Distilbert-gnn: a Powerful Approach to Social Media Event Detection

Asres Temam Abagissa
asres_2121cs19@iitp.ac.in

Indian Institute of Technology Patna

Shruti Saxena
Indian Institute of Technology Patna

Joydeep Chandra
Indian Institute of Technology Patna

Research Article

Keywords: event detection, graph neural networks, distilBERT, social media

Posted Date: April 4th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4193412/v1

Additional Declarations: No competing interests reported.

DistilBERT-GNN: A Powerful Approach to Social Media Event Detection

Asres Temam Abagissa, Shruti Saxena, Joydeep Chandra

Computer Science & Engineering, IIT Patna, Bihta, Patna, 801106, Bihar, India.

Contributing authors: asres_2121cs19@iitp.ac.in; shruti_2021cs31@iitp.ac.in; joydeep@iitp.ac.in;

Abstract

Finding events actively discussed locally or globally is a significant problem when mining social media data streams. Identifying such events can serve as an early warning system in an event such as an accident, a protest, an election, or other breaking news. However, with the massive volume of social media feeds streaming, early detection of such events is inherently complex. Despite the advances in social media event detection, existing methods often struggle with the dynamic nature of social media, the volume and velocity of data, and the ambiguity in user-generated content. On the contrary, several relational aspects are present in social media that, if suitably handled and exploited, can improve detection performance. To mitigate these challenges, we propose "DistilBERT-GNN," an incremental event detection framework that leverages DistilBERT and Graph Neural Networks (GNNs). By integrating DistilBERT's real-time contextual understanding with GNNs' ability to capture evolving relationships in social media networks, our framework aims to detect and track events as they emerge and evolve. We assess the effectiveness of our approach through comparative analysis against various state-of-the-art event detection methods on a real-world Twitter dataset. Our experimental result demonstrates that DistilBERT-GNN outperforms the baselines with NMI, AMI, and ARI metrics by 0.72, 0.53, and 0.24, respectively.

Keywords: event detection, graph neural networks, distilBERT, social media

1 Introduction

Social media platforms like Twitter, Facebook, Weibo, and WeChat have gained importance in everyday life (Cao et al., 2021). Because of the expanding popularity of smartphones and the availability of cheap data services, people are engaging more on social media platforms to report significant incidents around them (Dwivedi et al., 2021). A key component of modern digital discourse analysis is social media event detection, an essential tool for identifying noteworthy events within the constant stream of online evaluation (Luo et al., 2021). This procedure entails locating and keeping track of events actively discussed on social media platforms, from elections and other breaking news to accidents and demonstrations (Mredula et al., 2022). In today's fast-paced digital world, detecting an event is vital. It provides an early warning system for various scenarios and a unique insight into public perception and reaction to developing events.

In contrast to conventional news outlets, social media offers an ever-changing and unorganized compilation of user-generated content, with crucial information sometimes lost in the sea of information. The difficulty is sorting through this massive, constant stream of data to identify events in real-time, frequently before traditional media channels report on them (Zhou et al., 2023). Event Detection (ED) in social media intends to turn this chaotic stream into a cohesive narrative by utilizing cutting-edge techniques like GNN and machine learning algorithms (Peng et al., 2019).

Although a broad spectrum of works has been dedicated to using social media event detection (Cao et al., 2021), handling such data has three significant issues.

- **Vocabulary gap**: The language used in social media postings undergoes constant changes, making it challenging to keep up with the essential content. Words possess lexical expansions and equivalents, leading to a discrepancy in vocabulary. A conventional approach involves referring to a dictionary. However, this method is unsuitable for social media posts. Several terms, such as slang and abbreviations often seen in tweets, are absent from dictionaries (Kolajo et al., 2022).
- **Short and unstructured tweets**: This issue makes it difficult to interpret the semantics of short informal writings on Twitter (Lilleberg et al., 2015; Murshed et al., 2023) . The shortness of tweets sometimes results in inadequate context, making determining a post's mood or genuine intention difficult.
- **Involvement of events**: The continuous and growing flow of social media data is essential for several fields, such as disaster management, public security, and trend analysis. Although classic event detection approaches (Aggarwal & Subbian, 2012; L. Hu et al., 2017; Ozdikis et al., 2017; K. Zhang et al., 2007), are successful in certain situations, they often face difficulties in adapting to the continuous flow of social media data (Rebuffi et al., 2017) and the changing attributes of new events.

Researchers (Graves & Schmidhuber, 2005) utilize representation learning models such as BiLSTM to capture bidirectional, long-term dependencies between words to handle the vocabulary gap in social media posts. HeteRs (Sun et al., 2023) resolves event involvement, formulate the historical record into a heterogeneous graph, and apply a random walk with restarts and **M**ultivariate **M**arkov **C**hain (MMC) to find probabilistic similarity between the events. QSGNN (Ren et al., 2022), along with handling involvement of events by incorporating diversity-based active learning principles, also handles unstructured tweets by capturing the semantic relationships. Word2Vec and GloVe (Nguyen et al., 2019) translate words into dense vectors that capture semantic relationships, helping to bridge the vocabulary gap by understanding similar meanings despite different word choices.

While handling the above problems, some challenges in incremental event detection still exist. These challenges are a) **Data volume and velocity**; social media platforms generate an overwhelming volume of data rapidly. This influx of information demands event detection techniques that can process and adapt to new data points incrementally without compromising efficiency or accuracy. b) **Dynamic event characteristics**; events on social media platforms can exhibit diverse characteristics that evolve over time. The dynamic nature of events, such as changing keywords, hashtags, and context, presents a challenge for traditional event detection methods that rely on predefined patterns or static models. and c) **Noise and irrelevance**; social media's dynamic nature often leads to noise, irrelevant content, and false positives. Incremental event detection methods must address these issues to identify newly emerging events and accurately update existing ones.

To mitigate the dynamic engagement of events in social media, we propose "DistilBERT-GNN," an incremental, adaptable, and contextually aware event detection framework that harnesses the power of DistilBERT and GNNs. DistilBERT, a distilled version of the BERT model, handles the data's volume, velocity, and noise along with a real-time contextual understanding of textual content. By integrating DistilBERT with GNNs, we aim to capture the evolving relationships and interactions within the social media network, allowing for detecting and tracking events as they emerge and evolve (Danday & Murthy, 2022). Our model features real-time adaptation, contextual understanding, noise reduction, and refinement.

DistilBERT-GNN uses word embeddings to discover term synonyms for term extension, which are learned from millions of tweets and imitate the current language and its usage on Twitter (Velickovic et al., 2019). A pre-trained word2vec embedding is used and fed into DistilBERT-GNN to classify and enhance the precision and efficacy of Twitter data. Likewise, the features map collects both local and global statistical characteristics and trains on non-zero items in a co-occurrence matrix. This data serves as input to the model to learn words, sub-words, phrases, and their relationships incrementally. Incremental learning allows learning from sequential data (Slim et al., 2022). When a new message arrives, the model updates the homogeneous message graph and labels it with the specific event. Thus, identifying class depends on the similarity between new and current messages. The learning process does not remove all obsolete messages but instead uses messages that closely resemble the newly arrived message (Yuan et al., 2022).

Our novelty lies in skillful DistilBERT-driven node filtering, which optimizes node selection while mitigating noise in social media data, further supported by empirical confirmation of the DistilBERT-GNN model's effectiveness. We evaluate our model based on several performance metrics to validate its performance by examining how closely its clustering outcome aligns with the real cluster of events. This approach better enables us to understand which cluster we should classify the event. To summarize our main contributions:

- The proposed DistilBERT-GNN is efficiently applied to filter and choose significant nodes.
- The filtering technique also removes noise from data derived from social media platforms.
- The proposed DistilBERT-GNN model's effectiveness is validated through experimental results.

The subsequent sections are structured in the following manner. In Section 2, we review the literature on event detection. Section 3 provides problem statements. Our methodology is explained in Section 4. Section 5 briefly discusses the experimental evaluation, and the results are discussed in Section 6. Finally, Section 7 presents the study's conclusion.

2 Literature Review

The event detection process on social media ED employs numerous methods. These methods are pivotal to detecting events on social media. They are discussed in detail, providing a comprehensive understanding of their individual and collective impact on the state-of-the-art. The current methods for detecting social media events can be classified as either feature-pivot (Kleinberg, 2002) or document-pivot models (Y. Yang et al., 1998).

Feature-pivot models analyze word distributions and group them to detect events. Researchers identify events based on grouping bursty phrases (Mathioudakis & Koudas, 2010). However, this approach needs a more robust probabilistic basis and only focuses on event detection. It does not consider critical event-related posts or influential users participating in this category of events. Wavelet analysis (Ilina et al., 2012) has utilized frequency-based raw signal analysis of words to generate signals for tokens and filter out insignificant tokens by evaluating their related signal auto-correlations. This kind of approach detects the events via a modular-based graph partitioning strategy. Regardless, it solely focuses on detection and does not consider essential postings or influential spreaders, leading to real-time challenges in monitoring and controlling events in (Ren et al., 2022). Some classification methods are categorized as feature-pivot based and can detect irrelevant or unconfirmed tweets by using such as eXtreme Gradient Boosting (XGBoost) (Gumaei et al., 2022), Multi-Layer Perceptron (MLP) (Mredula et al., 2022), and Naive Bayes (NB) (Sufi & Khalil, 2022). Compared to others, XGBoost and MLP outperform other approaches, particularly for opinion identification in response to a specific event (Bravo-Marquez et al., 2014).

The document-pivot approach for event detection involves grouping texts based on their semantic similarities. Researchers (Wang et al., 2019) proposed a model based on probabilistic latent semantic analysis (pLSA) that uses document pivoting to identify related bursty patterns across multiple text streams. Additionally, researchers apply

Logistic Regression (LR) techniques to monitor disasters from social media posts (Olaleye et al., 2022). Document-pivot-based classification on ED. The paper (AlSumait et al., 2008) suggested using an LDA topic model to model the topics in text streams and tracking their evolution using an evolution matrix. Similarly, (L. Hu et al., 2017) suggested the LDA model for identifying bursty global events that use the same pivoting principle. Researcher (Ozdikis et al., 2017) introduced a topic model called Bursty Event dEtection (BEE) that discovers a rapidly new event by modeling it. Furthermore, all event detection algorithms have flaws in automatically detecting the number of topics and identifying linked central posts and significant spreaders participating in these significant events. Models for event detection that rely on document pivots have a history of application in programs such as Topic Detection and Tracking (TDT)(Fedoryszak et al., 2019; K. Zhang et al., 2007). TDT solutions provide general insights and guidelines for event detection. Nonetheless, irrelevant posts and the volume of regular users make these algorithms inadequate for pinpointing important events within the vast social media data.

To summarize, the previously mentioned models have shortcomings in event detection for several reasons. Firstly, one must adequately utilize micro-blogging features such as post relationships to determine their significance. Instead, these approaches focus on identifying events by grouping words, as seen in (AlSumait et al., 2008; Rebuffi et al., 2017) and Second, present methods (Liu et al., 2020; Ozdikis et al., 2017) are solely concerned with event detection and not finding critical postings. To address the above-mentioned issues, we offer a GNN-based approach in this study, which automatically determines the number of themes and recognizes significant posts among many noisy posts. Our proposed model uses post relationships with GNN representation to create semantic-based event detection. In conclusion, our proposed techniques fill the shortcomings of the current methods, leading to enhanced precision and effectiveness in identifying important events.

3 Problem statement and notation

We introduce a few terminologies, the primary notations used in this study (in Table 1), and a formal definition of the problem.

Definition 1. *A **social stream** is a continuous and time-based ordered sequence of social message blocks, represented as $S = M_0, ..., M_{i-1}, M_i...$, where each block of the message M_i contains all the messages that arrived at the specific period $[t_1, t_{i+1})$. M_i is a set of individual messages, m_j, where a message block $|M_i| = \{m_j | 1 \leq j \leq |M_i|\}$ and $|M_i|$ denotes overall count of the messages within the block. Each social message, denoted as m_j, consists of a set of keywords d_j, users (sender and mentioned) denoted as u_j, and time stamp denoted as t_j.*

Definition 2. *A **social event**, denoted as e, is a collection of interrelated social messages associated with an ordinary real-world happening. It is assumed that each social message is related to only one event. The social event e comprises a collection of social messages, denoted as $e = \{m_i | 1 \leq i \leq |e|\}$.*

Definition 3. *When provided with a block of the message M_i, a **social ED** algorithm is trained to learn a model $f(M_i; \theta) = E_i$, where E_i is a set of events present either* *within M_i or new, denoted as $\{e_k | 1 \leq k \leq |E_i|\}$. The parameter θ represents the variable inputs of the model f.*

Definition 4. *The problem for* ***incremental social ED*** *to create a series of ED models $(f_0, ..., f_{t-w}, f_t, ...)$ based on a given social stream (S), which are trained to detect events in the message block in $M_i | (t+1) \leq i \leq (t+w)$, the model $f_t(M_i; \theta_t, \theta_{t-w}) = E_i$ identifies a list of events in M_i, where $E_i = e_k | 1 \leq k \leq |E_i|$. The model's window size for updating is represented by w, θ_t is the parameter of f_t, and θ_{t-w} is the parameter of f_{t-w}. Here, f_t relies on the knowledge of its prior f_{t-w} and uses θ_{t-w}. However, since there is no predecessor for the initial model, f_0 is the first model and does not depend on any previous model.*

Table 1: Description of notations

Notation	**Description**
S;M	Social stream; message block
E	Set of event
t	Time frame
G	Message graph
o;e;u	Words; Named entities; Users
l;L	Number of GNN layers; total number of layers
h_{m_i}	The final representation of m_i
$h_{m_l}^{(l)}$	Representation of m_i at the l^{th} layer
m_{i+}	A message that clustered to class of m_i
m_{i-}	A message that is not clustered to class of m_i
b	The number assigned to a particular mini-batch.
B	The overall count of mini-batches.
$\{m_b\}$	A messages set that belong to the mini-batch b.
w	Window size
w_{mk}	The adjacency matrix between m and node k.
A	Adjacency matrix
X	Initial feature vectors of G.
$\varepsilon(X, A)$	GNN encoder that creates message embeddings of G
N	The overall count of messages within G
$c_1, ..., c_l$	The count of neighboring nodes chosen at each layer
s	The summary vector of G
$\mathcal{L}_t$	Triplet loss
$\mathcal{L}_p$	Global-local pair loss

4 Methodology

Our methodology comprises six phases, briefly explaining the overall work of the model. The pre-processing phase is explained in Section 4.1, and heterogeneous social message modeling is described in Section 4.2, briefly explaining heterogeneous graph generation. Incremental message embedding is explained in Section 4.3, and the node filtering technique in Section 4.4. Message clustering in Section 4.5, and lastly, the loss function is explained in Section 4.6.

We use a technique that uses the DistilBERT model embedded in GNN to capture the relational dynamics expressed in social media posts rapidly. The DistilBERT-GNN framework is designed to identify and rank relevant information nodes to improve the detection of major events by considering the complex connections between context and relationships in social media communication. This simple approach aims to increase detection accuracy and adaptability in the fast-paced social media platform.

4.1 Pre-processing Phase

The pre-processing phase encompasses steps to filter noise and irrelevant information. These steps involve the removal of URLs, elimination of stop words, extraction of meaningful tokens, removal of special characters, and application of stemming. An integral part of this phase is the Named Entity Recognition (NER), which identifies individuals, locations, and organizations.

While NER focuses on recognizing specific entities, the remaining steps eliminate extraneous elements, thus converting unstructured data into a structured format conducive to the subsequent phases. Additionally, this phase includes the pivotal vectorization process within the Vector Space Model (VSM). This vectorization process contributes to creating a meaningful and organized representation of the data by representing keywords from posts or tweets.

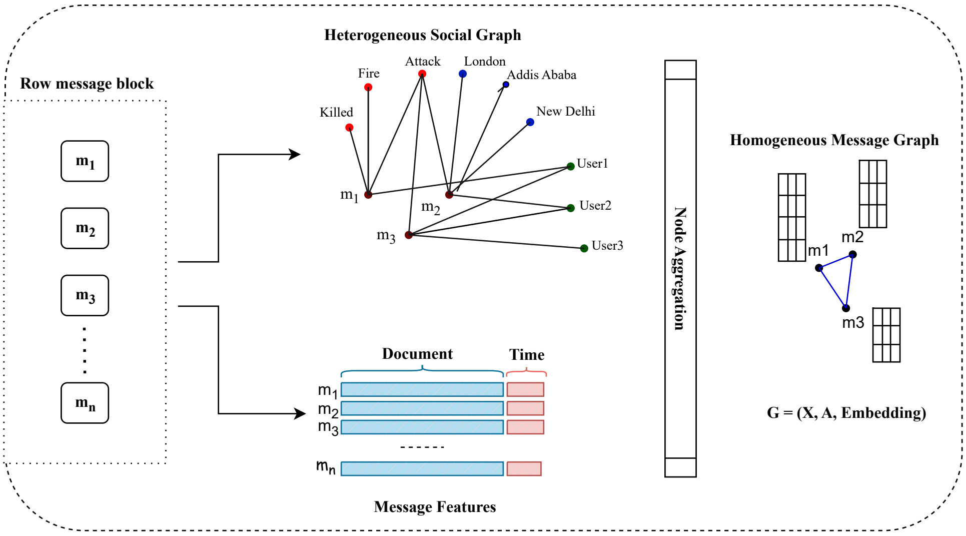


Fig. 1: Data pre-processing and graph generation

4.2 Heterogeneous Social Message Modeling

In the pre-processing stage, our primary objectives revolve around optimizing data utilization by extracting salient and diverse segments from the messages. Simultaneously, we aim to organize these extracted components within a coherent framework, laying the foundation for subsequent processing stages. To achieve these aims, we leverage the power of Heterogeneous Information Networks (HINs) (Peng et al., 2019).

Figure 1 visually portrays a heterogeneous social message graph interacting with different nodes and edges. Our methodology aggregates identified entities and words from relevant documents to analyze a specific message, denoted as m_i. Subsequently, the collected information and users associated with m_i and m_j are incorporated into the HIN as distinct nodes. To maintain clarity and distinction between the messages m_i and their constituent elements, well-defined boundaries are established to distinguish the individual parts of messages (m_i) and their related elements, such as keywords, users, and entities within our approach. For instance, consider Figure 1, where the tweet node m_1 is extracted, accompanied by word nodes like 'killed' and 'attack' (note that the illustration highlights only two words; however, several more words require retrieval). Moreover, other nodes such as 'user1' and 'user2' originating from m_1 contribute to the network.

The methodology includes eliminating redundant nodes, establishing connections between m_1 and other nodes sharing information with m1, and replicating this process for all messages. As an illustration, we establish an edge between two messages if they have a common keyword. The outcome is an evolving social network enriched with information, encompassing every communication fragment and its integral constituents.

4.3 Knowledge Preserving Incremental Message Embedding

Definition 4 presents the incremental operation of our model, depicted in Figure 2 and Algorithm 1. The model follows a life cycle comprising pre-training, detection, and maintenance phases. In the pre-training phase, an initial message graph is constructed and trained. Subsequently, the graph is updated with input message blocks in the detection phase, facilitating event detection. The model continuously processes message blocks until transitioning to the maintenance stage. In this stage, the model is retrained using data from the preceding time frame. This iterative process ensures the model discards irrelevant information and remains up-to-date with the latest data. Consequently, the enhanced model becomes adept at identifying events in the subsequent time frame. In this manner, the model dynamically adjusts to incoming input, detects new events, and takes the learned knowledge as current for the next frame.

Heterogeneous GNNs have primarily concentrated on integrating different node types to capture node representations, as evident from prior studies (Z. Hu et al., 2020; Wang et al., 2019; C. Yang et al., 2020; Yun et al., 2019; C. Zhang et al., 2019). These models encompass nodes with diverse characteristics and features. However, our document pivot model adopts a different approach. We focus on uncovering connections between messages rather than emphasizing the variety of nodes. To achieve this, we utilize a unique architecture that transforms the diverse social graph into a uniform messaging graph. The resulting graph consists of nodes connected by edges linking messages with similar attributes. This transformation ensures that the message correlations initially present in the mixed graph are retained in the new homogeneous structure.

$$A_{i,j} = min\{[\sum_k W_{mk}.(W_{mk})^T]_{i,j}, 1\}, k \in \{o, e, u\} \quad (1)$$

The adjacency matrix A represents a graph where each node is a message, and N is the total number of messages. The node type is denoted by k (words, named entities, or users) and $[.]^T$ denotes the matrix transpose. The function min$\{,\}$ chooses the minimum value. W_{mk} refers to a sub-part of the matrix within the adjacency matrix of the heterogeneous social graph. When m_i and m_j are connected to the same type of k nodes, the value of $[W_{mk}.\ W_{mk}^T]_{i,j}$ will be at least one, and this sets the value of $A_{i,j}$ to one.

The process of generating message feature vectors utilizes the semantic and temporal information of the data. To calculate document features, we take the average of all words in pre-trained word embeddings (Mikolov, Sutskever, et al., 2013). We create Temporal characteristics by encoding the timestamps and transforming them into OLE dates, from which we create 2-dimensional vectors by combining the fractional and integral components (Cao et al., 2021). These vectors are then concatenated for each message in the data. The homogeneous graph, denoted as $\mathcal{G} = (X, A)$, is constructed using initial feature vectors of message nodes, represented by $X = x_{m_i} \in \mathbb{R}^d | 1 \le i \le N$. The initial feature vectors have a dimension of d and are linked to message nodes in the graph.

As the graph $\mathcal{G}$ constantly changes, whenever a new message block is received for detection, we modify $\mathcal{G}$ by incorporating new nodes, forming connections with pre-established nodes, and generating internal connections. Similarly, we regularly resemble the graph's new nodes and corresponding edges.

After constructing the message graph, a GNN encoder is applied to learn the complete representations. This encoder learns node representations by repeated feature aggregation extracted from their immediate neighbors to include the rich semantics and structure properly. The term $h^l_{m_l}$ denotes representation of m_i at l^{th} layer, and its updated representation in $(l+1)^{th}$ layer is as follows:

$$h^{l+1}_{m_i} \leftarrow \overset{head}{||} \Big(h^l_{m_i} \oplus \underset{\forall m_j \epsilon \mathcal{N}(m_i)}{Aggregator} \Big(Extractor \big(h^l_{m_j} \big) \Big) \Big) \quad (2)$$

Where $\mathcal{N}(m_i)$ denotes the immediate neighbor of m_i, $\oplus$ denotes aggregation, and $\overset{head}{||}$ denotes the head-wise concatenation. $Extractor(.)$ function captures relevant features from representations of neighboring messages of current layer(m_j), whereas the $Aggregator(.)$ function averages embedding node information within the neighborhood.

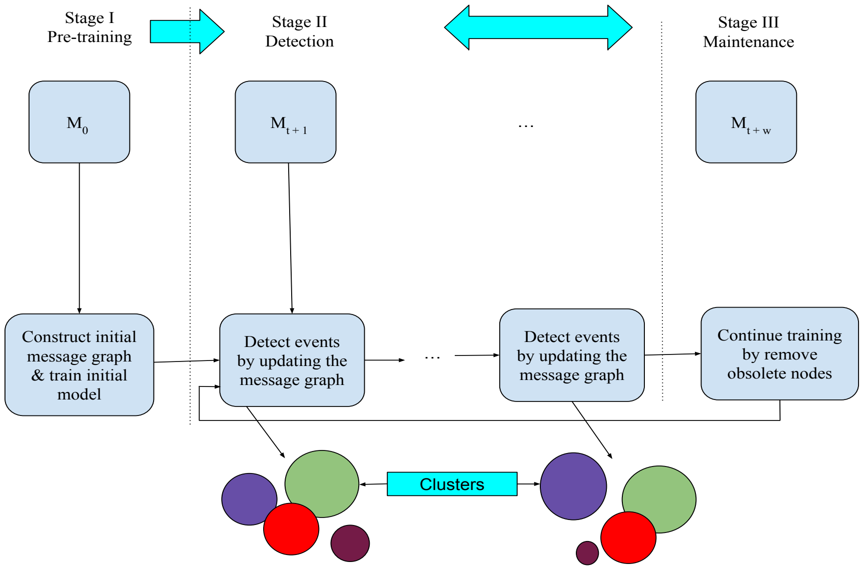


Fig. 2: Life cycle of incremental event detection. **Stage I**, this stage utilizes the pre-trained model to identify the social events in texts that have not been encountered before. **Stage III**, the model is kept up-to-date training continuous through using newly arrived messages provided during **Stage II**.

4.4 Node Filtering

Social media networks have complex structures with many irrelevant nodes for detecting events. Node filtering improves event detection accuracy by discarding irrelevant messages and reducing computation time. However, the additional step of node filtering can also increase complexity and resource demands. Choosing filtering criteria is challenging and requires expertise or experimentation. Balancing these trade-offs and finding the optimal configuration is crucial. We propose two techniques for node filtering, which are discussed below.

4.4.1 DistilBERT-GNN Model for Node Filtering Based on Sentiment Analysis

Node filtering is a widely used graph analysis technique that identifies and focuses on specific node groups of interest (Zola et al., 2022). One approach uses sentiment analysis to determine the emotional tone of associated messages or comments before constructing the graph. A natural language processing model called DistilBERT is employed to predict message emotions. As illustrated in Figure 3b, DistilBERT is initially trained on a dataset of text messages, enabling sentiment prediction of communications connected to nodes (Kusal et al., 2022). The model assigns a confidence score

Algorithm 1 Algorithm of DistilBERT GNN model

Input: A Social message: $S = M_0, M_1, ..., M_n$
Output: Set of Events: $E_0, E_1, ..., E_n$
while $t = 0, 1, 2, ..., n$ **do**
  if $t == 0$ **then**
    $G \leftarrow$ Construct initial message graph (Section 4.2)
    Apply DistilBERT
  else
    $G \leftarrow$ update M_t into message graph (Section 4.2)
  end if
  for $l = 1, 2, 3, ..., L$ **do**
    $h_t^{(l)} \leftarrow$ Eqn 2, $\forall m_i \in M_t$
  end for
  $E_t \leftarrow$ Message Clustering (Section 4.5)
  if $(t\%w = 0)$ **then**
    if $t \neq 0$ **then**
      $G \leftarrow$ extract similar messages
    end if
    for $b = 1, ..., B$ **do**
      $\{m_b\} \leftarrow$ sample mini-batch of G
      for $l = 1, ..., L$ **do**
        $h_{m_b}^{(l)} \leftarrow$ Eqn 2, $\forall m_i \in \{m_b\}$
      end for
      $h_{m_i} \leftarrow h_{m_i}^{(L)}, \forall m_i \in \{m_b\}$
      $T \leftarrow$ triplet sampling, $\forall m_i \in \{m_b\}$ (Section 4.6)
      $L_t \leftarrow$ Eqn 3, $\forall m_i \in \{m_b\}$
      $L_p \leftarrow$ Eqn 4, $\forall m_i \in \{m_b\}$
      Calculate the summary and corrupted representation of $m_i, \forall m_i \in \{m_b\}$ (Section 4.6)
      Back-propagation to update the parameters;
    end for
  end if
end while

indicating its prediction certainty. If the score exceeds 50%, the message has reliable sentiment prediction and is included as a node. Otherwise, the message is excluded. Node filtering using sentiment analysis can uncover communities, influencers, or topics of interest in social network analysis by understanding message emotional tones. Incorporating sentiment filtering provides a more detailed and informative graph representation based on emotional content.

4.4.2 Centrality Measure Based Model for Node Filtering Technique

The literature has covered centrality, which rates node importance in a network. By considering node density, we propose another node filtering method using centrality measurements to separate significant topics from random messages. Three centrality measurements are used: degree, proximity, and betweenness. The proper density factor is determined using these. We calculate the centrality measures of all nodes and rank them by their scores. Nodes below a threshold are removed. Identifying the most critical nodes is essential for understanding network dynamics and structure with applications in social network research and event detection. Centrality-based node filtering produces a sharper network representation with insights into key elements. The optimal node filtering technique depends on the study's objectives, graph characteristics, and nature of the analyzed data. Here, the DistilBERT-GNN model is chosen over centrality-based filtering.

However, centrality-based filtering may miss event-relevant nodes with low centrality scores. While useful for understanding overall network structure by focusing on key nodes, centrality methods can overlook nodes important for analyzing specific events. The choice between DistilBERT-GNN and centrality-based filtering depends on the study's goals. Centrality-based filtering is more suitable when identifying key network nodes to understand overall dynamics and structure.

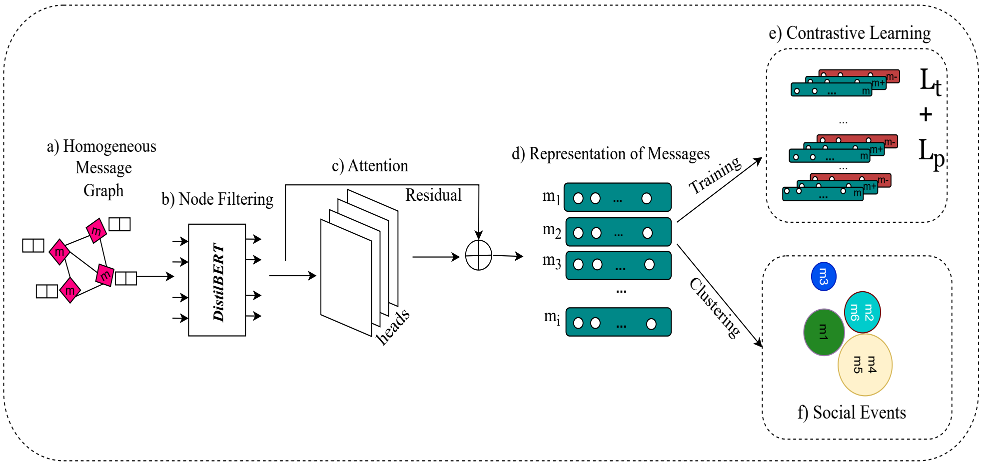


Fig. 3: The architectural design of DistilBERT-GNN

4.5 Message Clustering

In this phase, the message is clustered into several events. The model takes one message block at a time, updates the message graph, and detects events. As illustrated in Figure 3f, the colorful bubbles' indicate how the model has categorized messages into clusters, each corresponding to a unique event. We group messages based on our learned message representations at the detection stage. Distance-based clustering algorithms like K-Means and density-based ones like DBSCAN (Bravo-Marquez et al., 2014) can quickly cluster the representations. Among these, DBSCAN suits incremental event detection since it does not require the total number of classes. Based on Definition 2, the message clusters produced by the model are regarded as social events.

4.6 Scalability Through Contrastive Learning

Event detection in social media occurs in a dynamic data flow where new unseen events arise, making cross-entropy loss inapplicable. Triplet loss is used instead, requiring triplets T of an anchor, positive, and negative sample. An anchor message is an instance from which the learning process starts. It serves as a reference point around which other instances are compared. If the anchor message is m_i, positive and negative samples are m_{i+} and m_{i-} respectively. The triplet loss is:

$$\mathcal{L}_t = \sum_{(m_i, m_{i+}, m_{i-}) \in T} \max\{D(h_{m_i}, h_{m_{i+}}) - D(h_{m_i}, h_{m_{i-}}) + a, 0\} \tag{3}$$

Where $D(,)$ is the Euclidean distance and $a \in \mathbb{R}$ is the hyper-parameter that governs the minimum distance between negative and positive messages. Additionally, the max operation selects the larger value from two operands, and T represents a sequence of triplets chosen in real time. Our method employs the hard triplet, focusing on the distance between the anchor message (h_{m_i}) and the negative message ($h_{m_{i-}}$), which must be less than the distance between the anchor message (h_{m_i}) and the positive message ($h_{m_{i+}}$). In other words, minimizing the intra-distance of the dataset rather than the inter-distance of the datasets is crucial, as using hard triplets results in quicker convergence, and the training set is semi-supervised during training. The triplet loss function is instrumental in addressing the challenge of discovering new event types in dynamic message streams where the total number of social event categories is unknown.

To consider the features of the graph, we use global-local pair loss, which allows the model to recognize and maintain the properties of similar local structures. This is illustrated in the accompanying diagram. The global-local pair loss is developed based on noise-constructive principles (Velickovic et al., 2019) and aims to maximize mutual information by reducing the cross-entropy between the local representations and the aggregation of the entire graph.

$$\mathcal{L}_p = \frac{1}{N} \sum_{i=1}^{N} \Big(\log S(h_{m_i}, s) + \log((1 - s)(\tilde{h}_{m_i}, s)) \Big) \tag{4}$$

Where $s \in \mathbb{R}^d$ represents the message graph summary, which is obtained by averaging all message representations, $\tilde{h}_{m_i}$ denotes a corrupted version of m_i obtained through learning by $\varepsilon(\tilde{X}, A)$, where $\tilde{X}$ is constructed by shuffling the rows of X. A bilinear scoring function, $S(,)$, is utilized to determine the probability that two operands are drawn from a joint distribution, indicating that they were learned from the same graph. The concept of cross entropy applies to dynamic message graphs. To compare the similarity between two probability distributions, one representing the expected distribution and the other representing the actual distribution. Cross-entropy is a notion that may be used in dynamic graph communications. It can be used to quantify the discrepancy between the predicted and actual message representations in the context of dynamic graph messaging. Gradient-based optimization approaches can be used to reduce it while learning. In particular, the cross-entropy loss is frequently employed as the objective function in Message Passing Neural Networks (MPNNs) to train the model to predict a graph's target label or feature reliably. We demonstrate with experiments how $\mathcal{L}_p$ improves the performance.

During training, we used mini-batch sub-graph sampling (Mikolov, Chen, et al., 2013) to make the model scalable to large message graphs. Each mini-batch creates the triplets used in $\mathcal{L}_t$. Each sub-graph's h_{m_i}, $\tilde{h}_{m_i}$, and s in $\mathcal{L}_p$ are likewise computed. Since the model is incremental, it is not trained only once but continuously. Instead, as illustrated in Figure 2 above, we restart the training regularly to maintain the model's information up to the current. In the maintenance stage, the training is continued utilizing the updated data received during the most recent time frame rather than starting over from scratch based on prior knowledge (i.e., the current model parameters). The model needs labels to calculate $\mathcal{L}_p$, but it doesn't require full labeling because it can sample T from labeled messages. Even unlabeled messages can provide helpful information because their features and structural details can be incorporated into labeled statements through propagation. On the other hand, no labels are needed to compute $\mathcal{L}_p$. This idea is helpful for situations where hashtags are used as labels and social media feeds are only partially labeled in real-world scenarios. The overall loss is the sum of triplet and global-local pairwise-wise loss.

$$\mathcal{L} = \mathcal{L}_p + \mathcal{L}_t \tag{5}$$

Addressing the role of hard negative samples is an important component of fine-tuning our event detection algorithm in social media streams. Specifically, we look at the effect of eliminating hard negative data, which are cases in which the model incorrectly labels as events despite being irrelevant or noisy. By eliminating these hard negatives from our training approach, we want to direct the model's attention to more clear-cut and unequivocal instances of true occurrences. This strategy is based on the premise that hard negatives, because of their difficulty, may contribute noise and lead to over-fitting, in which the model becomes too customized to the characteristics of the training data, hurting its capacity to generalize to unknown data. By removing such samples, we improve the model's capacity to differentiate and learn the underlying patterns and signals indicative of major events, increasing its accuracy and dependability in real-world situations where early event detection precision is critical. This stage in our technique is intended to balance the model's sensitivity to actual events against its risk of being affected by erroneous information, improving its performance in detecting events actively discussed in social media streams.

5 Experiments

We tested the performance and efficiency of our proposed methods and streaming system against state-of-the-art models on real-world event detection and evolution tasks. We also examined the effectiveness of the aggregation technique.

5.1 Datasets

To evaluate DistilBERT-GNN, we used Twitter datasets collected over 28 days from October 10 to November 7, 2022, covering events like earthquakes and storms. The dataset has 68,841 messages, 16,358,812 edges, and 503 event categories (Ren et al., 2022). We divided the samples into 70% training, 20% test, and 10% validation sets in a random state.

5.2 Experimental Setting

The authors in (Mathioudakis & Koudas, 2010) set the total number of topics for LDA (Blei et al., 2003) as 10, and we assume 50 to capture several classes and hyper-parameter adoption for EventX (Liu et al., 2020). For graph-based event-related methods like KPGNN (Cao et al., 2021) as well as QSGNN (Ren et al., 2022), (Peng et al., 2019) and BiLSTM (Graves & Schmidhuber, 2005), we set the number of heads to 4, dimension of word embedding to 64, the number of layers to 2, learning rate to 0.001, use Adam optimizer, and apply 200 training epochs with patience of 5 to stop early to prevent over-fitting. Table 3 summarizes all the parameters used in our implementation.

Table 2: The statistics of the social stream.

Blocks	# of messages	Blocks	# of messages
M_0	20,254	M_{11}	1,232
M_1	8,722	M_{12}	3,237
M_2	1,491	M_{13}	1,972
M_3	1,835	M_{14}	2,956
M_4	2,010	M_{15}	2,549
M_5	1,834	M_{16}	910
M_6	1,276	M_{17}	2,676
M_7	5,278	M_{18}	1,887
M_8	1,560	M_{19}	1,399
M_9	1,363	M_{20}	893
M_{10}	1,096	M_{21}	2,410

We performed all the experiments five times and reported mean and standard variance in the result. DistilBERT-GNN does not require the entire set of event types to be predefined classes, unlike some baselines such as WMD(Kusner et al., 2015), LDA(Blei et al., 2003), Word2vec(Mikolov, Chen, et al., 2013), BiLSTM(Graves & Schmidhuber, 2005), BERT(Devlin et al., 2018) and KPGNN(Cao et al., 2021). To

Table 3: Model parameter setting

Parameters	Value
Number of heads	4
Embedding dimension	64
Number of layers	2
Learning rate	0.001
Optimizer	Adam
epoch	200
early stopping	5
Window size	3
Size of mini-batch	200
Margin for triplet	3
number of neighbors sampled	800

ensure a fair comparison, DistilBERT-GNN does not predetermine the number of detected classes after calculating the message dissimilarity matrix from WMD(Kusner et al., 2015) and the representations from all models except for EventX(Liu et al., 2020). Spectral and K-Means clustering methods are utilized to cluster the data, and the total number of classes is set to match the number of known classes. However, when the total number of classes is uncertain, such as in cases of incremental detection, the DBSCAN algorithm can be used instead. We implemented all the experiments on 48 core processor AuthenticAMD with 3.1GHz, 256GB RAM, cuda version of 11.2, and AMD EPYC 7552 model.

5.3 Baselines

We evaluate our model against various state-of-the-art ED methods, including those that learn generic message representations and measure similarity, methods for detecting social events offline, and methods for incremental detection. The baselines include:

- **Word2vec** (Mikolov, Chen, et al., 2013) applies a technique to generate word vector representations from a substantial dataset. The model showcases that these vectors perform well in evaluating both words' syntactic and semantic similarity.
- **LDA** (Blei et al., 2003) presents a probabilistic model designed for discrete data collections, such as text corpora, over an underlying hierarchy of topics.
- **WMD** (Kusner et al., 2015) proposes a distance metric for measuring the dissimilarity between two text documents.
- **BERT** (Devlin et al., 2018) pertains profound bidirectional encoding from unlabelled text, simultaneously considering both preceding and succeeding context across all layers.
- **BiLSTM** (Graves & Schmidhuber, 2005) capturing long-term relationships among words within a message bidirectionally.
- **PPGCN** (Peng et al., 2019) utilizes a heterogeneous information network and a weighted adjacent matrix to capture the semantic relatedness of social events, integrating information from external knowledge bases.
- **EventX** (Liu et al., 2020) uses an online scheme to extract events from vast news and organize them logically.

- **KPGNN** (Cao et al., 2021) adapts complex and varied social messages into an incrementally formed unified message graph, fostering effective data utilization and probing capabilities of GNNs in extracting knowledge.
- **QSGNN** (Ren et al., 2022) is a Quality-aware Self-improving Graph Neural Network (QSGNN) that addresses the constraints of previous GNN-based approaches in open set social event detection by extending information from known events to unknown ones via trustworthy knowledge transfer and a unique supervised paired loss. The method uses existing occurrences as reference grounds for unknown events, improving knowledge acquisition and transmission.

5.4 Metrics

We measure the similarities between the detected message groups and the recognized clusters to determine the systems' effectiveness. We make use of normalized mutual information (NMI), adjusted mutual information (AMI), and the adjusted rand index (ARI).

- **NMI**: It measures the extent of data mutually exchanged between ground truth events related to a particular collection of tweets and how clusters are assigned. We calculate the Mutual Information (MI) between the two data sets, typically between the "ground truth labels" and the "predicted labels" obtained from a clustering algorithm. Then, normalize the mutual information using entropy measures to obtain the NMI score. It is useful because it quantifies the degree of agreement or similarity between two sets of labels while considering the chance agreement. It ranges from 0 to 1, with higher values indicating better agreement or clustering quality. A score of 1 indicates perfect agreement, while 0 indicates no agreement beyond what would be expected by chance.
- **AMI**: An evaluation metric commonly used in clustering analysis and community detection tasks. AMI is a variation of NMI that incorporates an adjustment to account for chance agreement more effectively. It measures the agreement between two sets of labels while considering their mutual information and expected mutual information due to chance.
  We first calculate the MI between the two sets of labels, typically the ground truth labels and the predicted labels obtained from a clustering algorithm, and calculate the expected mutual information (expected MI) by considering the labels as random variables and calculating what the mutual information would be if they were randomly assigned. Subtract the expected MI from the actual MI and normalize the result to obtain the AMI score.
- **ARI**: It measures how similar two data clusters are and is used extensively in statistics and machine learning. Unlike NMI and AMI, it is based on the Rand Index (RI), a measure of the similarity between two data clusterings, but with an adjustment for chance. The ARI accounts for the fact that random clustering decisions can lead to a non-zero RI score. The ARI normalizes the RI such that random clustering has an ARI close to 0, while an ARI of 1 indicates perfect agreement between two clusterings. The ARI is particularly useful for comparing the performance of

different clustering algorithms or validating the consistency of a single algorithm, as it provides a more robust and chance-corrected measure of clustering similarity.

To clarify, we define a group of clusters as $C = (C_1, C_2, ..., C_j)$, where each C_i represents an individual cluster. Similarly, we denote a group of events as $E = (E_1, E_2, ..., E_k)$, with each E_i signifying an individual event. Both clusters and events consist of tweets. We use n to represent the total tweets across all clusters and events. The metrics used in our analysis are formulated as follows.

$$NMI(C, E) = \frac{MI(C, E}{H(C) + H(E)/2} \tag{6}$$

Where,

$$MI(C, E) = \sum_j \sum_k \frac{|e_k \cap c_j|}{n} \log \frac{n * |e_k \cap c_j|}{|e_k| * |c_j|} \tag{7}$$

$$H(C) = -\sum_j \frac{|c_j|}{n} \log \frac{|c_j|}{n} \tag{8}$$

$$H(E) = -\sum_k \frac{|e_k|}{n} \log \frac{|e_k|}{n} \tag{9}$$

$$AMI(C, E) = \frac{MI(C, E) - E\{MI(C, E)\}}{\{(H(C) + H(E))/2\} - E\{MI(C, E)\}} \tag{10}$$

From the formulas above, C denotes the vector of the ground-truth label, and E denotes the vector of the predicted label. H denotes entropy, $MI(,)$ denotes mutual information, and $E\{.\}$ denotes expected value.

6 Results

This section provides an in-depth examination of these results, including a comprehensive evaluation of the model's performance, relative effectiveness, and resilience in various circumstances. By combining quantitative evaluations and qualitative insights, we aim to provide a precise and all-encompassing perspective on how our method contributes to the area and paves the way for future advancements.
We assess for both offline and online evaluation, including baselines, and update the graph accordingly. In addition, an ablation study and hyper-parameter sensitivity were performed to investigate the effect and effectiveness of the parameters. Finally, we discussed the model in many aspects, such as execution time and convergence.

6.1 Offline Evaluation

Offline evaluation is conducted using historical or pre-collected data without the need for real-time user interactions. So, we take the whole dataset together rather than dividing it into message blocks. Table 4 compares offline evaluation results for several models on the Twitter dataset using three separate metrics: NMI, AMI, and ARI. The table compares the proposed DistilBERT-GNN model and many well-known techniques as a baseline. The DistilBERT-GNN model demonstrates superior performance compared to other models in all measures, with NMI scores of 0.72±01, AMI scores of 0.53±02, and ARI scores of 0.24±02.

To compare the performance of DistilBERT-GNN against the baseline models in terms of NMI, AMI, and ARI, we calculate the percentage improvement of DistilBERT-GNN over each baseline. Specifically, the DistilBERT-GNN model has demonstrated superior results across various metrics compared to other models. It outperforms Word2Vec with a 28% higher NMI, a 40% higher AMI, and a 22% higher ARI. Against LDA, the improvements are even more pronounced with a 43% increase in NMI, a 49% increase in AMI, and a 23% increase in ARI. When measured against WMD, the enhancements are more modest but still notable at 7% for NMI, 3% for AMI, and 16% for ARI. BERT and BiLSTM models also fall behind, with DistilBERT-GNN showing an 8% and 9% improvement in NMI, respectively, and even greater leads in AMI and ARI. PP-GCN and EventX show narrower outperform margins, but DistilBERT-GNN still surpasses them in all evaluated metrics. Even the closely related KPGNN and QSGNN models are outperformed by margins up to 4% across the board, confirming DistilBERT-GNN's effectiveness.

Our model outperforms traditional methods hugely compared to graph-based methods. This indicates that GNN has a precise determination of semantic relationships within the data. It is worth mentioning that traditional methods such as Word2vec and LDA provide much worse results, highlighting the efficacy of combining GNNs with transformer-based models to extract meaningful patterns from social media data. The standard deviation of each score reflects the level of consistency in the model's performance over several experimental runs or subsets of the dataset.

Table 4: Offline evaluation results on the Twitter dataset

Metrics	NMI	AMI	ARI
Word2vec (Mikolov, Chen, et al., 2013)	.44±.00	.13±.00	.02±.00
LDA (Blei et al., 2003)	.29±.00	.04±.00	.01±.00
WMD (Kusner et al., 2015)	.65±.00	.50±.00	.06±.00
BERT (Devlin et al., 2018)	.64±.00	.44±.00	.07±.00
BiLSTM (Graves & Schmidhuber, 2005)	.63±.00	.41±.00	.17±.00
PP-GCN (Peng et al., 2019)	.68±.02	.50±.02	.20±.01
EventX (Liu et al., 2020)	.72±.00	.19±.00	.05±.00
KPGNN (Cao et al., 2021)	*.70±.01*	*.52±.01*	*.22±.01*
QSGNN (Ren et al., 2022)	.70±.01	.51±.02	.20±.02
DistilBERT-GNN	**.72±01**	**.53±02**	**.24±02**

6.2 Online Evaluation

Online evaluation refers to evaluating a system or model in real-time user interactions as it is being used in a live environment. In particular, the messages from the initial week are used to create the first message block, M0, while the messages from subsequent weeks are organized into the subsequent message blocks, M1, M2, ..., and M21. Table 2 outlines the statistics of the generated social stream. Table 5 presents a comparative analysis of NMI across multiple blocks for different baselines. The highest NMI score in each block is bolded, drawing attention to the models with the best performance in capturing the mutual information, the predicted labels, and the true labels. The scores vary across the blocks, indicating that the effectiveness of each block may depend on the specific characteristics of data in each block. DistilBERT, in each message block, outperforms other baselines, showing higher NMI values in most blocks.

The AMI scores for a range of blocks (M1 to M21) across different baselines are shown in Table 6. Standard deviations accompany the AMI values; the highest scores indicate the top-performing models in each block. This table allows for assessing each baseline's performance in preserving the information shared between the clusters assigned and the true classes, adjusted for chance. DistilBERT-GNN shows notable performance, often achieving the highest scores, suggesting its robustness in clustering or similar tasks where mutual information is a critical performance measure. The consistency of the scores within and across models provides insights into their stability and reliability in different data segments or experimental conditions.

Online evaluation of ARIs for each block, as shown in Table 7, presents the ARI scores for several baselines across multiple message blocks (M1 to M21). The ARI measures the similarity between two data clusters, adjusted for chance, with higher scores indicating a greater similarity to the true cluster assignments. The highest ARI scores in each block are particularly bolded, indicating the most effective models for the given block. The variation in scores across different blocks and models can offer insights into these models' comparative robustness and effectiveness in clustering tasks, with our model often emerging as a strong performer. This tabular assessment is crucial for understanding how well each DistilBERT can discover and align with the inherent clustering in the data.

6.3 Message updating strategies

In the detection phase, as illustrated in Stage II of Figure 2, DistilBERT-GNN integrates new messages into the graph G. Furthermore, in the maintenance phase (Stage III of Figure 2), it regularly purges old messages from G while continuing its training to accommodate new messages.

The method employed for updating and maintaining DistilBERT-GNN is crucial in determining its computational efficiency, the extent of knowledge it conserves, and overall performance. This subsection is dedicated to analyzing and contrasting three

Table 5: Online evaluation of NMIs for each block.

Blocks	M1	M2	M3	M4	M5	M6	M7
Word2vec (Mikolov, Chen, et al., 2013)	.19±.00	.50±.00	.39±.00	.34±.00	.41±.00	.53±.00	.25±.00
LDA (Blei et al., 2003)	.11±.00	.27±.01	.28±.00	.25±.00	.26±.00	.32±.00	.18±.01
WMD (Kusner et al., 2015)	.32±.00	.71±.00	.67±.00	.50±.00	.61±.00	.61±.00	.46±.00
BERT (Devlin et al., 2018)	.36±.00	.78±.00	.75±.00	.60±.00	.72±.00	.78±.00	.54±.00
BiLSTM (Graves & Schmidhuber, 2005)	.24±.00	.50±.00	.39±.00	.40±.00	.41±.00	.50±.00	.33±.00
PP-GCN (Peng et al., 2019)	.23±.00	.57±.02	.55±.01	.46±.01	.48±.01	.57±.01	.37±.00
Eventx (Liu et al., 2020)	.36±.00	.68±.00	.63±.00	.63±.00	.59±.00	.70±.00	.51±.00
KPGNN (Cao et al., 2021)	.39±.00	.79±.01	.76±.00	.67±.00	.73±.01	.82±.01	.55±.01
QSGNN (Ren et al., 2022)	.43±.01	.81±.02	.78±.01	.71±.02	.75±.00	.83±.01	**.57±.01**
DistilBERT-GNN	**.62±.01**	**.86±.00**	**.79±.01**	**.84±.02**	**.84±.01**	**.84±.02**	.47±.01

Blocks	M8	M9	M10	M11	M12	M13	M14
Word2vec (Mikolov, Chen, et al., 2013)	.46±.00	.35±.00	.51±.00	.37±.00	.30±.00	.37±.00	.36±.00
LDA (Blei et al., 2003)	.37±.01	.34±.00	.44±.01	.33±.01	.22±.01	.27±.00	.21±.00
WMD (Kusner et al., 2015)	.67±.00	.55±.00	.61±.00	.50±.00	.60±.00	.54±.00	.66±.00
BERT (Devlin et al., 2018)	.79±.00	.70±.00	.74±.00	.68±.00	.59±.00	.63±.00	.64±.00
BiLSTM (Graves & Schmidhuber, 2005)	.49±.00	.43±.00	.50±.00	.49±.00	.39±.00	.46±.00	.44±.00
PP-GCN (Peng et al., 2019)	.55±.02	.51±.02	.55±.02	.50±.01	.45±.01	.47±.01	.44±.01
EventX (Liu et al., 2020)	.71±.00	.67±.00	.68±.00	.65±.00	.61±.00	.58±.00	.57±.00
KPGNN (Cao et al., 2021)	.80±.00	.74±.02	.80±.01	.74±.01	.68±.01	.69±.01	.69±.00
QSGNN(Ren et al., 2022)	.79±.01	**.77±.02**	**.82±.02**	.75±.01	.70±.00	.68±.02	.68±.01
DistilBERT-GNN	**.87±.00**	.76±.02	**.82±.00**	**.80±.01**	**.75±.01**	**.83±.01**	**.77±.00**

Blocks	M15	M16	M17	M18	M19	M20	M21
Word2vec (Mikolov, Chen, et al., 2013)	.27±.00	.49±.00	.33±.00	.29±.00	.37±.00	.38±.00	.31±.00
LDA (Blei et al., 2003)	.21±.00	.35±.01	.19±.00	.18±.00	.29±.01	.35±.00	.19±.00
WMD (Kusner et al., 2015)	.51±.00	.60±.00	.55±.00	.63±.00	.54±.00	.58±.00	.58±.00
BERT (Devlin et al., 2018)	.54±.00	.75±.00	.63±.00	.57±.00	.66±.00	.68±.00	.59±.00
BiLSTM (Graves & Schmidhuber, 2005)	.40±.00	.53±.00	.45±.00	.44±.00	.44±.00	.48±.00	.41±.00
PP-GCN (Peng et al., 2019)	.39±.01	.55±.01	.48±.00	.47±.01	.51±.02	.51±.01	.41±.02
EventX (Liu et al., 2020)	.49±.00	.62±.00	.58±.00	.59±.00	.60±.00	.67±.00	.53±.00
KPGNN (Cao et al., 2021)	.58±.00	.79±.01	.70±.01	.68±.02	.73±.01	.72±.02	.60±.00
QSGNN (Ren et al., 2022)	**.59±.01**	.78±.01	.71±.01	.70±.01	.73±.00	**.73±.02**	**.61±.01**
DistilBERT-GNN	.56±.00	**.85±.01**	**.73±.01**	**.76±.02**	**.74±.02**	.71±.00	**.61±.02**

Table 6: Online evaluation of AMIs for each block.

Blocks	M1	M2	M3	M4	M5	M6	M7
Word2vec (Mikolov, Chen, et al., 2013)	.08±.00	.41±.00	.31±.00	.24±.00	.33±.00	.40±.00	.13±.00
LDA (Blei et al., 2003)	.08±.00	.20±.01	.22±.01	.17±.00	.21±.00	.20±.00	.12±.01
WMD (Kusner et al., 2015)	.30±.00	.69±.00	.63±.00	.45±.00	.57±.00	.57±.00	.46±.00
BERT (Devlin et al., 2018)	.34±.00	.76±.00	.73±.00	.55±.00	.71±.00	.74±.00	.50±.00
BiLSTM (Graves & Schmidhuber, 2005)	.12±.00	.41±.00	.31±.00	.30±.00	.33±.00	.36±.00	.20±.00
PP-GCN (Peng et al., 2019)	.21±.00	.55±.02	.52±.01	.42±.01	.46±.01	.52±.02	.34±.00
EventX (Liu et al., 2020)	.01±.00	.45±.00	.09±.00	.07±.00	.04±.00	.14±.00	.02±.00
KPGNN (Cao et al., 2021)	.37±.00	.78±.01	.74±.00	.64±.01	.71±.01	.79±.01	.51±.01
QSGNN (Ren et al., 2022)	**.41±.02**	.80±.01	.76±.01	.68±.01	.73±.00	**.80±.01**	**.54±.01**
DistilBERT-GNN	.24±.00	**.82±.01**	**.77±.00**	**.66±.01**	**.73±.02**	.77±.01	.50±.03
Blocks	M8	M9	M10	M11	M12	M13	M14
Word2vec (Mikolov, Chen, et al., 2013)	.33±.00	.24±.00	.39±.00	.26±.00	.23±.00	.23±.00	.26±.00
LDA (Blei et al., 2003)	.24±.01	.24±.00	.36±.01	.25±.01	.16±.01	.19±.00	.15±.00
WMD (Kusner et al., 2015)	.63±.00	.46±.00	.57±.00	.42±.00	.58±.00	.50±.00	.64±.00
BERT (Devlin et al., 2018)	.75±.00	.66±.00	.70±.00	.65±.00	.56±.00	.59±.00	.61±.00
BiLSTM (Graves & Schmidhuber, 2005)	.35±.00	.32±.00	.39±.00	.37±.00	.32±.00	.31±.00	.34±.00
PP-GCN (Peng et al., 2019)	.49±.02	.46±.02	.51±.02	.46±.01	.42±.01	.43±.01	.41±.01
EventX (Liu et al., 2020)	.21±.00	.19±.00	.24±.00	.24±.00	.16±.00	.16±.00	.14±.00
KPGNN (Cao et al., 2021)	.76±.01	.71±.02	.78±.01	.71±.01	.66±.01	.67±.01	.65±.00
QSGNN (Ren et al., 2022)	.75±.02	**.75±.03**	**.80±.01**	**.72±.01**	**.68±.00**	.66±.01	.66±.01
DistilBERT-GNN	**.81±.01**	.68±.03	.78±.01	**.72±.00**	.67±.00	**.68±.01**	**.67±.01**
Blocks	M15	M16	M17	M18	M19	M20	M21
Word2vec (Mikolov, Chen, et al., 2013)	.15±.00	.36±.00	.24±.00	.21±.00	.28±.00	.24±.00	.21±.00
LDA (Blei et al., 2003)	.13±.00	.27±.01	.13±.00	.12±.00	.22±.01	.23±.00	.13±.00
WMD (Kusner et al., 2015)	.47±.00	.59±.00	.57±.00	.60±.00	.49±.00	.55±.00	.52±.00
BERT (Devlin et al., 2018)	.50±.00	.72±.00	.60±.00	.53±.00	.63±.00	.62±.00	.57±.00
BiLSTM (Graves & Schmidhuber, 2005)	.26±.00	.41±.00	.35±.00	.35±.00	.35±.00	.34±.00	.31±.00
PP-GCN (Peng et al., 2019)	.35±.01	.52±.01	.45±.00	.45±.01	.48±.02	.45±.02	.38±.02
EventX (Liu et al., 2020)	.07±.00	.19±.00	.18±.00	.16±.00	.16±.00	.18±.00	.10±.00
KPGNN (Cao et al., 2021)	.54±.00	.77±.01	.68±.01	.66±.02	**.71±.01**	.68±.02	.57±.01
QSGNN (Ren et al., 2022)	.55±.01	.76±.02	**.69±.01**	.68±.01	.70±.00	**.69±.02**	**.58±.00**
DistilBERT-GNN	**.56±.02**	**.79±.00**	.66±.02	**.73±.00**	.66±.01	**.69±.01**	.56±.00

Table 7: Online evaluation of ARIs for each block.

Blocks	M1	M2	M3	M4	M5	M6	M7
Word2vec (Mikolov, Chen, et al., 2013)	.01±.00	.49±.00	.16±.00	.07±.00	.17±.00	.25±.00	.02±.00
LDA (Blei et al., 2003)	.00±.00	.08±.00	.02±.01	.07±.00	.06±.00	.07±.01	.00±.00
WMD (Kusner et al., 2015)	.04±.00	.48±.00	.28±.00	.11±.00	.26±.00	.16±.00	.08±.00
BERT (Devlin et al., 2018)	.03±.00	.64±.00	.43±.00	.19±.00	.44±.00	.44±.00	.07±.00
BiLSTM (Graves & Schmidhuber, 2005)	.03±.00	.49±.00	.17±.00	.11±.00	.19±.00	.18±.00	.12±.00
PP-GCN (Peng et al., 2019)	.05±.00	.67±.03	.47±.01	.24±.01	.34±.00	.55±.03	.11±.02
EventX (Liu et al., 2020)	.01±.00	.45±.00	.09±.00	.07±.00	.04±.00	.14±.00	.02±.00
KPGNN (Cao et al., 2021)	.07±.01	.76±.02	.58±.01	.29±.01	.47±.03	.72±.03	**.12±.00**
QSGNN (Ren et al., 2022)	.05±.01	.76±.02	.59±.01	.30±.01	**.49±.00**	.59±.02	.10±.00
DistilBERT-GNN	**.09±.01**	**.79±.02**	**.65±.01**	**.31±.01**	.48±.02	**.64±.01**	**.12±.00**
Blocks	M8	M9	M10	M11	M12	M13	M14
Word2vec (Mikolov, Chen, et al., 2013)	.17±.00	.08±.00	.23±.00	.09±.00	.09±.00	.06±.00	.10±.00
LDA (Blei et al., 2003)	.03±.00	.03±.01	.09±.02	.03±.01	.02±.00	.00±.00	.02±.00
WMD (Kusner et al., 2015)	.22±.00	.12±.00	.20±.00	.12±.00	.27±.00	.13±.00	.33±.00
BERT (Devlin et al., 2018)	.50±.00	.33±.00	.44±.00	.27±.00	.31±.00	.14±.00	.30±.00
BiLSTM (Graves & Schmidhuber, 2005)	.17±.00	.13±.00	.30±.00	.16±.00	.14±.00	.10±.00	.17±.00
PP-GCN (Peng et al., 2019)	.43±.04	.31±.02	.50±.07	.38±.02	.34±.03	.19±.01	.29±.01
EventX (Liu et al., 2020)	.09±.00	.07±.00	.13±.00	.16±.00	.07±.00	.04±.00	.10±.00
KPGNN (Cao et al., 2021)	.60±.01	**.46±.02**	.70±.06	.49±.03	.48±.01	.29±.03	**.42±.02**
QSGNN (Ren et al., 2022)	.65±.01	.44±.02	.59±.02	.48±.00	.49±.00	.29±.01	.31±.02
DistilBERT-GNN	**.67±.01**	.45±.00	**.73±.02**	**.51±.00**	**.51±.01**	**.36±.00**	**.42±.02**
Blocks	M15	M16	M17	M18	M19	M20	M21
Word2vec (Mikolov, Chen, et al., 2013)	.03±.00	.19±.00	.10±.00	.07±.00	.14±.00	.10±.00	.06±.00
LDA (Blei et al., 2003)	.00±.00	.11±.01	.02±.00	.02±.00	.03±.00	.02±.01	.00±.01
WMD (Kusner et al., 2015)	.16±.00	.32±.00	.26±.00	.35±.00	.12±.00	.19±.00	.19±.00
BERT (Devlin et al., 2018)	.10±.00	.41±.00	.24±.00	.24±.00	.32±.00	.33±.00	.18±.00
BiLSTM (Graves & Schmidhuber, 2005)	.08±.00	.27±.00	.22±.00	.19±.00	.16±.00	.20±.00	.16±.00
PP-GCN (Peng et al., 2019)	.15±.00	.51±.03	.35±.03	.39±.03	.41±.02	.41±.01	.20±.03
EventX (Liu et al., 2020)	.01±.00	.08±.00	.12±.00	.08±.00	.07±.00	.11±.00	.01±.00
KPGNN (Cao et al., 2021)	**.17±.00**	.66±.05	.43±.05	.47±.04	.51±.03	.51±.04	.20±.01
QSGNN (Ren et al., 2022)	.15±.02	.64±.00	.39±.01	.49±.00	.50±.00	.49±.02	.25±.00
DistilBERT-GNN	.14±.02	**.67±.02**	**.45±.01**	**.51±.00**	**.54±.01**	**.54±.00**	**.40±.01**

varied methodologies for the update and maintenance process. We illustrate this in Figure 4a, 4b, and 4c for our metrics NMI, ARI and AMI, respectively.

6.3.1 Keep all

The latest message block is directly added to G during detection. In the maintenance phase, training proceeds with all messages in G. Essentially, this allows DistilBERT-GNN to retain all messages it has ever processed. However, this approach is not feasible in practice, as the accumulation of messages in G can gradually decelerate the model and ultimately surpass the embedding space capacity of the message encoder ε. As illustrated in Figure 4, this type of updating strategy degrades the model's performance. We utilize this method solely for the sake of comparison.

6.3.2 Keep relevant

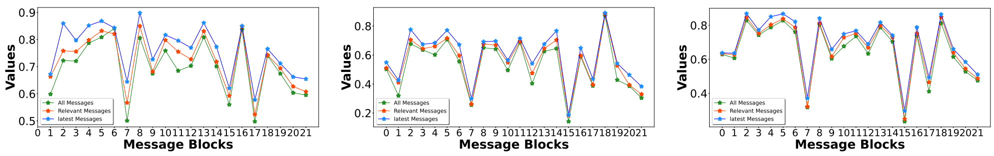
At the detection phase, the newly received message block is added to G. In the maintenance phase, messages with no connections to recent messages (those received within the last time window) are removed first. Then, the training is carried out with all remaining messages in G. This approach allows DistilBERT-GNN to discard outdated messages (arrived outside the time window) and unrelated (to new messages received within the window). While the system prioritizes relevant data, this method ensures that it still benefits from the accumulated knowledge of past messages, although in a transformed and stored format within the model's parameters. Figure 4 shows that this updating strategy is better than the keep-all updating strategy but not better than the keep-latest message updating strategy. It's important to note that the insights gained from these removed messages are retained through the model's parameters.

6.3.3 Keep latest

During the detection phase of this process, our approach focuses exclusively on using the most recent message block to reconfigure G, the graph structure. Moving into the maintenance stage, the training is conducted utilizing all messages currently held within G, which, in this context, consists solely of the latest message block. This means that the DistilBERT-GNN system is configured to retain only the information from the newest message block, effectively disregarding all previous messages. However, it's important to note that the valuable insights and information gleaned from these discarded messages are not lost entirely. Instead, they are encoded and preserved within the model's parameters. As illustrated in Figure 4, this strategy performs better than the other two message-updating strategies. This method ensures that while the system prioritizes current data, it still benefits from the accumulated knowledge of past messages, although in a transformed and stored format within the model's parameters.

6.4 Ablation Study

We rigorously analyze the influence of numerous components of DistilBERT-GNN in our ablation research to understand their individual and combined contributions to its performance. To validate the importance of DGl, we ignored and noticed the difference



(a) NMI (b) ARI (c) AMI

Fig. 4: Metrics comparison with different updating strategies

in triplet loss.
Our ablation study examines the impact of considering vs. removing hard negative samples on the NMI metric across different message block sizes. As illustrated in Figure 5, the bar chart shows two conditions in our model's evaluation: the blue bars indicate the NMI when hard negative samples are removed. At the same time, the red bars with a dotted pattern represent the NMI when hard negative samples are factored into the analysis. Across the spectrum of message block sizes, we observe a discernible variance in NMI scores between the two conditions. Notably, including hard negatives consistently contributes to higher NMI values, underscoring their significance in our model's capability to capture more informative embeddings. This trend is persistent across all message blocks, indicating the robustness of our findings and highlighting the potential benefits of hard negative sampling during the training time.

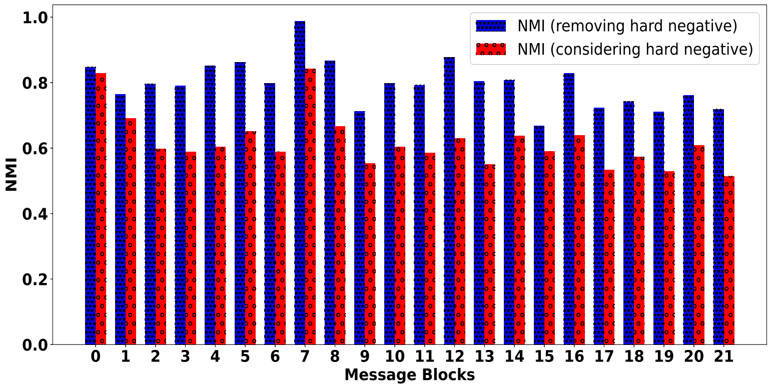


Fig. 5: Result with selection of the data points

6.5 Hyper-parameter Sensitivity

This section examines the impacts of varying two key parameters, namely w, which represents window size used for preserving DistilBERT-GNN, and $|\{mb\}|$, which stands for mini-batch size, within the context of incremental social event detection studies.

6.5.1 Effect of Window Size

The study found that adopting a smaller window size (1 or 3) generally performs slightly better than a larger one. For example, as depicted in Figure 7, the block-wise average NMIs for window sizes 1 and 3 were 0.75, while those for windows 9 and 11 were slightly lower, at 0.74. This suggests that smaller window sizes are slightly more favorable regarding NMI. Generally, smaller window sizes give better performance than larger window sizes. Using smaller window sizes has a lot of advantages over larger window sizes. This happens for various reasons, such as noise reduction, increased sensitivity to local patterns, over-fitting avoidance, and reduced computational efficiency.

6.5.2 Effect of Batch Size

Illustration in Figure 7 reveals that DistilBERT's performance metrics, namely NMI and AMI, are largely stable across different mini-batch sizes, with NMI values consistently around 0.75 and 0.76, suggesting the model's insensitivity to batch size variations. However, slight fluctuations in the ARI are not significantly tied to variations in either mini-batch size or window size, with ARI scores marginally varying from 0.57 to 0.62. This overall stability across different mini-batch and window sizes highlights the robustness of the DistilBERT model in handling varying batch sizes without substantial impact on its clustering performance metrics.

Figure 5 illustrates the impact of varying hyper-parameters on our metrics of DistilBERT that shows the sensitivity of NMI, ARI, and AMI to changes in four key hyper-parameters: 6a the number of neighborhood samples, 6b the point of early stopping during training, 6c the embedding dimension of the input data, and 6d the number of attention heads in a model layer. The metrics are measured across different values for each hyper-parameter.

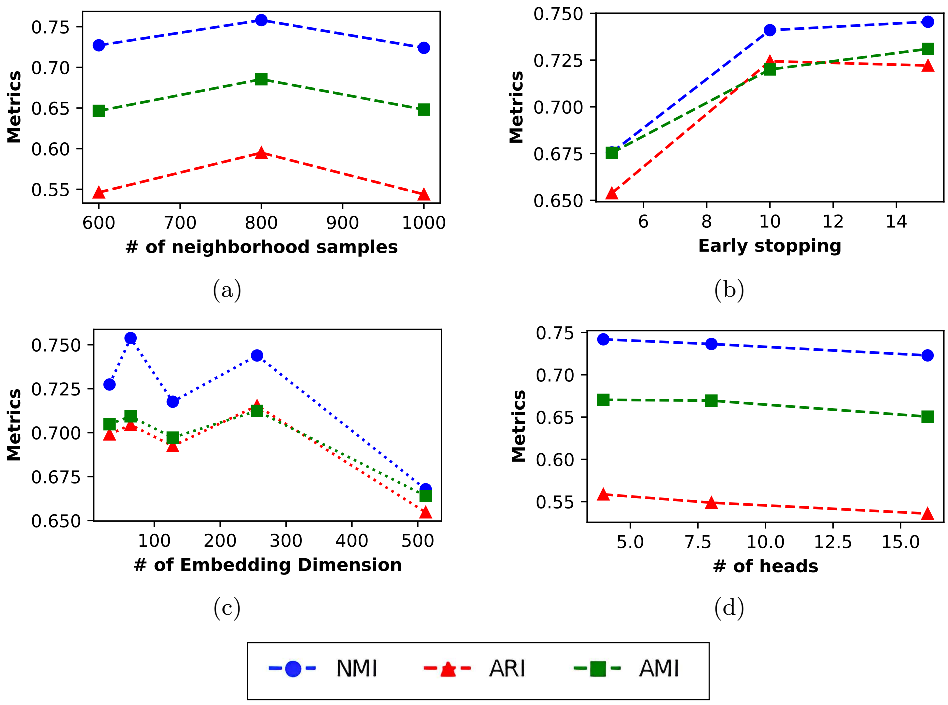


Fig. 6: Parameter Sensitivity Analysis.

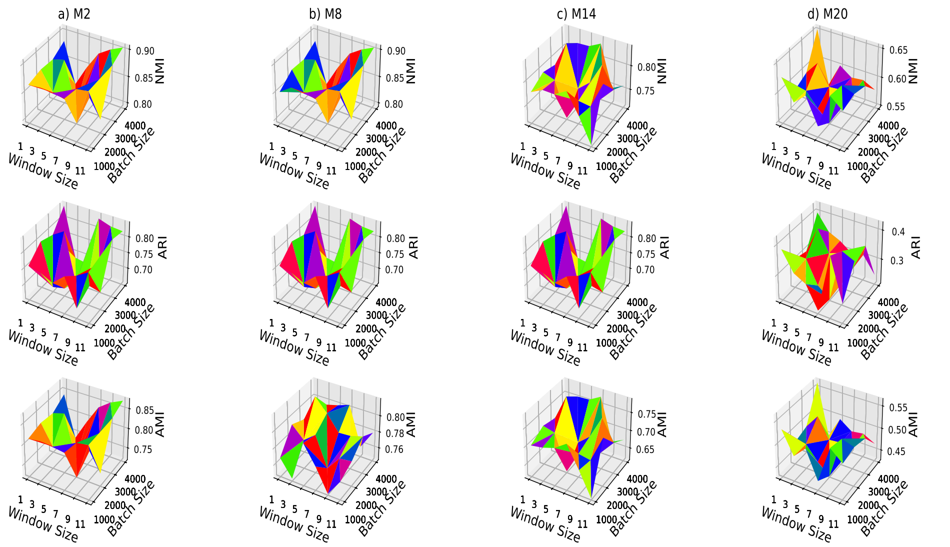


Fig. 7: The model with various parameters: The hues on display depict the rhythmic undulations in values: submerged regions adorned in shades of blue, while elevated contours adorned in lush shades of green.

6.6 Model Discussion

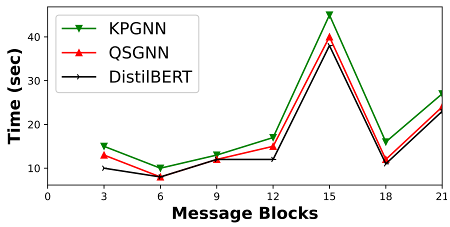
In this section, we discuss the examination of DistilBERT-GNN, specifically focusing on its performance and efficiency in event detection across various message blocks. This discussion is divided into two critical subsections, namely, Execution Time and Convergence, each providing a distinct lens through which the model's efficacy is examined. into two critical subsections, namely, execution time and convergence, each providing a distinct lens through which the model's effectiveness is examined.

6.6.1 Excusion Time

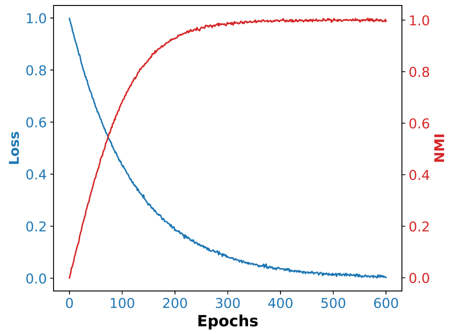
Figure 8a compares execution times for three models, KPGNN, QSGNN, and DistilBERT, across various message block sizes. It is evident from the data that DistilBERT consistently outperforms the other two models, demonstrating lower processing times across the majority of the tested message block intervals. Particularly at the 15-message block mark, DistilBERT shows a significant advantage in time efficiency, indicating its optimized performance for handling this data size. This suggests that DistilBERT is a more computationally efficient model in this context, offering faster execution. This could be a critical factor in live data, such as social media applications, where processing speed is crucial. The variation in processing speed can be attributed to the differing levels of intricacy in the messages' content, the assortment of vocabulary across the various blocks, and the message lengths within each block. As illustrated in Figure 8a, block 15 demonstrates a more prolonged processing duration than block 6 due to its greater length and the more complex construction of sentences.

6.6.2 Convergence

We now turn to an examination of the convergence characteristics of DistilBERT. To illustrate, Figure 8b plots the evolution of DistilBERT's performance in NMI and Total loss over 500 epochs. As can be seen, the loss exhibits a steady decline across epochs, indicating an improving model fit over time. Correspondingly, the NMI score escalates, achieving its apex within the initial 250 epochs, after which it levels off, demonstrating consistent performance despite further increases in epochs. This pattern underscores the efficiency of DistilBERT in reaching a stable, high-quality representation early in the training process, with sustained performance in subsequent iterations.



(a) Execution time for latest message updating strategy



(b) Epoch vs loss convergence

Fig. 8: Visualizing analysis of execution time and convergence

7 Conclusion

This research examined the difficulty of recognizing social events as they occur, with the extra limitation of progressively expanding the model's knowledge base over time. To tackle this problem, we have proposed a model that leverages structural and semantic information inherent in social messages. DistilBERT-GNN uses dynamic social streams to continuously learn from new incoming data and adapt its parameters accordingly. Through extensive experimentation, we have demonstrated the superiority of our approach over several baseline models, exhibiting higher precision. Our work advances the leading edge in event detection and paves the way for ongoing efforts in this exciting study area. Overall, the experimental outcome average of NMI, AMI, and ARI is 0.72, 0.53, and 0.24, respectively. In future works, we are considering different modalities to capture various social media data.

Declarations

Conflict of interest The authors declare no competing interests.

References

Aggarwal, C. C., & Subbian, K. (2012). Event detection in social streams. *Proceedings of the 2012 SIAM international conference on data mining*, 624–635.

AlSumait, L., Barbará, D., & Domeniconi, C. (2008). On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. *2008 eighth IEEE international conference on data mining*, 3–12.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Bravo-Marquez, F., Mendoza, M., & Poblete, B. (2014). Meta-level sentiment models for big social data analysis. *Knowledge-based systems*, *69*, 86–99.

Cao, Y., Peng, H., Wu, J., Dou, Y., Li, J., & Yu, P. S. (2021). Knowledge-preserving incremental social event detection via heterogeneous gnns. *Proceedings of the Web Conference 2021*, 3383–3395.

Danday, A., & Murthy, T. S. (2022). Twitter data analysis using distill bert and graph based convolution neural network during disaster.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dwivedi, Y. K., Ismagilova, E., Hughes, D. L., Carlson, J., Filieri, R., Jacobson, J., Jain, V., Karjaluoto, H., Kefi, H., Krishen, A. S., et al. (2021). Setting the future of digital and social media marketing research: Perspectives and research propositions. *International journal of information management*, *59*, 102168.

Fedoryszak, M., Frederick, B., Rajaram, V., & Zhong, C. (2019). Real-time event detection on social data streams. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2774–2782.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, *18*(5-6), 602–610.

Gumaei, A., Al-Rakhami, M. S., Hassan, M. M., De Albuquerque, V. H. C., & Camacho, D. (2022). An effective approach for rumor detection of arabic tweets using extreme gradient boosting method. *Transactions on Asian and Low-Resource Language Information Processing*, *21*(1), 1–16.

Hu, L., Zhang, B., Hou, L., & Li, J. (2017). Adaptive online event detection in news streams. *Knowledge-Based Systems*, *138*, 105–112.

Hu, Z., Dong, Y., Wang, K., & Sun, Y. (2020). Heterogeneous graph transformer. *Proceedings of the web conference 2020*, 2704–2710.

Ilina, E., Hauff, C., Celik, I., Abel, F., & Houben, G.-J. (2012). Social event detection on twitter. *Web Engineering: 12th International Conference, ICWE 2012, Berlin, Germany, July 23-27, 2012. Proceedings 12*, 169–176.

Kleinberg, J. (2002). Bursty and hierarchical structure in streams. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 91–101.

Kolajo, T., Daramola, O., & Adebiyi, A. A. (2022). Real-time event detection in social media streams through semantic analysis of noisy terms. *Journal of Big Data*, *9*(1), 90.

Kusal, S., Patil, S., Choudrie, J., Kotecha, K., Vora, D., & Pappas, I. (2022). A review on text-based emotion detection–techniques, applications, datasets, and future directions. *arXiv preprint arXiv:2205.03235*.

Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015). From word embeddings to document distances. *International conference on machine learning*, 957–966.

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and word2vec for text classification with semantic features. *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, 136–140.

Liu, B., Han, F. X., Niu, D., Kong, L., Lai, K., & Xu, Y. (2020). Story forest: Extracting events and telling stories from breaking news. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *14*(3), 1–28.

Luo, H., Cai, M., & Cui, Y. (2021). Spread of misinformation in social networks: Analysis based on weibo tweets. *Security and Communication Networks*, *2021*, 1–23.

Mathioudakis, M., & Koudas, N. (2010). Twittermonitor: Trend detection over the twitter stream. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 1155–1158.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, *26*.

Mredula, M. S., Dey, N., Rahman, M. S., Mahmud, I., & Cho, Y.-Z. (2022). A review on the trends in event detection by analyzing social media platforms' data. *Sensors*, *22*(12), 4531.

Murshed, B. A. H., Mallappa, S., Abawajy, J., Saif, M. A. N., Al-Ariki, H. D. E., & Abdulwahab, H. M. (2023). Short text topic modelling approaches in the context of big data: Taxonomy, survey, and analysis. *Artificial Intelligence Review*, *56*(6), 5133–5260.

Nguyen, V. Q., Anh, T. N., & Yang, H.-J. (2019). Real-time event detection using recurrent neural network in social sensors. *International Journal of Distributed Sensor Networks*, *15*(6), 1550147719856492.

Olaleye, T., Olaleye, A., Ofoegbunam, E., Abodunrin, G., Abioye, T., & Ahiara, W. C. (2022). Opinion mining analytics of iot ecosystem by profile of mood state with logistic regression. *2022 5th Information Technology for Education and Development (ITED)*, 1–7.

Ozdikis, O., Karagoz, P., & Oğuztüzün, H. (2017). Incremental clustering with vector expansion for online event detection in microblogs. *Social Network Analysis and Mining*, *7*, 1–17.

Peng, H., Li, J., Gong, Q., Song, Y., Ning, Y., Lai, K., & Yu, P. S. (2019). Fine-grained event categorization with heterogeneous graph convolutional networks. *arXiv preprint arXiv:1906.04580*.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). Icarl: Incremental classifier and representation learning. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Ren, J., Jiang, L., Peng, H., Cao, Y., Wu, J., Yu, P. S., & He, L. (2022). From known to unknown: Quality-aware self-improving graph neural network for open set social event detection. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1696–1705.

Slim, H., Belouadah, E., Popescu, A., & Onchis, D. (2022). Dataset knowledge transfer for class-incremental learning without memory. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 483–492.

Sufi, F. K., & Khalil, I. (2022). Automated disaster monitoring from social media posts using ai-based location intelligence and sentiment analysis. *IEEE Transactions on Computational Social Systems*.

Sun, B., Wei, X., Cui, J., & Wu, Y. (2023). Social activity matching with graph neural network in event-based social networks. *International Journal of Machine Learning and Cybernetics*, *14*(6), 1989–2005.

Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2019). Deep graph infomax. *ICLR (Poster)*, *2*(3), 4.

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., & Yu, P. S. (2019). Heterogeneous graph attention network. *The world wide web conference*, 2022–2032.

Yang, C., Xiao, Y., Zhang, Y., Sun, Y., & Han, J. (2020). Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, *34*(10), 4854–4873.

Yang, Y., Pierce, T., & Carbonell, J. (1998). A study of retrospective and on-line event detection. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 28–36.

Yuan, K., Xu, W., Li, W., & Ding, W. (2022). An incremental learning mechanism for object classification based on progressive fuzzy three-way concept. *Information Sciences*, *584*, 127–147.

Yun, S., Jeong, M., Kim, R., Kang, J., & Kim, H. J. (2019). Graph transformer networks. *Advances in neural information processing systems*, *32*.

Zhang, C., Song, D., Huang, C., Swami, A., & Chawla, N. V. (2019). Heterogeneous graph neural network. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 793–803.

Zhang, K., Zi, J., & Wu, L. G. (2007). New event detection based on indexing-tree and named entity. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 215–222.

Zhou, S., Kan, P., Huang, Q., & Silbernagel, J. (2023). A guided latent dirichlet allocation approach to investigate real-time latent topics of twitter data during hurricane laura. *Journal of Information Science*, *49*(2), 465–479.

Zola, F., Segurola-Gil, L., Bruse, J. L., Galar, M., & Orduna-Urrutia, R. (2022). Network traffic analysis through node behaviour classification: A graph-based approach with temporal dissection and data-level preprocessing. *Computers & Security*, *115*, 102632.