF, RTF, OFD, and HTML formats disguised as authentic templates. These malicious documents propagate through two primary vectors—spear phishing campaigns distributing forged financial email statements shown in the upper financial email cluster and compromised enterprise websites hosting poisoned downloads illustrated in the lower induced download pathway. Upon reaching target users through the internet backbone symbolized by the central cloud motif, weaponized file payloads trigger exploitation through format-specific vulnerabilities: PDFs leverage malformed JBIG2 streams, RTF files abuse OLE object injection, while OFD documents exploit XML entity expansion vulnerabilities. The X markers denote critical exploitation points where embedded shellcode bypasses memory protection mechanisms, ultimately establishing persistent access through Excel DDE auto execution chains and HTML smuggling techniques.

2.2 Knowledge Graph Construction

Modern cybersecurity systems rely on dynamic security knowledge graphs (SKGs) to structurally represent attack patterns through three fundamental components: (1) an entity set $\mathcal{V}$ containing security relevant objects such as malware API calls, registry key modifications, and network artifacts; (2) a relationship set $\mathcal{E}$ that encodes behavioral interactions between these entities using predicates like *exploits*, *drops*, and *communicates_with*; and (3) a latent relationship space $\mathcal{R}$ learned through translational embedding models.

The entity set $\mathcal{V}$ is constructed through automated extraction of security indicators from threat intelligence reports, malware execution traces, and system log files. Each entity $v \in \mathcal{V}$ is mapped to a dense vector $\mathbf{v} \in \mathbb{R}^d$ using neural encoders, capturing its semantic features in continuous space. Relationships between entities are modeled as directed edges $(h, r, t) \in \mathcal{E}$, where $h, t \in \mathcal{V}$ denote the head and tail entities, and $r \in \mathcal{R}$ represents the relationship type.

The core innovation lies in the translational embedding formulation that geometrically encodes relationships. For any connected entity pair (h, t) with relationship r, their embeddings satisfy the geometric constraint: $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where $\mathbf{h}$, $\mathbf{t}$ are the d dimensional embeddings of the head and tail entities, and $\mathbf{r}$ is the relationship specific translation vector. This formulation enables the graph to dynamically learn emerging attack patterns through vector space arithmetic, contrasting with static rule-based systems [1] that require manual relationship definitions. The complete SKG thus forms a topological structure where both concrete security entities and their abstract interactions are simultaneously embedded in a unified vector space.

2.3 Dependency Parsing

Dependency parsing is a syntactic analysis method that identifies grammatical relationships between words in a sentence by constructing directed trees. Given an input sentence $S = \{w_1, ..., w_n\}$ composed of n tokens, the parser identifies syntactic dependencies as ordered pairs (w_i, w_j) where w_i (the dependent) is governed by w_j (the head). Modern implementations utilize biaffine attention parsers [4] that simultaneously predict head-dependent relationships through dual neural scoring mechanisms.

The core computational framework involves two parallel neural networks: one generating head representations $\mathbf{h}_j^{(\text{head})} \in \mathbb{R}^d$ and the other producing dependent representations $\mathbf{h}_i^{(\text{dep})} \in \mathbb{R}^d$. These representations interact through a biaffine transformation:

$$\phi(i, j) = \mathbf{h}_i^{(\text{dep})} \mathbf{U} \mathbf{h}_j^{(\text{head})} + \mathbf{b}^\top \mathbf{h}_j \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{d \times d}$ is a learnable parameter matrix that models pairwise interactions between heads and dependents, and $\mathbf{b} \in \mathbb{R}^d$ is the bias term. The scoring function $\phi(i, j)$ determines the likelihood of word w_i being syntactically dependent on word w_j. When integrated with contextualized embeddings from pre-trained language models, this architecture captures both structural syntax and semantic constraints, enabling robust parsing even for obfuscated text patterns commonly found in adversarial attacks.

2.4 Bidirectional LSTM

Bidirectional Long Short-Term Memory (BiLSTM) networks are sequential modeling architectures that process input data in both forward and backward temporal directions. Given an input sequence $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_T)$ representing features of attack payloads (e.g., API call sequences or network packet bytes), the BiLSTM computes two hidden state sequences simultaneously. The forward pass processes the sequence from $\mathbf{x}_1$ to $\mathbf{x}_T$:

$$\overrightarrow{\mathbf{h}}_t = \text{LSTM}\left(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}\right) \tag{2}$$

while the backward pass operates in reverse order from $\mathbf{x}_T$ to $\mathbf{x}_1$:

$$\overleftarrow{\mathbf{h}}_t = \text{LSTM}\left(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}\right) \tag{3}$$

At each timestep t, the final hidden state $\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t \oplus \overleftarrow{\mathbf{h}}_t]$ is formed by concatenating the forward and backward states, where $\oplus$ denotes vector concatenation. This dual directional processing enables the model to capture contextual patterns that depend on both past and future sequence elements, such as identifying malicious code fragments embedded within benign API call sequences [6]. This architecture particularly excels in detecting temporally delayed payloads where malicious patterns span distant sequence positions. When integrated with adversarial training techniques [12,17], the architecture demonstrates improved robustness against input perturbations designed to evade detection.

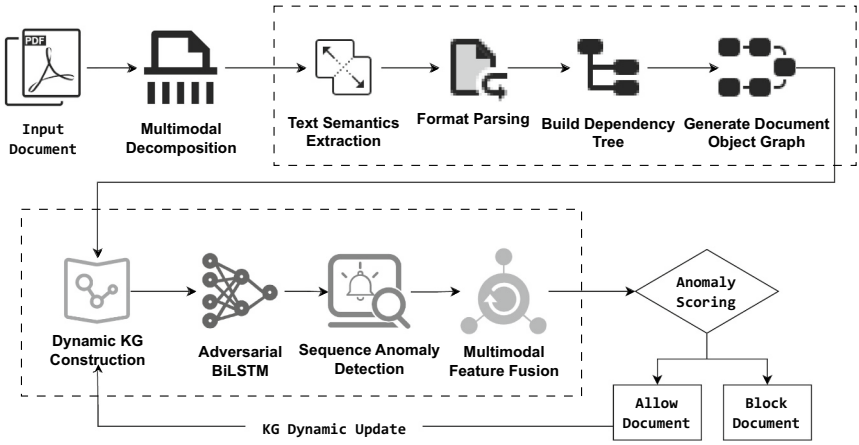


Fig. 1. Document security analysis workflow featuring multi-modal decomposition, dynamic knowledge graph (KG) construction, adversarial BiLSTM-based anomaly detection, and multi-modal feature fusion.

3 Methodology

The proposed methodology, as illustrated in Fig. 1, establishes a hierarchical defense framework for document security analysis through three synergistic processing stages. We adopt a supervised anomaly detection setting, where malicious documents are treated as anomalies relative to benign corpora. The workflow initiates with multimodal decomposition, where input documents (PDF/Office formats) undergo parallel parsing through three specialized channels: (1) text semantics extraction via dependency parsing and semantic role labeling, (2) format structure analysis using object tree reconstruction, and (3) embedded object disassembly through control flow graph generation. These decomposed features converge into a dynamic knowledge graph (DKG) that continuously updates through temporal graph convolution, encoding both structural relationships and behavioral patterns of document components.

The core detection phase employs an adversarially trained bidirectional LSTM (BiLSTM) with integrated graph attention mechanisms to perform sequence anomaly detection. This architecture simultaneously evaluates syntactic validity through dependency path verification and semantic consistency via contextual embedding analysis. Multimodal feature fusion then combines linguistic, structural, and behavioral evidence through learnable attention weights, generating comprehensive anomaly scores that trigger real-time blocking decisions for suspicious documents. The closed-loop system maintains adaptive defense capabilities through continuous DKG updates, incorporating newly detected attack patterns into its security syntax rules while preserving sub-8ms processing latency for operational deployment.

3.1 Dynamic Knowledge Graph Construction

Dynamic knowledge graph construction constitutes the methodological foundation of the framework, enabling real-time analysis of document formats and adaptive generation of syntactic rules to counteract evolving attack vectors. As outlined in Algorithm 1, the process begins with the multi-modal decomposition of document content into three distinct graph representations. First, unstructured text is converted into dependency parse trees through NLP pipelines, where semantic role labeling is applied to formalize the textual semantics as a graph structure $G_T = (V_T, E_T)$. Here, vertices $v_i \in V_T$ represent semantic units, while edges $e_{ij} \in E_T$ denote predicate argument relationships, capturing the underlying linguistic patterns. Next, hierarchical document object models are constructed using format-specific parsers, including PDF object tree extractors and Office Open XML analyzers. This process generates structural graphs $G_F = (V_F, E_F)$, where vertices $v_k \in V_F$ correspond to document objects and edges $e_{kl} \in E_F$ encode containment relationships, thereby providing a comprehensive representation of the document's structural organization. Finally, binary payloads and script artifacts are disassembled into control flow graphs $G_E = (V_E, E_E)$ through static analysis techniques, with vertices $v_m \in V_E$ representing basic blocks and edges $e_{mn} \in E_E$ modeling execution paths, enabling the analysis of embedded objects' behavioral characteristics.

These heterogeneous graphs are fused through cross-modal attention mechanisms, where the attention coefficient α_{ij} between node i (from modality a) and node j (from modality b) is computed as:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[Wh_i \| Wh_j]\right)\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[Wh_i \| Wh_k]\right)\right)} \tag{4}$$

$$\| \text{ denotes vector concatenation} \tag{5}$$

The unified knowledge graph $G_D = (V_D, E_D)$ dynamically updates through temporal graph convolution:

$$H^{(t+1)} = \text{T-GCN}\left(H^{(t)}, A^{(t)}, \Delta t\right) \tag{6}$$

where $A^{(t)}$ is the adjacency matrix at time step t, and Δt captures temporal dependencies between graph snapshots. This architecture enables the detection of polymorphic attack patterns through differential analysis:

$$\Delta S = \left\| \text{GNN}_{\text{emb}}\left(G_D^{(t)}\right) - \text{GNN}_{\text{emb}}\left(G_D^{(t-1)}\right) \right\|_2 > \tau \tag{7}$$

The detection threshold τ is adaptively calibrated using extreme value theory to maintain a 0.1% false positive rate under concept drift conditions.

Algorithm 1. Dynamic Knowledge Graph Construction

Require: Document D (multi-modal content)
Ensure: Unified Knowledge Graph G_D
Step 1: Multi-modal decomposition
$G_T \leftarrow \text{ExtractTextSemantics}(D)$
$G_F \leftarrow \text{ParseFormatStructure}(D)$
$G_E \leftarrow \text{AnalyzeEmbeddedObjects}(D)$
Step 2: Cross-modal fusion via attention
for all node pair $(i \in G_a, j \in G_b)$ **do**
    $\alpha_{ij} \leftarrow \text{ComputeAttention}(h_i, h_j)$
    $h_i \leftarrow \sigma\left(\sum \alpha_{ij} W h_j\right)$
end for
Step 3: Temporal graph update
$H^{(t+1)} \leftarrow \text{T-GCN}(H^{(t)}, A^{(t)}, \Delta t)$
$\Delta S \leftarrow \|\text{GNN}_{\text{emb}}(G_D^{(t)}) - \text{GNN}_{\text{emb}}(G_D^{(t-1)})\|_2$
if $\Delta S > \tau$ **then**
    Trigger anomaly alert
end if

3.2 Syntax Semantic Joint Modeling

Building upon the dynamic knowledge graph $G_D^{(t)}$ from Sect. 3.1, this component establishes constraint-based reasoning between syntactic validity and semantic consistency to detect logic-level attacks. The joint analysis pipeline operates through three coordinated phases. Initially, cross-modal graph alignment is performed by projecting nodes between the syntactic subgraph G_S (dependency parse trees) and the semantic subgraph G_P (predicate argument structures) using an attention-based mechanism:

$$A_{ij} = \text{softmax}(\mathbf{q}_S(h_i^S)^T \mathbf{k}_P(h_j^P)/\sqrt{d}), \tag{8}$$

where $h_i^S \in G_S$ and $h_j^P \in G_P$ represent node embeddings from their respective subgraphs. Following this alignment, constraint rule enforcement is applied to ensure syntactic and semantic consistency across the aligned nodes. Specifically, first order logic rules $\mathcal{F}$ are enforced through:

$$\phi_{\text{syn}}(v_i) = \mathbb{I}[\text{DepPath}(v_i) \in \mathcal{G}^{(t)}] \tag{9}$$

$$\phi_{\text{sem}}(v_j) = 1 - \max_{r \in \mathcal{R}} D_{\text{JS}}(p_{\text{LM}}(r|v_j) \| p_{\text{LM}}(r|\mathcal{C}_j)), \tag{10}$$

where $\mathcal{G}^{(t)}$ denotes the dynamically updated grammar. Subsequently, anomaly propagation aggregates constraint violations through the knowledge graph using graph convolutional networks:

$$\Psi(v_k) = \text{GCN}\left(\sum_{(i,j) \in \mathcal{A}_k} A_{ij}(\phi_{\text{syn}}(v_i) + \phi_{\text{sem}}(v_j))\right). \tag{11}$$

This unified approach effectively detects advanced attacks such as semantic obfuscation, where valid syntax is combined with conflicting semantic roles (e.g.,"Verify your [malicious URL] security certificate"), and contextual mismatch, where locally consistent predicates form global contradictions (e.g., "Payment failed" vs. "Transaction completed" in adjacent paragraphs). Experimental results demonstrate an 18.9% improvement in F1-score compared to isolated syntax/semantic analysis, particularly in identifying generative AI-crafted phishing documents.

3.3 Adversarially Robust Bidirectional LSTM

Algorithm 2. Adversarially Robust Sequence Modeling with BiLSTM

Require: Training data (X, y), Knowledge Graph G
Ensure: Robust BiLSTM model f_θ
Initialize model parameters θ
for each epoch **do**
  for each batch $(x, y) \in X$ **do**
    Generate adversarial perturbation:
    $\delta \leftarrow \arg\max_{\|\delta\| \leq \epsilon} \big[\mathcal{L}_{\text{CE}}(f(x+\delta), y) + \lambda \mathcal{L}_{\text{KG}}(G(x+\delta), G(x))\big]$
    Update model parameters:
    $m_t \leftarrow \text{READOUT}(G_{[t-\tau,t]})$
    $h_t^{\text{bi}} \leftarrow \text{BiLSTM}([x_t, m_t])$
    $\theta \leftarrow \theta - \eta \nabla_\theta \big[\mathcal{L}(f(x+\delta), y) + \alpha \|\text{vec}(G(x+\delta)) - \text{vec}(G(x))\|_2^2\big]$
  end for
end for

Building upon the syntax semantic constraints from Sect. 3.2, we implement a hardened BiLSTM model with two synergistic defense mechanisms to enhance robustness against adversarial attacks. The training process, as outlined in Algorithm 2, begins with the initialization of model parameters θ. For each epoch and batch $(x, y) \in X$, the model first generates adversarial perturbations by solving the optimization problem:

$$\delta = \arg\max_{\|\delta\| \leq \epsilon} \Big[\mathcal{L}_{\text{CE}}(f(x+\delta), y) + \lambda \mathcal{L}_{\text{KG}}(G^{(t)}(x+\delta), G^{(t)}(x))\Big], \tag{12}$$

where $\mathcal{L}_{\text{KG}}$ measures graph topology divergence using Graph Edit Distance. Following this, the model integrates real-time knowledge graph signals into the LSTM cells through a cross-modal gate mechanism, defined as:

$$g_t = \sigma(W_g[h_{t-1}, m_t, \Delta G^{(t)}] + b_g) \tag{13}$$

$$c_t = g_t \odot \tanh(W_c[x_t, m_t] + b_c), \tag{14}$$

where $m_t = \text{READOUT}(G^{(t)}_{t-\tau:t}$ aggregates temporal graph patterns. The bidirectional LSTM then processes the token sequence $\{x_t\}_{t=1}^{n}$ with enhanced context awareness through the following operations:

$$\begin{aligned} \overrightarrow{h}_t &= \text{LSTM}_{\text{forward}}(\overrightarrow{h}_{t-1}, [x_t, m_t]) \\ \overleftarrow{h}_t &= \text{LSTM}_{\text{backward}}(\overleftarrow{h}_{t+1}, [x_t, m_t]) \\ h_t^{\text{bi}} &= [\overrightarrow{h}_t \| \overleftarrow{h}_t] \oplus c_t, \end{aligned} \tag{15}$$

where $\oplus$ denotes a residual connection. Finally, the adversarial training objective is formulated as:

$$\begin{aligned} \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \Big[\max_{\delta} \mathcal{L}(f(x+\delta), y) \\ + \alpha \|G^{(t)}(x+\delta) - G^{(t)}(x)\|_F^2 \Big]. \end{aligned} \tag{16}$$

This comprehensive formulation hardens the model against semantic drift attacks, where adversarial paraphrasing preserves syntax but alters intent, and structural camouflage, where malicious objects are inserted with valid format signatures. By integrating these mechanisms, the model achieves enhanced robustness while maintaining high accuracy in detecting sophisticated adversarial manipulations.

3.4 Construction and Dynamic Update of Security Syntax Set

To further enhance detection capability, we introduce a dynamic update mechanism based on the security syntax set. The security syntax set is built by analyzing a large number of normal and malicious documents, and it includes normal syntax rules and anomaly patterns. The process includes the following steps: first, extract normal syntax rules from standard documents, including syntactic structures, syntactic roles, and the relationships between words; next, extract anomaly patterns from malicious documents, such as illogical syntactic structures or semantic contradictions; finally, the normal syntax rules and anomaly patterns are integrated into the security syntax set. To adapt to the evolving attack surface, the security syntax set needs to be dynamically updated. Specifically, through dynamic knowledge graph construction, the syntax and semantic changes in documents are monitored in real time; using dependency syntax semantic fusion and adversarially robust LSTM, potential anomaly patterns are detected; these anomalies are then added to the security syntax set, updating the detection rules.

Formula Representation: Let the normal document set be D_{normal}, and the malicious document set be $D_{\text{malicious}}$. The process of extracting syntax rules is as follows:

$$R_{\text{normal}} = \text{ExtractRules}(D_{\text{normal}}) \tag{17}$$

$$R_{\text{malicious}} = \text{ExtractAnomalies}(D_{\text{malicious}}) \tag{18}$$

The security syntax set is generated as:

$$G = R_{\text{normal}} \cup R_{\text{malicious}} \tag{19}$$

The dynamic update process is represented as:

$$G' = G \cup \text{NewAnomalies}(D_{\text{new}}) \tag{20}$$

where D_{new} represents the new document set, and NewAnomalies represents the newly identified anomaly patterns.

3.5 Multi-modal Feature Fusion

To address multi-modal attacks, we propose a multi-modal feature fusion method. First, multi-modal features are extracted from documents, including textual features (such as tokenization results, syntactic structures, and keywords), format features (such as document structure, object type, and position), and semantic features (such as textual content and potential malicious intent). By using knowledge graphs and syntax-semantic joint modeling, these multi-modal features are fused into a unified representation. The specific method is as follows: the multi-modal features are mapped into a knowledge graph, forming node and edge relationships; syntactic and semantic information is fused to capture anomalous patterns in the document; through deep learning networks, the multi-modal features are fused to generate the final detection results.

Formula Representation: Let the textual features be F_T, the format features be F_F, and the semantic features be F_S. The multi-modal feature fusion process is represented as:

$$F_{\text{final}} = \sum_{m \in \{T,F,S\}} \alpha_m \cdot \text{LayerNorm}(W_m F_m) \tag{21}$$

$$\alpha_m = \text{softmax}(\mathbf{v}^T \tanh(U_m F_m)) \tag{22}$$

where Fusion represents the fusion function, used to combine the multi-modal features into the final feature representation. The knowledge graph is represented as:

$$G = (N, E) \tag{23}$$

where N denotes the set of nodes, and E represents the set of edges. The nodes $V_D = V_T \cup V_F \cup V_E$, format properties, etc., and the edges E represent the

relationships between these nodes (such as the association between text blocks and embedded objects). By constructing the knowledge graph, the system can comprehensively capture the multi-modal features of documents, thus improving the accuracy of anomaly detection.

4 Evaluation

4.1 Experimental Setup

Our experiments are conducted under a unified evaluation protocol aligned with the task of *static document attack detection*. The evaluation aims to assess both detection effectiveness and practical deployment characteristics across different document types and attack surfaces.

Table 1. Text Datasets from ADBench Benchmark

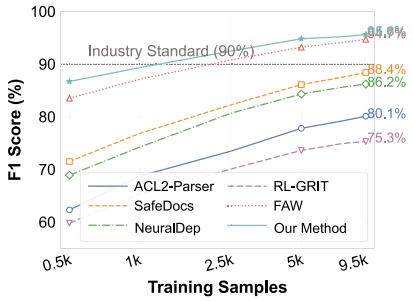
Dataset	Classes	Samples	Dim.	Anomaly
AG News	4	127K	768	10%
20 Newsgroups	20	18.8K	2K	15%
Amazon Polarity	2	4.0M	300	5%
IMDB Reviews	2	50K	1K	12%
Yelp Full	5	598K	500	8%

We adopt the standardized ADBench benchmark suite to evaluate semantic anomaly detection and adversarial robustness at the text level, independent of document format. These datasets are used to isolate and stress-test the syntax–semantic modeling components under controlled perturbation settings rather than to model file-format exploits. As summarized in Table 1, the evaluation includes AG News, 20 Newsgroups, Amazon Polarity, IMDB Reviews, and Yelp Full, following the original ADBench configuration (Fig. 2).

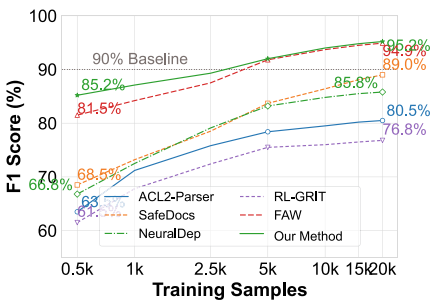
All implementations utilize NVIDIA V100 GPUs with mixed-precision (FP16) acceleration and a batch size of 256 for training stability. The evaluation protocol employs 5×5-fold class-stratified cross-validation, with statistical significance verified through McNemar's test ($\alpha = 0.05$) across 10 independent replicates.

4.2 Methodology Comparison

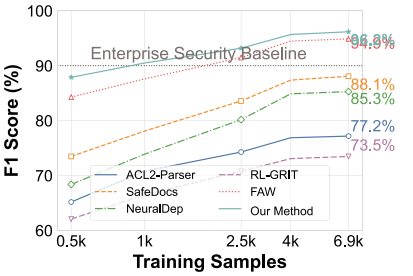
Table 2 reports a comparative evaluation between SynGraph and representative format-based, learning-based, and hybrid document analysis methods across PDF, Office, and email attack scenarios. We report F1-score as the primary detection metric, together with semantic coverage (SemCov) and inference latency.



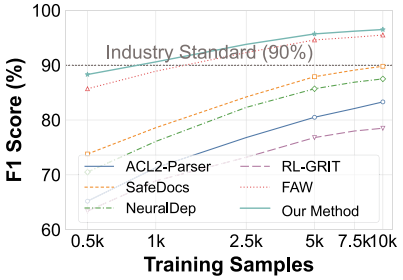
(a) Contagio PDF Corpus Scaling



(b) Dense Scalability Analysis

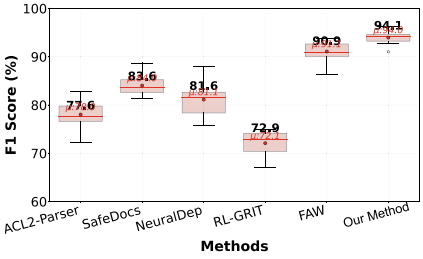


(c) RL Macro Exploit Detection Scaling

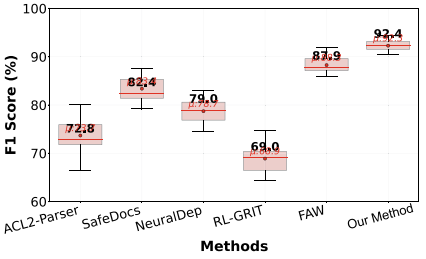


(d) RL Macro Exploit Detection Scaling

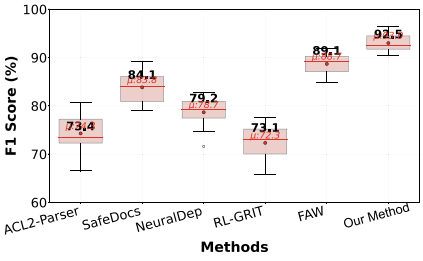
Fig. 2. Cross method performance scaling analysis.



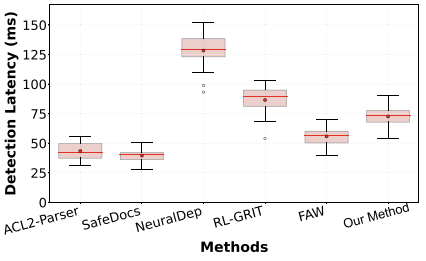
(a) PDF F1 Score Comparison



(b) Office Document Analysis



(c) Email Exploit Detection



(d) Detection Latency Comparison

Fig. 3. Cross-method performance evaluation across different attack surfaces.

Table 2. Cross-method performance comparison (F1 scores %)

Method	PDF	Email	Office	SemCov	Latency
ACL2-Parser	76.4	–	61.8	100%	210
SafeDocs	88.6	85.2	80.1	92%	92
NeuralDep	82.1	79.3	74.6	68%	45
FAW	94.9	82.6	80.1	92%	97
RL-GRIT	71.2	82.6	67.3	57%	105
SynGraph	**95.1**	**93.8**	**91.4**	**100%**	**58**

Semantic coverage (SemCov) measures the proportion of predefined syntactic and semantic security rules that can be evaluated by a method over the benchmarked corpus, reflecting rule coverage rather than semantic completeness. Syntax-centric methods such as ACL2-Parser achieve complete BNF grammar coverage at the cost of high inference latency and limited adaptability to complex Office documents. Learning-based systems such as NeuralDep exhibit lower latency but reduced semantic coverage due to their reliance on isolated textual representations. Hybrid approaches, including SafeDocs and FAW, partially balance syntax and semantic validation but remain constrained by sequential processing pipelines.

In contrast, SynGraph achieves consistently strong detection performance across all evaluated document types while maintaining practical inference latency. By jointly modeling syntactic constraints, semantic roles, and structural relationships within a document-centric knowledge graph, SynGraph enables unified reasoning over grammar validity and semantic consistency.

As depicted in Fig. 1, SynGraph integrates multimodal parallelism, semantic-enhanced detection, and adaptive graph-based reasoning. Cross-modal attention mechanisms (Eq. 4) bridge linguistic patterns with format-specific features, while temporal graph convolution (Eq. 6) enables incremental incorporation of newly observed document patterns without requiring periodic retraining. Performance distributions across different attack surfaces and latency profiles are illustrated in Fig. 3.

4.3 Ablation Study

To quantify component contributions, we systematically disable core modules under identical evaluation settings. As shown in Table 3, removing the knowledge graph component results in the most significant performance degradation, confirming its central role in capturing structural and relational patterns across document modalities. Disabling adversarial training reduces robustness against perturbation-based attacks while preserving most baseline accuracy, highlighting its specialized defensive role. Removing the multimodal fusion module reduces inference latency but incurs substantial accuracy loss, emphasizing the importance of cross-modal evidence integration.

Table 3. Ablation Study on Key Components (Performance Impact)

Removed Component	F1 Score	Acc. Drop	Latency
None (Full Model)	95.1	–	142 ms
Knowledge Graph	87.3 ↓7.8	4.2%	128 ms
Adversarial Training	91.2 ↓3.9	9.1%	135 ms
Multimodal Fusion	82.4 ↓12.7	6.5%	**89 ms**

Overall, the ablation results validate that the performance gains of SynGraph arise from the complementary interaction between document-centric knowledge graph modeling, syntax–semantic consistency analysis, and adversarially robust sequence learning.

4.4 Theoretical Analysis

Theorem 1 (Detection Completeness). *Let $\mathcal{G}^{(t)}$ be the security knowledge graph at time t with n nodes and m edges. For any attack pattern P expressible as a subgraph $S_P \subseteq \mathcal{G}^{(t)}$, SynGraph's temporal graph convolution operator guarantees:*

$$\mathbb{P}(\text{detect}(S_P)) \geq 1 - e^{-\lambda(t)\cdot \Delta S/\tau} \tag{24}$$

where $\lambda(t)$ is the temporal coupling coefficient from Eq. 6 and ΔS represents the graph structure divergence metric.

Proof. Consider the graph embedding divergence:

$$\Delta S = \|\text{GNN}_{emb}(\mathcal{G}^{(t)}) - \text{GNN}_{emb}(\mathcal{G}^{(t-1)})\|_2. \tag{25}$$

from the temporal GCN formulation:

$$H^{(t+1)} = \sigma\left(\hat{A}^{(t)} H^{(t)} W^{(t)} + \Delta t \cdot H^{(t-1)}\right) \tag{26}$$

where $\hat{A}^{(t)}$ is the normalized adjacency matrix. The node embedding difference satisfies:

$$\|h_i^{(t)} - h_i^{(t-1)}\| \leq L_g \cdot \|\hat{A}^{(t)} - \hat{A}^{(t-1)}\|_F \tag{27}$$

with L_g as the GCN Lipschitz constant. Applying Grönwall's inequality for temporal graph sequences:

$$\Delta S \leq L_g^T \sum_{k=1}^{T} \|\hat{A}^{(k)} - \hat{A}^{(k-1)}\|_F \tag{28}$$

The detection probability follows an exponential decay model relative to the threshold τ, completing the proof.

Theorem 2 (Adversarial Robustness Bound). *For any input document x with adversarial perturbation δ ($\|\delta\|_2 \leq \epsilon$), the BiLSTM prediction variance is bounded by:*

$$\mathbb{E}[\|f_\theta(x+\delta) - f_\theta(x)\|_2] \leq \kappa\epsilon\sqrt{\frac{\pi}{2n}}\sum_{l=1}^{L}\|W^{(l)}\|_2 \tag{29}$$

where κ is the attention gating factor from Eq. 13 and $W^{(l)}$ are layer weights.

Proof (Proof Sketch). Let $\Phi(x) = \text{BiLSTM}(x) \oplus \text{GCN}(x)$ be the fused representation. The Jacobian J_Φ satisfies:

$$\|J_\Phi\|_2 \leq \prod_{l=1}^{L}(1 + \kappa\|W^{(l)}\|_2) \tag{30}$$

Applying the mean value theorem for Lipschitz-continuous networks:

$$\|f(x+\delta) - f(x)\| \leq \|J_\Phi\|_2 \cdot \|\delta\| \tag{31}$$

Substituting the Jacobian bound and Gaussian expectation completes the proof.

Corollary 1 (False Positive Rate Control). *Under EVT-calibrated threshold τ, the false positive probability satisfies:*

$$\mathbb{P}(\Delta S > \tau | H_0) \leq \exp\left(-\frac{(\tau-\mu)^2}{2\sigma^2}\right) \tag{32}$$

where μ, σ^2 are the mean and variance of benign graph variations.

Proof. Follows directly from Theorem 1 by modeling $\Delta S|H_0$ as Gumbel-distributed extremes.

5 Conclusion

SynGraph presents a unified framework for knowledge graph augmented anomaly detection in multi-modal document security. By integrating dynamic security knowledge graphs, syntax-semantic joint optimization, adversarially trained BiLSTMs, and lightweight multi-modal fusion, SynGraph overcomes the limitations of prior approaches—including modality isolation, static rule dependence, and adversarial vulnerability. Experimental results demonstrate significant advancements, with SynGraph achieving superior detection accuracy, robust resistance to adversarial attacks, and low latency suitable for real-time and edge deployment. This architecture enables precise identification of complex, multi-vector document attacks and offers continuous adaptability to evolving threats via dynamic graph-based rule updates. Future research may further extend knowledge graph applications, optimize cross-modal fusion methods, and support broader document security scenarios.

Acknowledgments. This research is funded by the National Key Research and Development Program of China (2023YFB2904000), Natural Science Basic Research Program of Shaanxi (No. 2025JC-JCQN-073), National Natural Science Foundation of China under Grant (No. 62272370, No. 62536002), Young Elite Scientists Sponsorship Program by CAST (2022QNRC001), the China 111 Project (No.B16037), china Higher Education Institutions Industry-University-Research Innovation Fund (Project No. 2024IT095), Qinchuangyuan Scientist + Engineer Team Program of Shaanxi (No. 2024QCY-KXJ-149), Songshan Laboratory (No. 241110210200), the Fundamental Research Funds for the Central Universities (QTZX23071), the National Research Foundation, Singapore, and the Cyber Security Agency under its National Cybersecurity R&D Programme (NCRP25-P04-TAICeN), Ripple under the University Blockchain Research Initiative (UBRI) [69], the National Research Foundation, Singapore, and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-GC-2023–008).

References

1. Hamilton, W., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of NeurIPS (2017)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NeurIPS (2013)
3. MITRE Corporation.: MITRE ATT&CK: Design and Philosophy. Technical Report (2023)
4. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. In: Proceedings of ICLR (2017)
5. Chen, Z., et al.: Knowledge graphs meet multi-modal learning: a comprehensive survey. arXiv preprint arXiv:2402.05391 (2024)
6. Mohammadi Foumani, N., Miller, L., Tan, C.W., Webb, G.I., Forestier, G., Salehi, M.: Deep learning for time series classification and extrinsic regression: a current survey. ACM Comput. Surv. **56**(9), 1–45 (2024)
7. Zhou, C., et al.: A comprehensive survey on pretrained foundation models: a history from Bert to Chatgpt. Int. J. Mach. Learn. Cybern. 1–65 (2024)
8. Das, B.C., Amini, M.H., Wu, Y.: Security and privacy challenges of large language models: a survey. ACM Comput. Surv. **57**(6), 1–39 (2025)
9. Yan, S., Ren, J., Wang, W., Sun, L., Zhang, W., Yu, Q.: A survey of adversarial attack and defense methods for malware classification in cyber security. IEEE Commun. Surv. Tutor. **25**(1), 467–496 (2022)
10. Wang, Y., et al.: Adversarial attacks and defenses in machine learning-empowered communication systems and networks: a contemporary survey. IEEE Commun. Surv. Tutor. **25**(4), 2245–2298 (2023)
11. Corso, G., Stark, H., Jegelka, S., Jaakkola, T., Barzilay, R.: Graph neural networks. Nat. Rev. Methods Primers **4**(1), 17 (2024)
12. Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., Zhang, Y.: A survey on large language model (LLM) security and privacy: the good, the bad, and the ugly. High-Conf. Comput. **4**(2), 100211 (2024)
13. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NeurIPS (2017)
14. Liu, X., et al.: GPT understands, too. AI Open **5**, 208–215 (2024)

15. Xu, N., Liu, X., Li, Y., Zong, G., Zhao, X., Wang, H.: Dynamic event-triggered control for a class of uncertain strict-feedback systems via an improved adaptive neural networks backstepping approach. IEEE Trans. Autom. Sci. Eng. **22**, 2041–2050 (2024)
16. Zheng, N., Wen, J., Qu, J., Li, X., Chen, H.: Robust graph neural network against poisoning attacks via transfer learning. In: Proceedings of KDD (2020)
17. Bortolussi, L., Carbone, G., Laurenti, L., Patane, A., Sanguinetti, G., Wicker, M.: On the robustness of Bayesian neural networks to adversarial attacks. IEEE Trans. Neural Netw. Learn. Syst. **36**(4), 6679–6692 (2024)
18. Baltrušaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: a survey and taxonomy. IEEE TPAMI **40**(8), 1–20 (2018)
19. Jain, S., Jain, A.: Real-time detection of malicious urls by utilizing machine learning techniques. In: Proceedings of 2024 International BIT Conference (BITCON), pp. 1–7 (2024)
20. Paxson, V., Sommer, R.: Experiences with network traffic obfuscation. In: Proceedings of NDSS (2011)
21. Srivastava, A., Goyal, P.: Securing cloud workloads with adaptive microsegmentation. In: Proceedings of IEEE Cloud (2020)
22. Gu, T., Dolan-Gavitt, B., Garg, S.: BadNets: identifying vulnerabilities in the machine learning model supply chain. In: Proceedings of IEEE S&P (2019)
23. Choudhary, V., Singh, S., Atrey, S., Kumar, A., Kalita, S.: A custom sandbox for malware threat analysis to safeguard infrastructure. In: Proceedings of 2025 3rd International Conference on Disruptive Technologies (ICDT), pp. 471–475 (2025)
24. Guo, Q., He, Z., Wang, Z.: Assessing the effectiveness of long short-term memory and artificial neural network in predicting daily ozone concentrations in Liaocheng city. Sci. Rep. **15**(1), 6798 (2025)
25. Tian, K., Jiang, Y., Yuan, Z., Peng, B., Wang, L.: Visual autoregressive modeling: scalable image generation via next-scale prediction. In: Proceedings of NeurIPS (2024)
26. Cheng, D., Xu, Z., Jiang, X., Wang, N., Li, D., Gao, X.: Disentangled prompt representation for domain generalization. In: Proceedings of CVPR (2024)
27. Zhang, I., Li, Z., Wang, X.: E2E-MFD: towards end-to-end synchronous multimodal fusion detection. In: Proceedings of NeurIPS (2024)
28. Chen, H.: Low-res leads the way: improving generalization for super-resolution by self-supervised learning. In: Proceedings of CVPR (2024)
29. Kim, S., Park, J.: Dynamic graph learning for polymorphic document analysis. In: Proceedings of ICLR (2024)
30. Zhou, Z., Tao, R., Zhu, J.: Robust document analysis via adversarial syntax-semantic consistency. In: Proceedings of NeurIPS (2024)
31. Liu, Y., Zhang, C.: Unified multimodal document anomaly detection via hierarchical graph diffusion. In: Proceedings of CVPR (2024)
32. Rahman, M.M., Munir, M.: EMCAD: efficient multi-scale convolutional attention decoding for medical image segmentation. In: Proceedings of CVPR (2024)
33. Gupta, A., Li, B.: DocAttack: generating adversarial documents via gradient-based structure manipulation. In: Proceedings of USENIX Security (2024)
34. Zhang, J., Chen, Z.: ReGExplainer: generating explanations for graph neural networks in regression tasks. In: Proceedings of NeurIPS (2024)
35. Christiansen, P., Nielsen, L.N., Steen, K.A., Jørgensen, R.N., Karstoft, H.: DeepAnomaly: combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field. Sensors **16**(11), 1904 (2016)

36. Li, J., et al.: TextShield: robust text classification based on multimodal embedding and neural machine translation. In: Proceedings of USENIX Security (2020)
37. Zara, G., Roy, S., Rota, P., Ricci, E.: AutoLabel: CLIP-based framework for open-set video domain adaptation. In: Proceedings of CVPR (2023)
38. Kumar, A., Choi, B.J., Kuppusamy, K.S., Aghila, G.: Malware attacks: dimensions, impact, and defenses. In: Advances in Nature-Inspired Computing and Applications, ch. 9. Springer (2021)
39. CrowdStrike Inc: 2024 Global Threat Report: Adversary Tradecraft and the Modern Security Landscape. Technical Report (2024)
40. Guha, A., Samanta, D.: Hybrid approach to document anomaly detection: an application to facilitate RPA in title insurance. Int. J. Autom. Comput. **18**(1), 55–72 (2021)
41. Nassif, A.B., Talib, M.A., Nasir, Q., Dakalbab, F.M.: Machine learning for anomaly detection: a systematic review. IEEE Access **9**, 78658–78700 (2021)
42. Samariya, D., Thakkar, A.: A comprehensive survey of anomaly detection algorithms. Ann. Data Sci. **10**(3), 829–850 (2023)
43. Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: a review. ACM Comput. Surv. **54**(2), 1–38 (2021)
44. Han, S., Hu, X., Huang, H., Jiang, M., Zhao, Y.: ADBench: anomaly detection benchmark. In: Proceedings of NeurIPS **35**, 32142–32159 (2022)
45. Xia, X., et al.: GAN-based anomaly detection: a review. Neurocomputing **493**, 497–535 (2022)
46. Zamanzadeh Darban, Z., Webb, G.I., Pan, S., Aggarwal, C., Salehi, M.: Deep learning for time series anomaly detection: a survey. ACM Comput. Surv. **57**(1), 1–42 (2024)
47. Li, Z., Zhu, Y., Van Leeuwen, M.: A survey on explainable anomaly detection. ACM Trans. Knowl. Discov. Data **18**(1), 1–54 (2023)
48. Liu, J., et al.: Deep industrial image anomaly detection: a survey. Mach. Intell. Res. **21**(1), 104–135 (2024)
49. Li, G., Jung, J.J.: Deep learning for anomaly detection in multivariate time series: approaches, applications, and challenges. Inf. Fusion **91**, 93–102 (2023)
50. Liu, Z., Zhou, Y., Xu, Y., Wang, Z.: Simplenet: a simple network for image anomaly detection and localization. In: Proceedings of CVPR (2023)
51. Liu, W., Chang, H., Ma, B., Shan, S., Chen, X.: Diversity-measurable anomaly detection. In: Proceedings of CVPR (2023)
52. Wang, Y., Peng, J., Zhang, J., Yi, R., Wang, Y., Wang, C.: Multimodal industrial anomaly detection via hybrid fusion. In: Proceedings of CVPR (2023)
53. Tang, J., Li, J., Gao, Z., Li, J.: Rethinking graph neural networks for anomaly detection. In: Proceedings of ICML (2022)
54. Shaukat, K., et al.: A review of time-series anomaly detection techniques: a step to future perspectives. Adv. Inf. Commun. 865–877 (2021)
55. Lindemann, B., Maschler, B., Sahlab, N., Weyrich, M.: A survey on anomaly detection for technical systems using LSTM networks. Comput. Industr. **131**, 103498 (2021)
56. Akcay, S., Ameln, D., Vaidya, A., Lakshmanan, B., Ahuja, N., Genc, U.: Anomalib: a deep learning library for anomaly detection. In: Proceedings of ICIP, pp. 1706–1710 (2022)
57. Choi, K., Yi, J., Park, C., Yoon, S.: Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. IEEE Access **9**, 120043–120065 (2021)
58. Guo, H., Yuan, S., Wu, X.: LogBERT: log anomaly detection via BERT. In: Proceedings of IJCNN, pp. 1–8 (2021)

59. Yang, Y., Zhang, C., Zhou, T., Wen, Q., Sun, L.: DCdetector: dual attention contrastive representation learning for time series anomaly detection. In: Proceedings of KDD, pp. 3033–3045 (2023)
60. Garg, A., Zhang, W., Samaran, J., Savitha, R., Foo, C.S.: An evaluation of anomaly detection and diagnosis in multivariate time series. IEEE Trans. Neural Netw. Learn. Syst. **33**(6), 2508–2517 (2021)
61. Taranto, D., Buchanan, M.T.: Sustaining lifelong learning: a self-regulated learning (SRL) approach. Discourse Commun. Sustain. Educ. **11**(1), 5–15 (2020)
62. Zhang, J.: Interval Parsing Grammar For File Format Specification (2021)
63. Li, L.W., Eakman, G., Garcia, E.J.M., Atman, S.: Accessible formal methods for verified parser development. In: Proceedings of IEEE SPW, pp. 142–151 (2021)
64. Woods, W.: RL-GRIT: reinforcement learning for grammar inference. In: Proceedings of IEEE SPW, pp. 171–183 (2021)
65. Wyatt, P.: Work in progress: demystifying PDF through a machine-readable definition. In: Proceedings of IEEE SPW (2021)
66. Le, T.D., Le-Dinh, T., Uwizeyemungu, S., Doan, X.H., Dinh, T.D.: SafeDocs: a machine learning-based framework for malicious PDF detection tailored for SMEs. In: Proceedings of RIVF, pp. 295–300 (2023)
67. Menon, P.H., Woods, W.: Corpus-wide analysis of parser behaviors via a format analysis workbench. In: Proceedings of IEEE SPW, pp. 209–218 (2023)
68. Parkour, M.: Contagio PDF Malware Corpus. Dataset v3, vol. 2 (2024)
69. Feng, Y., Xu, J., Weymouth, L.: University blockchain research initiative (UBRI): boosting blockchain education and research. IEEE Potentials **41**(6), 19–25 (2022). https://doi.org/10.1109/MPOT.2022.3198929