Next-Event Prediction in Cybercrime Complaint Narratives Using Temporal Event Scene Graphs

Mohammad Saad Rafeeq[1] [a], Narendra Bijarniya[2] and Chandramani Chaudhary[1] [b]

[1]*National Institute of Technology, Calicut, India*

[2]*Birla Institute of Technology and Science, Pilani, India*

Keywords: Temporal Event Scene Graphs, Event Prediction, Cybercrime, Sequence Modeling, BART, T5, GPT-2.

Abstract: Cybercrime complaint narratives encompass complex sequences of criminal activities that challenge conventional sequence modeling techniques. This work introduces a framework that employs dynamic temporal event scene graphs to represent each narrative as an evolving, structured network of entities and events. Our approach converts complaint texts into temporal event scene graphs in which nodes symbolize key entities and edges capture interactions, annotated with their sequential order. This structured representation provides a richer and more intuitive understanding of how cybercrime incidents unfold over time. To forecast missing or forthcoming events, we fine-tune a pre-trained BART model using a masked sequence-to-sequence paradigm.Our experiments are performed on a dataset comprising thousands of real-world cybercrime reports, containing roughly 76,000 distinct event descriptions—a scale that introduces significant sparsity and generalization challenges. Our results demonstrate that while models such as GPT-2 and T5 struggle to capture robust patterns in this diverse domain, the BART-based approach achieves modest yet promising improvements.

1 INTRODUCTION

Predicting the next event in a narrative sequence is an important task to understand and anticipate complex scenarios. In the domain of cyber-crime complaints, accurately forecasting the subsequent step in an attack or crime sequence could assist investigators and automated systems in taking proactive measures. However, this task is challenging due to the unstructured nature of free text reports and the enormous variety of events (Al-Zaidy et al., 2012). Each report may contain various entities, such as victim details, crime details, and additional items (e.g., banks, accounts, cryptocurrencies) along with idiosyncratic event descriptions. Traditional sequence modeling approaches for narrative understanding, such as narrative event chains, capture temporal event sequences centered on a protagonist, but might not fully utilize the rich connections between multiple entities involved in a cybercrime incident.

Graph-based textual representations, on the other hand, can encode complex relationships: for example, multiple entities like (victim, bank, money) might be involved in a single event ("unauthorized transaction without OTP"), and an entity can reappear across events, linking the storyline together.

Our approach is related to previous work on event prediction, often framed as script learning, which has laid the foundation for our method. Early studies introduced the concept of narrative schemas or event chains, where structured representations of common event sequences were learned from text corpora to predict likely subsequent events (Schank and Abelson, 1977), (Chambers and Jurafsky, 2008). More recent research has explored the use of knowledge graphs to capture complex interconnections between events (Li et al., 2018). Inspired by these advancements, our method explicitly represents each narrative as a dynamic graph of entities and actions, enabling the model to capture melianingful state changes and interevent relationships.

We represent each complaint narrative as a **temporal event scene graph (TESG)**, where entities (e.g., "Bank," "victim," "INR 4500") are nodes and actions (e.g., "transaction without OTP") are time-stamped edges. Converting text into these graphs exposes each entity's participation in events, but here we focus on the extracted event sequence to leverage BART for next-event prediction rather than on graph visualization.

[a] https://orcid.org/0009-0008-3049-9531

[b] https://orcid.org/0000-0003-2072-4391

Rafeeq, M. S., Bijarniya, N., Chaudhary and C.
Next-Event Prediction in Cybercrime Complaint Narratives Using Temporal Event Scene Graphs.
DOI: 10.5220/0013647000003967
In *Proceedings of the 14th International Conference on Data Science, Technology and Applications (DATA 2025)*, pages 693-700
ISBN: 978-989-758-758-0; ISSN: 2184-285X

Our backbone is **BART**, a Transformer based encoder decoder pretrained denoising autoencoder. Its bidirectional encoder and autoregressive decoder support next event prediction and missing event interpolation. To our knowledge, this is one of the first applications of temporal event scene graphs to forecast events in cybercrime narratives.We focus on two inference tasks that predict the next event and infer intermediate events while addressing the roles of arguments and coreference preprocessing. In summary, we demonstrate a novel representation of cybercrime narratives as temporal event scene graphs, a BART based sequence generation model for event prediction, and an empirical comparison of next event versus intermediate event masking strategies that advance incident forecasting, leveraging extensive experiments and robust evaluation metrics across diverse datasets.

We describe our modeling methodology in Section 3, present our data set and the graph construction process in Section 4, and outline the planned experiments in Section 5.

2 RELATED WORK

We review three related research directions: (i) event detection, (ii) temporal event prediction, and (iii) crime event detection.

2.1 Event Detection

Early event detection drew on script theory, modeling scripts (Schank and Abelson, 1977). Chambers and Jurafsky (Chambers and Jurafsky, 2008) introduced unsupervised bootstrapping to induce chains. Recent methods integrate neural networks with knowledge graphs for event extraction. Li et al. (Li et al., 2018) propose a narrative event evolutionary graph capturing semantic relationships via graph-based attention. We employ dynamic temporal event scene graphs to enforce sequential ordering and isolate entity transitions in cybercrime narratives.

2.2 Temporal Event Prediction

Temporal event prediction models timing and event order. Du et al. (Du et al., 2016) introduced recurrent marked temporal point processes; Mei and Eisner (Mei and Eisner, 2017) proposed Neural Hawkes for irregular intervals. Kong et al.'s Language-TPP (Kong et al., 2025) adds continuous temporal tokens to Transformers. We instead employ dynamic temporal event scene graphs on noisy text.

2.3 Crime Event Detection

Crime event detection and prediction aid law enforcement. Yang (Yang, 2023) proposed *TransCrimeNet*, fusing text with criminal-network graph embeddings to predict crimes. Zhu and Xie (Zhu and Xie, 2022) introduced a spatiotemporal-textual point process for crime linkage. Khairova et al. (Khairova et al., 2023) applied cross-lingual transfer on parallel corpora for low-resource event extraction. These works underscore graph–text integration, motivating our dynamic temporal event scene graphs with BART.

3 METHODOLOGY

3.1 Construction

Each cybercrime complaint (a free text narrative) is converted into a temportal event scene graph, where nodes represent entities and edges denote actions. In this process, we extract key information by identifying entities such as people (e.g. the victim), organizations (e.g., a bank), and other relevant objects (e.g., amounts of money, devices), as well as the actions or incidents and the order in which they occur. For example, the first event is assigned timestamp T_0, the next T_1, etc. Each distinct action is represented as an edge connecting the involved entities along with its relative timestamp. The final result is a temporal event scene graph that captures how the narrative structure evolves with each successive event.

For example, consider a complaint: "The victim received a phishing email claiming to be from a [Private] Bank. Later, a transaction of Rs 4500 was made from his account without an OTP." From this narrative, we extract the entities $\{\text{Victim}, \text{Bank}, \text{Rs } 4500\}$. The first action, "phishing email", connects the Victim and the Bank (indicating that an email was received from a fraudster impersonating the bank), and is assigned timestamp T_0. The second action, "unauthorized transaction without OTP", involves the victim and the amount Rs 4500 (indicating a funds transfer), and is assigned a timestamp T_1. Figure 1, shows that these events, arranged chronologically, produce an event graph that reflects the initial phishing email between the victim and the bank, followed by the fraudulent transaction involving Rs 4500.

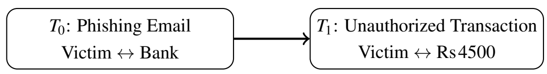

Figure 1: Dynamic TESG illustrating two-step event sequence in a phishing incident.

3.2 Sequence Generation with BART

We linearize the events from each event graph into an ordered sequence based on the timestamp, e.g.

$$[e_1, e_2, \ldots, e_{n+1}],$$

where each e_i represents an event description at timestamp T_{i-1}. We then employ the BART base model (Lewis et al., 2019) as our sequence-to-sequence predictor for the modeling of masked events. BART is particularly suitable here because it was pretrained with a span-masking objective (among others), learning to reconstruct text with missing spans. We fine-tune BART on our data by feeding it sequences of events with certain events replaced by a mask token and training it to generate the missing event text. The fine-tuning is configured in two modes corresponding to our experimental scenarios.

1. **Next-Event Prediction:**
   In this setup, we mask only the final event in the sequence and ask BART to predict it. Formally, given a sequence of events, $e_1, e_2, e_3, \ldots, e_{n+1}$, generated from a complaint description, where e_1 occurs at T_0 and e_{n+1} is the final recorded event at T_n. We create an input where e_{n+1} is replaced by the `[MASK]` token as shown in Figure 2. BART's encoder processes the sequence of past events $e_1, e_2, \ldots, e_n$ along with the mask token, and the decoder is trained to output e_{n+1} (the next event)This is analogous to a conditional next-step prediction task, except that BART, unlike a standard language model, can leverage the bidirectional encoding of the preceding events. The target output is the textual description of the held-out event e_{n+1}. By training on many such sequences, the model learns to infer what typically comes last given the earlier events in cybercrime scenarios (e.g., after "victim receives a phishing email", the next event might be "unauthorized bank transaction").

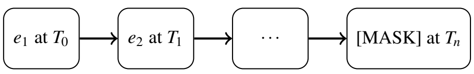

Figure 2: Dynamic event sequence illustrating a masked event at T_n within a sequential narrative.

2. **Intermediate-Event Prediction:**
   Here, we evaluate the model's ability to predict any single event in the sequence when that event is hidden (missing event), using the rest of the events as context. For a sequence

$$[e_1, e_2, e_3, \ldots, e_{n+1}],$$

   occurring at $T_0, T_1, T_2, \ldots, T_n$, we create multiple training examples by masking one event at a time. For instance, $(e_1, \texttt{[MASK]}, e_3, \ldots, e_{n+1})$ with target e_2; $(e_1, e_2, \texttt{[MASK]}, e_4, \ldots, e_{n+1})$ with target e_3; and so on, (including a case masking e_{n+1} as in the first experiment) as shown in Figure 3. BART is trained to fill in the blank in each case. This setup forces the model to use both preceding and following events to predict the missing one, tapping into its sequence infilling capability. It is effectively a form of data augmentation, since a single narrative yields multiple training samples (one for each event masked). During inference, we can similarly mask an event and have the model generate a prediction for what event should be at that position.

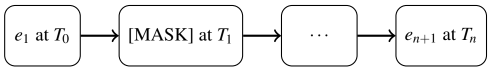

Figure 3: Dynamic event sequence demonstrating the masking of an intermediate event, where the context before and after the masked event is utilized for prediction.

Training BART with these masked event objectives requires a proper formulation of the input-output pair. We concatenate the sequence of event descriptions into a single textual input separating events with a delimiter (**semicolon ';'**), and use a special `[MASK]` token in place of the hidden event's text. The decoder's target is the text of the hidden event. We initialize the model with `Facebook/bart-base` weights and fine-tune for a fixed number of epochs, using a loss function based on the cross-entropy between the generated token sequence and the ground-truth event. Because event descriptions in our data can be a short phrase or a full sentence, we treat each action as a segment of text to be generated. Notably, BART's ability to consider the entire input sequence (with knowledge of both prior and subsequent events in the sequence for intermediate prediction) gives it an advantage over unidirectional models in the second scenario. The model effectively learns a conditional distribution for an event given its surrounding context events.

3.3 Baseline Models

We experimented with two pre-trained models as baselines: GPT-2 and T5. GPT-2 is a decoder-only language model that generates text left-to-right. We fine-tuned a GPT-2 model (117M parameters) to predict the next event given the preceding events as input. However, GPT-2 struggled with the highly variable event vocabulary, often generating generic or incoherent outputs when confronted with rare or complex events. Although it achieved a semantic similarity of 49.8%, GPT-2 failed to produce robust performance in terms of Hit and other metrics.

T5, an encoder-decoder transformer pre-trained on a wide variety of text-to-text tasks, was fine-tuned in a similar manner—treating the task as text generation, where the input is the sequence of prior events and the output is the next event. T5-generated outputs exhibited a semantic similarity of only 9.25%, making it less reliable. Consequently, both GPT-2 and T5 underperformed compared to BART. BART's superior performance can be partly attributed to its pretraining objective, which included an in-filling task (i.e., predicting masked spans), making it inherently well suited for our formulation. Therefore, our methodology focuses on BART for the reported results.

4 DATASET

We compiled a data set of cybercrime complaint narratives filed on the official online reporting portal (`https://cybercrime.gov.in/`), which serves as the primary source of our data. Each entry in the data set is a textual description of an incident written by the victim or a police transcript thereof. These narratives typically range from a few sentences to a few paragraphs detailing how the crime unfolded. From an initial pool of reports, we filtered and segmented the text to isolate discrete events and identify entities, yielding a collection of structured event sequences.

Examples to Illustrate the Data:

Consider the following example narrative.

"On [Date], a fraudulent transaction of 4500 Rs. occurred approximately [Time] from a Private Bank account, without the victim receiving any OTP or message regarding the transaction."

We extract entities, events, and a temporal event scene graph from this narrative.

- **Entities:** `['Private Bank', '4500 Rs.', 'Victim']`
- **Events:** `['fraudulent transaction of 4500 Rs.', 'no OTP received', 'no message received']`

Statistics: The processed data set contains approximately $N = 11{,}500$ narratives (reports). The number of events per narrative varies , while some reports record only a single event, the most detailed ones include up to 23 events. Importantly, the vocabulary of unique events is extremely large, on the order of 76,000 unique event descriptions. This indicates that authors of the reports seldom use identical phrasing, and many actions/events are very specific (e.g., "user's credit card limit was increased without authorization" might appear only once).

Table 1: Extracted data from example narrative.

Entities	`['Private Bank', '4500 Rs.', 'Victim']`
Events	`['fraudulent transaction of 4500 Rs.', 'no OTP received', 'no message received']`
Scene Graph	`{ 'nodes': [{ 'id': 'Private Bank' }, { 'id': '4500 Rs.' }, { 'id': 'Victim' }], 'edges': [{ 'source': 'Private Bank', 'target': '4500 Rs.', 'relationship': 'fraudulent transaction of', 'timestamp': 'T0' }, { 'source': '4500 Rs.', 'target': 'Victim', 'relationship': 'no OTP received', 'timestamp': 'T1' }, { 'source': '4500 Rs.', 'target': 'Victim', 'relationship': 'no message received', 'timestamp': 'T2' }] }`

4.1 Preprocessing

We preprocess raw multilingual cybercrime narratives by prompting a large language model to produce unified English summaries that preserve key events, entities, and relations, thereby standardizing input for downstream analysis. Named Entity Recognition then extracts persons, organizations, and other salient entities, while dependency parsing isolates event predicates and arguments, yielding *entity–action–entity* triples with synthetic timestamps derived from narrative order. These actions serve as event labels. We do not yet merge semantically similar actions (e.g., "withdrew money from ATM" vs. "cash withdrawal at ATM"), leaving sparsity reduction to future work.

Sparsity and Pattern Learning: The sheer number of unique events means that most of the event sequences in the training set are nearly unique in their exact surface form. Therefore, the model cannot rely on memorizing frequent event diagrams or templates; it must learn higher-level patterns or analogies. For example, even if "phishing email $\rightarrow$ bank transaction" appears only once, the model might learn a broader pattern that an event involving a social engineering attack (like phishing) is often followed by a financial fraud event. Our hope is that by representing events in context with their entities, the model can learn latent connections (e.g., if the same bank entity appears in two events, they might be related). We acknowledge that without explicit semantic clustering, this is a difficult task; the model must generalize from very few examples per event type. This characteristic of the data set underscores the need for a structured approach and informs our decision to compare different modeling strategies (standard language models vs. event-based BART).

Train/Validation/Test Split: We split the data set into training, validation, and test sets at the narrative level. The validation set comprises 25% of the entire dataset, while the remaining 75% is divided into training and test sets in an 80:20 ratio (resulting in approximately 60% training and 15% test). This strategy ensures each narrative appears in only one set, reducing train-test overlap.

5 EXPERIMENTS AND RESULTS

5.1 Next-Event Prediction

In this, we evaluate the prediction of the next event using a fine-tuned BART model. Here, the model is provided with a dynamic temporal event scene graph constructed from a cybercrime narrative up to time t—that is, it observes all events $e_1, e_2, \ldots, e_{n-1}$ prior to the incident to be predicted. The objective is to generate the subsequent event e_n at time $t+1$. Unlike approaches that benefit from both preceding and succeeding context (as in intermediate event inpainting), this next-event prediction task is constrained by the fact that only historical events—with the last observed event e_{n-1} serving as the immediate cue—are used for prediction. This limited context typically results in fewer generated examples and often leads the model to produce degenerate outputs, such as simply echoing the final observed event, thereby failing to introduce the necessary novelty for e_n.

To address these challenges, BART is fine-tuned to generate a descriptive textual output for the predicted event. This output is subsequently parsed into its structured components—source, target, relationship, and timestamp. The evaluation protocol is twofold. Firstly, we utilize text similarity metrics: ROUGE scores quantify the lexical overlap between the generated and true event descriptions, while an embedding-based semantic similarity metric assesses the alignment in meaning between them. Secondly, we employ rank-based metrics by reporting Hit@K (for $K = 1$ to 5), which indicate whether the ground-truth event appears among the top K model predictions. Table 2 summarizes these overall performance metrics for next-event prediction, including Semantic Similarity, ROUGE-1, ROUGE-2, ROUGE-L, and Hit@K values.

Table 2: Performance on Next-Event Prediction using BART.

Metric	Score
Semantic Similarity	0.5459
ROUGE-1	0.4283
ROUGE-2	0.2199
ROUGE-L	0.41695
Hit@1	0.0094
Hit@2	0.0135
Hit@3	0.0165
Hit@4	0.0183
Hit@5	0.0193

Additionally, we compute the accuracy for each individual event component to better understand the model's ability to correctly identify the roles and the temporal ordering within the event. Table 3 provides a detailed analysis of the first predicted event (i.e., Hit@1), reporting component-wise accuracies for source, target, relationship, and timestamp.

This experimental setup is inherently challenging because predicting an unseen future event using solely the historical sequence—where only e_{n-1} directly informs the prediction of e_n—provides less contextual information compared to settings that incorporate both past and future cues. Consequently, the limited number of training examples and the absence of forward-looking indicators lead to lower overall performance, particularly in terms of Hit@K scores and the accuracy of predicting the target and relationship components.

Table 2 summarizes the overall performance metrics for next-event prediction, ordered as: Semantic Similarity, ROUGE-1, ROUGE-2, ROUGE-L, followed by Hit@1 to Hit@5. The average semantic similarity between the predicted and ground-truth events is 0.5459, indicating that the model output is somewhat related in meaning to the intended events. The ROUGE scores are moderate, with ROUGE-1 at 0.4283, ROUGE-2 at 0.2199, and ROUGE-L at 0.41695, suggesting partial lexical overlap between predicted and actual events. The Hit metrics show that Hit@1 is only 0.0094 and Hit@5 is 0.0193, meaning that the correct (exact) event appears as the top prediction in less than 1% of cases and within the top five predictions in fewer than 2% of cases.The prediction components analysis (Table 3) reveals that while the model correctly identifies the source 49.72% of the time, it struggles with predicting the target (11.59%) and the relationship (4.02%), even though the timestamp is correctly predicted in 91.76% of instances.

For a more detailed analysis of the first predicted event (i.e. Hit@1), Table 3 reports the component-wise accuracies.

Table 3: Prediction Components Analysis for Hit@1 in Next-Event Prediction.

Component	Accuracy
Source Accuracy	0.4972
Target Accuracy	0.1159
Relationship Accuracy	0.0402
Timestamp Accuracy	0.9176

Table 4: Performance on Intermediate Event Inpainting using BART.

Metric	Score
Semantic Similarity	0.6073
ROUGE-1	0.4968
ROUGE-2	0.2920
ROUGE-L	0.4844
Hit@1	0.0287
Hit@2	0.0417
Hit@3	0.0489
Hit@4	0.0535
Hit@5	0.0559

5.2 Intermediate Event Inpainting

Here, we evaluate intermediate event inpainting using the same BART architecture. In this task, an event in the middle of a narrative is hidden, and the model must infer this missing event given the surrounding context (all prior events up to time t and subsequent events after time $t+1$). The model is provided with a narrative with a 'gap' and is asked to fill that gap with a plausible event that connects logically to both the preceding and the following events. We fine-tuned the model on this inpainting task, expecting that the additional future context would guide the generation of the missing event.

Table 4 shows the overall performance metrics for the inpainting task, ordered as: Semantic Similarity, ROUGE-1, ROUGE-2, ROUGE-L, followed by Hit@1 to Hit@5. The table shows that the average semantic similarity in intermediate event inpainting is 0.6073, which is higher than in the Next-Event Prediction—indicating that the painted events are semantically closer to the true events. The ROUGE scores also show improvement, with ROUGE-1 at 0.4968, ROUGE-2 at 0.2920, and ROUGE-L at 0.4844. The Hit metrics reveal that Hit@1 is 0.0287 and Hit@5 is 0.0559, signifying an increased likelihood of the correct event appearing in the top predictions when both past and future contexts are available.

Similarly, Table 5 details the prediction components for the first predicted event (Hit@1) in the inpainting experiment. The prediction components

Table 5: Prediction Components Analysis for Hit@1 in Intermediate Event Inpainting.

Component	Accuracy
Source Accuracy	0.5723
Target Accuracy	0.2261
Relationship Accuracy	0.0876
Timestamp Accuracy	0.9768

analysis (Table 5) indicates enhanced performance in event component prediction, with source accuracy at 57.23%, target accuracy at 22.61%, relationship accuracy at 8.76%, and timestamp accuracy at 97.68%.

5.3 Comparative Analysis of Results

Comparing the results of the next event prediction and intermediate event inpainting side by side, we observe a consistent improvement in all metrics when performing intermediate event inpainting instead of the next event prediction. The semantic similarity increases from 0.5459 to 0.6073, and ROUGE scores are higher in intermediate event inpainting (e.g., ROUGE-1 improves from 0.4283 to 0.4968 and ROUGE-L from 0.41695 to 0.4844). The Hit metrics approximately triple, with Hit@1 increasing from 0.0094 to 0.0287 and Hit@5 from 0.0193 to 0.0559. These improvements suggest that providing both preceding and subsequent context allows the model to generate missing events more accurately.The prediction component accuracies are also notably higher in intermediate event inpainting, particularly for the target and relationship, which roughly double in accuracy compared to Next-Event Prediction, while the source and timestamp predictions show moderate improvements.

5.4 Qualitative Analysis and Error Discussion

Table 6 shows both successes and failure modes. The most frequent issue is *entity confusion*: in the third row the model predicts "phone – priced at → Rs 38 000" at T1, whereas the gold event is "suspect – claimed to be selling on → MARKETPLACE," misassigning subject and predicate. A second pattern is loss of fine-grained detail: in the second row it outputs "individual – made payment via → PAYMENT" instead of the precise "individual – completed payment." Yet, with unambiguous context the model can be exact, as in the first row where it perfectly recovers "hacker – sent inappropriate messages → friend" at T1. These observations align with moderate ROUGE and semantic-similarity scores: predictions capture the event frame but often slip on roles or verbs. Prior experiments with

Table 6: Qualitative examples for Intermediate Event Inpainting (three illustrative cases). Each row shows the masked input sequence, the model's prediction, and the ground-truth event. Entity tokens replace personal information and proper nouns (e.g., SOCIAL, ECOMMERCE, PAYMENT).

Input (with [MASK])	Predicted Event	Target Event
Complete: account - account hack → hacker at T0 ; [MASK] ; hacker - sent nude photos → friend at T2 ; victim - seeking assistance regarding the incident → assistance at T3	hacker - sent inappropriate messages → friend at T1	hacker - sent inappropriate messages → friend at T1
Complete: individual - encountered → SOCIAL at T0 ; SOCIAL - misleading advertisement for a sale → ECOMMERCE at T1 ; individual - clicked link to fraudulent website → ECOMMERCE at T2 ; individual - purchased → MOBILE at T3 ; [MASK] ; PAYMENT - payment amount → Rs 1999 at T5 ; individual - transaction identified as fraudulent → fraudsters at T6 ; individual - requested action against → fraudsters at T7 ; individual - requested to freeze associated accounts → fraudsters at T8	individual - made payment via → PAYMENT at T4	individual - completed payment → PAYMENT at T4
Complete: suspect - sold on → phone at T0 ; [MASK] ; suspect - manipulated into paying → victim at T2 ; victim - paid total amount of → Rs 38000 at T3 ; suspect - requested additional payment of → Rs 11000 at T4 ; suspect - falsely claimed penalty to → victim at T5 ; suspect - claimed to be in → MILITARY at T6	phone - priced at → Rs 38000 at T1	suspect - claimed to be selling on → MARKETPLACE at T1

GPT-2 and standard T5 were even less accurate, likely due to sparse event inventories and long-range dependencies. BART, guided by dynamic temporal scene graphs, offers a stronger inductive bias but still requires richer role-aware encodings and semantic clustering to curb entity confusion and sparsity.

6 DISCUSSIONS

The experimental findings highlight key insights into modeling cybercrime narratives through event prediction. Quantitatively, the intermediate event inpainting task demonstrates a clear advantage over next-event prediction. The availability of both preceding and subsequent context yields higher semantic similarity and improved ROUGE scores, as well as significantly enhanced Hit@K metrics. These improvements confirm that additional future context helps mitigate challenges inherent to unidirectional prediction, such as the limited cue provided by the immediate past event.

Qualitatively, the analysis reveals critical error patterns that impact overall performance. In the next-event prediction task, a notable issue observed with models such as T5 is the tendency to simply repeat the final observed event. Additionally, there is a prevalent confusion between the roles of entities, specifically a misassignment of source and target, which is particularly detrimental. In contrast, the inpainting approach benefits from a more robust context that reduces these errors, although difficulties in precisely capturing complex relationships remain. These challenges are further compounded by the lack of unique event patterns in cybercrime data, highlighting the importance of effective long-range dependency modeling. These observations underscore the importance of structured inputs, such as dynamic temporal event scene graphs, and suggest that future work should focus on refining entity role differentiation and exploring hybrid architectures. Such advancements may lead to further improvements in the capture of the nuances of cybercrime narratives and the improvement of prediction accuracy.

7 CONCLUSION

We presented a dynamic temporal event scene graph approach for next-event prediction in cybercrime narratives. By converting free-text reports into event sequences, we harnessed pretrained BART to predict missing events. We compared next-event prediction and intermediate-event inpainting. Quantitative evaluations (Hit@K, ROUGE, semantic similarity) show next-event prediction remains challenging ($Hit@1 < 1\%$), while inpainting leveraging both prior and subsequent context triples. Qualitative analysis revealed issues like event repetition and entity confusion. Future work will address these via event clustering, entity resolution, event standardization, and external knowledge integration. Our framework also boosts interpretability and situational awareness by detecting subtle narrative shifts, offering actionable insights for law enforcement.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the Indian Space Research Organisation (ISRO) for supporting this work financially.

REFERENCES

Al-Zaidy, R., Fung, B. C., Youssef, A. M., and Fortin, F. (2012). Mining criminal networks from unstructured text documents. *Digital Investigation*, 8(3-4):147–160.

Chambers, N. and Jurafsky, D. (2008). Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. (2016, August). Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564.

Hou, M., Hu, X., Cai, J., Han, X., and Yuan, S. (2022). An integrated graph model for spatial–temporal urban crime prediction based on attention mechanism. *ISPRS International Journal of Geo-Information*, 11(5):294.

Khairova, N., Mamyrbayev, O., Rizun, N., Razno, M., and Galiya, Y. (2023). A parallel corpus-based approach to the crime event extraction for low-resource languages. *IEEE Access*, 11:54093–54111.

Kochakarn, P., De Martini, D., Omeiza, D., and Kunze, L. (2023, May). Explainable action prediction through self-supervision on scene graphs. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1479–1485. IEEE.

Kong, Q., Zhang, Y., Liu, Y., Tong, P., Liu, E., and Zhou, F. (2025). Language-TPP: Integrating Temporal Point Processes with Language Models for Event Analysis. *arXiv preprint arXiv:2502.07139*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Li, Y. and Liu, W. (2022). Sudden event prediction based on event knowledge graph. *Applied Sciences*, 12(21):11195.

Li, Z., Ding, X., and Liu, T. (2018). Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081*.

Mei, H. and Eisner, J. M. (2017). The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, 30.

Roshankar, R. and Keyvanpour, M. R. (2023, November). Spatio-temporal graph neural networks for accurate crime prediction. In *2023 13th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 168–173. IEEE.

Schank, R. C. and Abelson, R. P. (1977). *Scripts, plans, goals and understanding: an inquiry into human Knowledge structures*. Lawrence Erlbaum, Oxford.

Slam, M. I. K., Saifuddin, K. M., Hossain, T., and Akbas, E. (2024, December). Dygcl: Dynamic graph contrastive learning for event prediction. In *2024 IEEE International Conference on Big Data (BigData)*, pages 559–568. IEEE.

Xia, L., Huang, C., Xu, Y., Dai, P., Bo, L., Zhang, X., and Chen, T. (2022). Spatial-temporal sequential hypergraph network for crime prediction with dynamic multiplex relation learning. *arXiv preprint arXiv:2201.02435*.

Yang, C. (2023). TransCrimeNet: A transformer-based model for text-based crime prediction in criminal networks. *arXiv preprint arXiv:2311.09529*.

Zhu, S. and Xie, Y. (2022). Spatiotemporal-textual point processes for crime linkage detection. *The Annals of Applied Statistics*, 16(2):1151–1170.