The VLDB Journal (2025) 34:9
https://doi.org/10.1007/s00778-024-00878-5

REGULAR PAPER

A survey of multimodal event detection based on data fusion

Manuel Mondal[1] · Mourad Khayati[1] · Hông-Ân Sandlin[2] · Philippe Cudré-Mauroux[1]

Received: 21 July 2023 / Revised: 7 October 2024 / Accepted: 10 October 2024 / Published online: 13 December 2024

Abstract

With the emergence of the Internet of Things (IoT) and the rise of shared multimedia content on social media networks, available datasets have become increasingly heterogeneous. Several multimodal techniques for detecting events in data of different types and formats have emerged. Those techniques implement various detection algorithms and present different trade-offs in terms of data fusion. Unfortunately, little is known about their underlying detection mechanisms, as existing comparisons are limited to either unimodal event detection techniques or specific types or representations for multimodal techniques. Understanding the behavior of multimodal event detection techniques remains an acute open research problem. In this work, we present a systematic literature review of multimodal event detection techniques. We describe how various techniques leverage information from different modalities through data fusion. We further propose a novel taxonomy of multimodal event detection techniques according to their temporal orientation and the inner workings of their detection mechanism. Finally, we analyze the datasets and metrics used in previous works as well as their reported results. Our survey allows to uncover the properties of each approach and discuss future research directions in this field.

Keywords Event detection · Data fusion · Multimodal representation · Heteregenous data

1 Introduction

Real-world events are often captured as data that unfold over time and space [3, 129]. They are inherently temporal, in the sense that they occur at various time points, and also often change in nature over time [119, 127]. The automatic identification of event occurrences–known as event detection–has recently gained much attention in the research community. This task has become a fundamental operation in many applications such as identifying real-time events on social media platforms [95, 124], monitoring critical infrastructure [19, 86], detecting acute health events for patients [57, 63, 101], predicting imminent cybersecurity hazards [76, 115], or mining multimedia archives for potential events of interest [22, 122, 125].

Event detection is commonly applied to homogeneous datasets of a single type such as time series, textual content, audio and video recordings, geographic coordinates, images, or social interactions. Such unimodal detection aims to retrieve either specific events by matching a query to a pattern or by identifying open-ended events by searching for statistical anomalies. A large number of techniques have been introduced in this context [12, 16, 17, 64, 69, 71, 77, 105, 109, 118]. Several surveys and benchmarks have compared those unimodal event detection techniques in multiple domains ranging from social media networks [6, 31, 44, 46, 51, 54, 79, 83, 123] to wireless sensor networks [50, 82], audio streams [8, 25, 112], and biomedical signals [49, 64].

With the emergence of the Internet of Things (IoT) and the growing trend of sharing multimedia content on social networks in real-time, available datasets have become increasingly heterogeneous, offering access to multiple data modalities at a time. Those modalities emanate either from the observed instances themselves (e.g., a social media post composed of both image and text) or by merging data from different sources (e.g., monitoring a patient's state by

✉ Mourad Khayati
mourad.khayati@unifr.ch

Manuel Mondal
manuel.mondal@unifr.ch

Hông-Ân Sandlin
hongan.sandlin@ar.admin.ch

Philippe Cudré-Mauroux
philippe.cudre-mauroux@unifr.ch

[1] University of Fribourg, Fribourg, Switzerland

[2] armasuisse Science and Technology, Thun, Switzerland

combining physiological measurements and textual medical records).

Multimodal event detection techniques attempt to leverage information gathered from several heterogeneous sources jointly. They have been applied to solve a myriad of real-world problems. In the medical domain, for instance, Chen et al. [26] combine clinical notes with physiological sensor readings to alert for imminent health events. Another example is social networks, where Brenner et al. [22] use query-based event detection by merging images, metadata, and text. A third example is road networks where Pan et al. [89] combine social media messages with mobility data to detect traffic events.

One of the most critical challenges in multimodal event detection is combining multiple data types into a uniform representation. This merging process is called *data fusion*. According to Gao et al. [43], data fusion aims to "integrate the data of different distributions, sources, and types into a global space in which both intermodality and cross-modality can be represented in a uniform manner". This is often achieved by leveraging the information each modality provides, as well as the cross-modality interactions between data types. While handling several modalities is technically more complex than homogeneous event detection, it has at least two main advantages.

First, some events are imperceptible when modalities are treated individually but can be effectively detected using multiple modalities. One example in that context is detecting medical events while recording patients' real-time physiological measurements and storing their health assessments in textual clinical notes. Relying on the sensor readings solely without taking into account the health context of the patient stored in the clinical notes might result in missing important events [48, 63].

Second, erroneous event detection can often be avoided by considering additional modalities. In the same medical example, a physiological sensor can be temporarily faulty and cause an anomaly in the signal, even if no event is occurring as confirmed by additional sensors and modalities [28, 53]. Relying on multiple modalities can, therefore, reduce the occurrence of false positives.

Ultimately, the performance of multimodal techniques heavily depends on the underlying data fusion method they use. Existing comparison studies on data fusion, such as [139], are generic in the sense that they do not focus on event detection. Hence, they do not clearly represent the intricate differences between event detection categories that have been recently introduced. Unlike those works, we identify three different categories of data fusion mechanisms depending on the stage they have been applied. Data characterization-based fusion extracts the features from each modality and combines them directly. Transformation-based fusion is achieved by training a model that maps each modality into latent representations, which are combined in a later stage. Lastly, decision-based fusion applies event detection to each modality separately and integrates the individual event detection results to obtain a combined score.

This diversity in those fusion mechanisms calls for a comparative study of event detection techniques. A handful of surveys have been proposed to achieve this goal, such as Hu el al. [54] and Xiao et al. [126]. They, however, restrict the definition of event detection to a single task and, thus, overlook a large body of techniques that represent the different tasks we introduce later in the paper. Additionally, those surveys do not include the temporal dimension in their analysis and focus only on the techniques that detect events in the past. By doing so, they omit several techniques that aim to detect events occurring in real-time or in the imminent future. To the best of our knowledge, no prior work has compared multimodal event detection based on these two criteria. This survey aims to bridge this gap.

Specifically, we propose a new taxonomy of event detection techniques based on their temporal orientation, further distinguishing different families of approaches within each category. Additionally, we report the detection results as introduced by the authors of those techniques and describe the datasets and metrics commonly used in this field. We discuss the limitations of those techniques in terms of datasets and metrics and highlight their domain applicability.

Lastly, we present the recurring applications of multimodal detection, including social event detection, medical event detection, and multimedia event detection. We point out research problems that remain unsolved and present potential remedies. In particular, we discuss the need to benchmark the quality and efficiency performance of existing techniques and the limitations of the available datasets. We also discuss emerging trends in IoT and hardware acceleration for multimodal processing and uncover future event detection use cases.

The rest of this paper is organized as follows. After introducing some terminology and background related to event detection in Sect. 2, we lay the groundwork for the various data fusion mechanisms in Sect. 3. Section 4 discusses the different classes of multimodal event detection and groups them into retroactive, real-time, and forecasting techniques. An overview of evaluation metrics and datasets is provided in Sect. 5. Finally, applications and future opportunities are presented in Sects. 6 and 7, respectively.

2 Event detection

The definition of event detection varies broadly across the literature, depending on the task at hand. Table 1 provides a comparative summary of existing techniques, highlighting how each technique complements the event detection

Table 1 Summary of the surveyed multimodal event detection techniques

Technique	Task	Use-case	Properties
Cecaj et al. [24]	Anomaly	Identifying unusual social activity	Detects anomalies using distinct geographic regions
Han et al. [48]	Anomaly	Identifying unusual social activity	Detects anomalies based on topic and location clusters
Banerjee et al. [9]	Anomaly	Identifying unusual social activity	Hidden Markov Modeling-based detection of anomalies
Rodrigues et al. [106]	Anomaly	Identifying unusual social activity	Deep learning-based prediction combining text and mobility data
Ould et al. [86]	Anomaly	Infrastructure monitoring	Computes a geographically weighted voting scheme on a wireless sensor network
Chen et al. [27]	Anomaly	Social media monitoring	Identifies anomalous subgraphs on an ad-hoc heterogeneous social graph
Khadanga et al. [63]	Anomaly	Clinical monitoring	Deep learning-based joint modeling of sensor data and clinical notes
Pan et al. [89]	Anomaly	Traffic monitoring	Models traffic as activity on a graph and combines it with social media data
Yilmaz et al. [132]	Labeling	Social media mining	Estimates the latent variables underlying distributions of tweets
Cai et al. [23]	Labeling	Social media mining	Uses topic models to represent events as mixtures of multimodal distributions
Peng et al. [94]	Labeling	Social media mining	Builds a message-based Heterogeneous Information Network (HIN)
Ghaemi et al. [45]	Labeling	Social media mining	Extends DBSCAN to handle geospatial heterogeneity
Oh et al. [84]	Labeling	Video archive mining	SVM-based feature learning of visual and audio streams
Yang et al. [131]	Labeling	Image archive mining	Semi-supervised learning-based fusion and DBSCAN-based clustering
Ma et al. [78]	Labeling	Image archive mining	Extends SVD using similarity-based adjacency matrices
Petkos et al. [97]	Labeling	Image archive mining	Learns similarity measures across modalities using spectral clustering
Petkos et al. [99]	Labeling	Image archive mining	Applies community detection on a similarity graph

Table 1 continued

Technique	Task	Use-case	Properties
Wang et al. [122]	Labeling	Image archive mining	Incremental clustering on learned representations of a social interaction graphs
Yang et al. [130]	Labeling	Image archive mining	Graph matrix factorization technique applied on a similarity graph
Reuter et al. [104]	Labeling	Image stream monitoring	Compares streaming instances to representatives of known event classes
Peng et al. [95]	Labeling	Social media monitoring	Extends the HIN-based approach to handle streaming event detection
Chen et al. [28]	Labeling	Traffic monitoring	Semi-supervised deep learning event classification
Chen et al. [26]	Labeling	Clinical monitoring	Tree-based prediction combining patient information and sensor data
Horng et al. [53]	Labeling	Clinical monitoring	SVM-based prediction combining patient information and sensor data
Yang et al. [128]	Labeling	Clinical monitoring	Deep learning-based prediction combining patient information and sensor data
Brenner et al. [22]	Query	Text search in image archives	SVM-based feature learning and event retrieval
Elhoseiny et al. [38]	Query	Text search in video archives	Embeds videos and queries into the same distributional semantic space
Wu et al. [125]	Query	Text search in video archives	Projects video and query embeddings into a common lexicon space
Younessian et al. [133]	Query	Text search in video archives	Retrieve events based on similarity of video features and queries

landscape. The columns “tasks”, “use-case”, and “properties” characterize each work by the type of tasks it supports, its application domain, and the underlying event detection mechanisms. All surveyed approaches tackle one of the event detection definitions introduced below, which determines the overarching objective of their technique: detecting anomalies, labeling instances into meaningful groups of events, or retrieving event-related instances based on queries.

We describe the properties of each technique in more detail later and summarize the main categories of event detection techniques according to the temporal dimension. We use this dimension to organize the surveyed multimodal event detection techniques.

2.1 Event detection tasks

We distinguish three recurrent tasks widely explored in multimodal event detection. Below, we introduce those tasks and provide a formal definition for each. Before describing in detail the tasks, we provide a brief overview of the different ways event detection is commonly perceived in the literature.

In the first conception of event detection, techniques aim at identifying which instances in a dataset deviate from the norm and thereby indicate that an unusual event is occurring. A second perspective assumes that all data instances pertain to an event and aims to find which event each instance belongs to. The third notion consists of retrieving the instances corresponding to an event description provided as a textual query.

2.1.1 Anomaly identification

Anomaly detection techniques compute an expected baseline of activity during normal runtime (e.g., usual sensor readings) and trigger the detection of an anomalous event whenever the behavior diverges significantly. Only a subset of all instances belong to an event; a technique will identify this subset as accurately as possible.

Definition 1 Identifying anomalies in a dataset consists of distinguishing between normal and anomalous instances. Formally, given a dataset of multimodal data instances $I = \{i_1, i_2, \ldots, i_n\}$, the event detection is a surjective function that identifies all the data instances associated with an anomalous event, i.e., $f : I \twoheadrightarrow \{0, 1\}$.

Existing works use one of two implementations to address Definition 1. The first set of techniques models the expected values when no event occurs and identifies statistical anomalies deviating from these baselines. Approaches [24, 48, 86] establish the statistical baseline with respect to the location the instances originate from. The authors in [27, 89] represent instances as nodes in a graph and identify statistically anomalous subgraphs. The second set of techniques, illustrated by the approaches in [9, 53, 63, 106, 128], train machine learning models to assign instances into either the normal or anomalous category.

2.1.2 Data classification/clustering

The second task assumes that every observed instance in a dataset pertains to an event. The event detection then assigns the appropriate event label to each instance. Depending on whether the types of events are specified in the dataset, the event labeling task is addressed using either classification models or clustering techniques.

Definition 2 Labeling events in a dataset consists of assigning an event label to each instance. Formally, given a dataset of multimodal data instances $I = \{i_1, i_2, \ldots, i_n\}$ and event labels $L = \{l_1, l_2, \ldots, l_k\}$, the event detection can be defined as a surjective function that assigns for each instance an event label $l_e \in L$, i.e., $g : I \twoheadrightarrow L$.

Similarly to the first task, Definition 2 is implemented in two variants, depending on whether the event labels are defined (supervised) or not (unsupervised). In a supervised setting, a list of event labels is pre-defined, e.g., $L = \{$“protest”, “accident”, etc.$\}$. The techniques introduced by the authors in [84, 94, 104] address this task by first training multi-class classification models. The event detection then consists of labeling each unseen instance with its corresponding event class.

When the list of labels is unknown beforehand, techniques use unsupervised approaches. The techniques presented in [45, 78, 95, 97, 99, 122, 130–132] use either clustering algorithms or graph community detection to identify groups of instances. Each instance is thus assigned to one of the unnamed event clusters $l_e \in L$.

2.1.3 Query retrieval

In the third task, events of interest are described by textual queries. Those queries can range from specific formulations, such as “Equifax 2017 data breach”, to generic event categories, such as “Public protest”. The provided queries need to be interpreted first, after which the inferred events are detected in the available instances.

Definition 3 Query-based event detection retrieves instances matching a textual query describing an event. Formally, given a dataset of instances $I = \{i_1, i_2, \ldots, i_n\}$ and a query q, the event detection function $h : (I, q) \rightarrow [0, 1]$ finds instances $i \in I$ corresponding to query q.

The authors in [22, 38, 125, 133] implement this definition by developing ranking processes, which compare the input queries with the training instances and their (potential) event labels. The models learn meaningful representations of the instances to perform the comparison and are trained to embed textual queries into the same embedding space. The event-related instances matching the queries in this space are then returned.

2.2 Event detection categories

As mentioned earlier, the temporal dimension is an inherent property of any event detection technique. The proposed approaches widely differ depending on whether the examined events occurred before the launch of the event detection system, during its deployment, or after its computation. Techniques can thus be categorized according to the temporal dimension into three classes: (i) *retrospective* historical techniques, (ii) *real-time* techniques, and (iii) *prospective* forecasting techniques.

Figure 1 introduces our taxonomy that describes this categorization. In the first level of the tree, event detection techniques are split according to their temporal orientation. In the second, techniques are further categorized into families of approaches. We briefly introduce each of those families below and discuss them in further detail in Sect. 4.

Retrospective Detection. Techniques from this class aim at detecting events *a posteriori*, based on historical data. Detecting events retrospectively generally resorts to either mining historical baselines to find and highlight anomalous instances or discovering clusters of instances that pertained to the same event. Detecting events in past data is frequently used to retrieve instances pertaining to a known event in large multimedia collections.

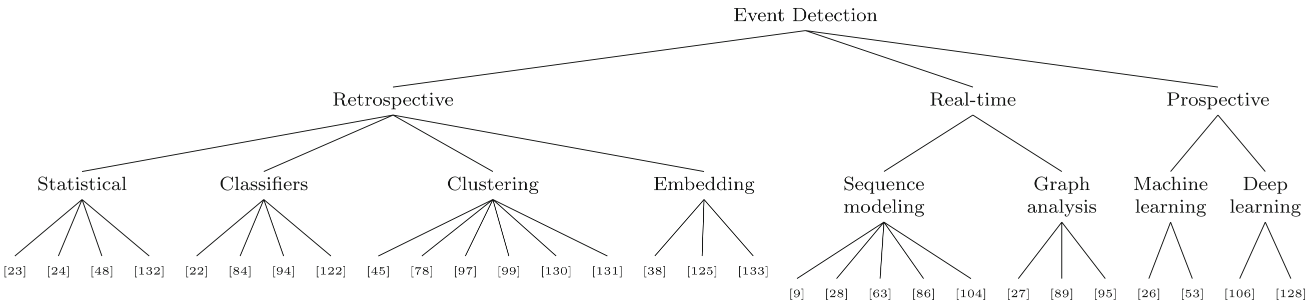

Fig. 1 Taxonomy of event detection techniques using data fusion

Real-time Detection. Techniques detecting events *at run-time* generally follow a two-stage process. First, they make use of past data to extract expected baselines or to pre-train models. Event detection is subsequently performed incrementally on incoming streaming data to detect anomalous behavior. The objective of such techniques is often the monitoring of physical systems or streams of social media content.

Prospective Detection. Forecasting the occurrence of *future events* is a natural extension of event detection in real-time. Events may be preceded by leading indicators, such as causally linked precursory events, or may follow seasonality or cycles. Predicting events often boils down to effectively detecting specific temporal patterns and chains of events in the data. Such techniques are often developed for the medical domain, where physiological measurements can be used to predict health events.

3 Data fusion

Data fusion is the cornerstone of multimodal event detection techniques that allows them to integrate information from more than one modality. The studied techniques adopt one of three data fusion mechanisms. Those mechanisms vary depending on when they occur in the event detection pipeline. Data characterization-based approaches operate at the feature level, transformation-based approaches learn the encoding of the features and fuse them at the representation level, and decision-based approaches aggregate detection scores. We discuss each of those mechanisms in detail below.

3.1 Data characterization-based fusion

Fusion based on data characterization is implemented by extracting key features from each modality and combining them into a joint representation. This concept is similar to the Extract, Transform, Load (ETL) process used in other fields such as data integration [2, 62] or warehousing [113]. The data characterization-based fusion mechanisms are composed of three stages. First, features are manually extracted from each modality. Second, the selected features are processed by applying normalization, scaling them to the same magnitude, or projecting them into the same vector space. Third, the modalities are combined based on the extracted characteristics into a fused data format.

For each data modality, there are several possible characterizations. Commonly used features when handling text are bag-of-words [131], tf-idf [104], bigram detection [53], keyword extraction [26, 94], and pre-trained word embeddings [48]. When considering time series, several techniques select summary features, such as the minimum, maximum, and average (e.g., [26, 53]). Image and video features are generally based on object recognition (e.g., [9]) or visual similarity (e.g., [131]).

Since the features extracted from the input modalities are usually of different scales, several techniques apply rudimentary preprocessing, such as normalization [53]. Other proposed fusion mechanisms (e.g., [78, 104, 131]) process features by computing a similarity score within each modality. The similarity in the textual content is commonly computed with the Jaccard index [41] between word occurrences or the cosine similarity of their tf-idf vectors [117]. Location features are often compared using the Haversine distance [51] between geographic coordinates and images using perceptual hash values [135]. All similarity scores are then scaled and used as the new feature values.

We detail the feature extraction and processing of a single modality with the example of text in Fig. 2. The words of a document are first embedded using an off-the-shelf text embedding library (e.g., fastText [21]). Duplicates are removed, and a dimensionality reduction technique, e.g., Principal Component Analysis (PCA) [37], is applied to obtain a preprocessed feature vector of the original document.

To combine the preprocessed features of the different modalities, some techniques opt for a simple concatenation of the values into a fused feature vector (e.g., [26, 53, 104, 131]). Alternatively, in [94], the authors combine the features extracted from multimodal social media messages by encoding the multimodal instances into a heterogeneous graph representation. They first extract entities from the messages (e.g., keywords, users, topics, etc.) and treat each entity as a

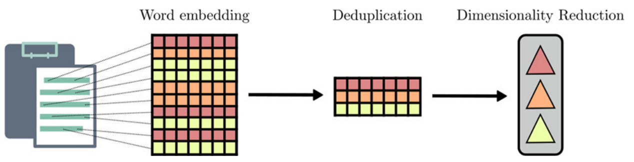

Fig. 2 Feature extraction depicted with the textual modality. Words are embedded, duplicates are removed, and a dimensionality reduction technique is applied to obtain a feature vector that encodes the text document

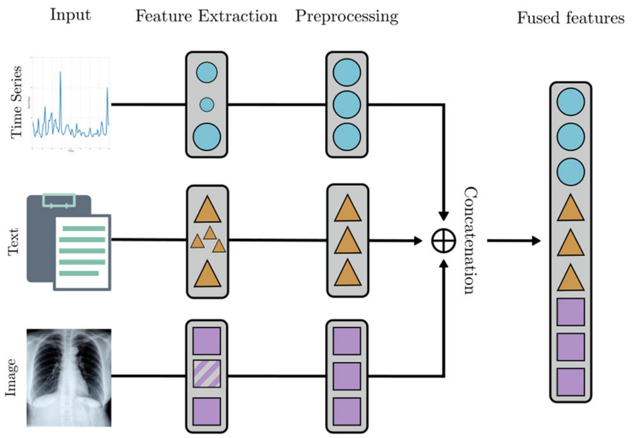

Fig. 3 Data characterization-based Fusion. A medical record composed of a time series, a clinical note, and an X-ray image is fused by (i) extracting features, (ii) applying preprocessing, and (iii) concatenating the features of each modality into a common feature vector

node. In the second step, the authors create edges between the nodes whenever the features of any modality are similar (e.g., co-occurring words, nearby locations, similar timestamps, etc.).

Figure 3 illustrates a complete example of Data Characterization-based fusion. A patient's data is composed of physiological sensor readings (as time series), clinical notes (as text), and an X-ray (as image). Time series features (e.g., minimum and maximum), text features (e.g., term frequency-inverse document frequency: tf-idf), and image features (e.g., recognized objects) are extracted, preprocessed (through scaling, deduplication, and inference of missing values), and concatenated into a fused feature vector. The event detection will take this fused vector as its input.

3.2 Transformation-based fusion

Transformation-based fusion mechanisms include a representation learning component, first trained on labeled data to transform each modality into an internal latent representation. The data fusion occurs only in the second step, by combining the transformed representations. In the representation learning setting, the learned models are trained in a supervised manner to select and transform the input features into high-level abstractions [121]. These abstract features, also known as latent representations, are then used to perform downstream tasks, such as event detection.

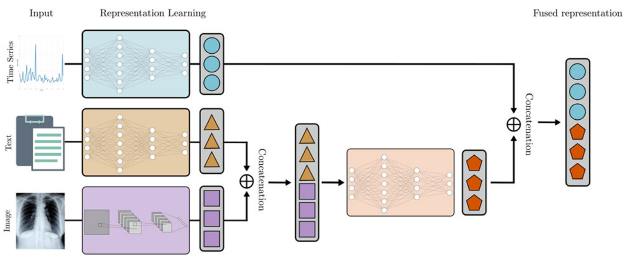

Fig. 4 Transformation-based fusion. A representation learning component (depicted with three neural networks) embeds a time series, a textual note, and an image separately. The transformed features are combined by concatenation

All the features of a data instance are typically encoded into a single vector, fed into a representation learning model. This model applies a series of parameterized transformations to the computed vector and produces the latent representation. The obtained representation vector is then used to perform a downstream task, and the performance of this task is used to update the parameters of the transformations. Representation learning is widely used to handle data in any modality, such as text, images, and sound [15].

Multimodal representation learning is achieved using neural network architectures, with distinct models for each modality. These models transform the input data into latent representations, which are then fused. For instance, the authors in [28, 63, 128] transform time series and text by training two deep neural networks to produce distinct latent vectors. These transformed representations are then fused, either by simple concatenation or using an additional fully connected neural network layer [128]. This architecture allows for an end-to-end pipeline, integrating the transformation-based fusion with the downstream event detection task. During the training phase of the neural networks, a backpropagation step is iteratively applied to improve the learned transformation functions.

Figure 4 illustrates this approach using the same medical example as in Fig. 3. Three neural networks are first trained to learn to represent each modality. In this example, a recurrent model produces an embedding of the time series. The textual note and image are encoded (using a fully connected and convolutional network) and are combined by another fully connected model into a latent representation. This intermediate representation is finally fused with the time series embedding by concatenation to produce the transformed representation vector.

An alternative approach to transformation-based data fusion mechanisms is used when the transformed features are manually selected. The authors in [97, 99] first choose features from the metadata of images (e.g., temporal infor-

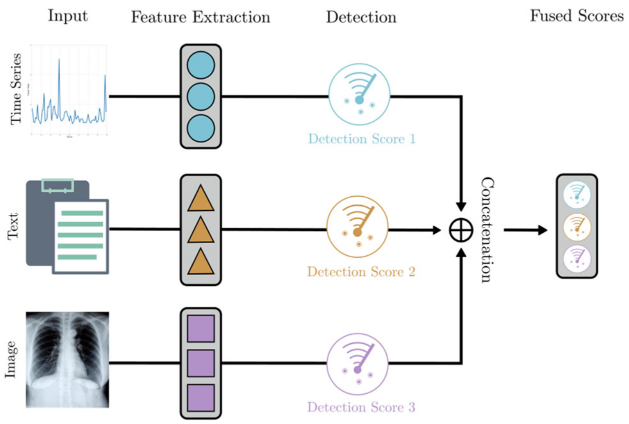

Fig. 5 Decision-based fusion. Three distinct event detection scores are computed on a time series, a textual document, and an image. The individual scores are combined into a fused vector

mation, geolocation, textual descriptions, etc.). Then, they compute pairwise similarities within each modality. The representation learning occurs on these extracted features. It constitutes a set of Support Vector Machines (SVMs), each trained to assign images to event clusters. The result of each SVM is finally concatenated into a new fused representation vector [97] or transformed into a graph [99].

3.3 Decision-based fusion

Decision-based fusion consists of fusing event detection scores, such as labels or probabilities, obtained from each modality separately. To combine the results at the decision level, data fusion mechanisms compute averages of the individual scores or train a model by taking the individual scores as input and producing a fused result.

Several approaches are proposed to combine distinct detection scores. Some authors (e.g., [125]) find that simply averaging the decisions of each modality yields satisfactory results. Another type of solutions infuses the combination of individual scores with information from one of the modalities. For instance, the authors in [86] compute a weighted sum of anomaly scores, where the weights are proportional to the distance between sensors in a network. Finally, the authors in [84, 133] use ensemble learning. They develop sophisticated models with learnable parameters, fine-tuned on labeled examples, to combine the scores of several base models.

Figure 5 illustrates decision-level fusion with the same medical example introduced previously. For each modality, an event detection score is produced separately. The three detection scores are then concatenated into a fused detection score vector and passed to the downstream model.

All the data fusion techniques introduced above are compatible with each of the three categories of multimodal event detection. Any data fusion mechanism can combine the different modalities in each of the three categories. The classification scheme we propose in this survey distinguishes, first and foremost, between techniques aimed at retrospective, real-time, or prospective event detection.

4 Multimodal event detection techniques

In this section, we introduce existing multimodal event detection techniques. For each described technique, we discuss how the data fusion mechanisms exploit information from the modalities to detect events.

Table 2 provides a comparative summary of existing techniques, highlighting how our survey dissects the event detection field. The column "modality" indicates which data types are fused by each event detection technique. Each examined technique uses one of four internal latent data representations: time series, graphs, learned embeddings, or tabular values. One of the three "fusion mechanisms", introduced in Sect. 3 (data characterization-based fusion, transformation-based fusion, or decision-based fusion), is used to combine the different modalities.

4.1 Retrospective event detection

The first category of techniques we consider encompasses approaches to detecting events in past data. Within this category, we distinguish four families of approaches. (1) Statistical techniques detect events using probability distribution and topic modeling. (2) Classification model techniques assign a class from a pre-defined list of event types to each data instance. (3) Clustering techniques split all instances into groups pertaining to the same event in an unsupervised manner. (4) Embedding-based approaches retrieve instances matching a textual query through which events are described in natural language.

4.1.1 Statistical techniques

The techniques from this category use statistical and probabilistic modeling to detect events. A first set of approaches computes expected baselines and detects events by identifying statistical anomalies deviating from this baseline. A second set uses probabilistic latent variable models to detect events.

Cecaj et al. [24] consider the problem of event detection by fusing temporal and location data. They focus on city life events using call detail records and Twitter messages. Their approach considers detecting events whenever the amount of activity significantly deviates from an established baseline, fusing the modalities at an early stage to take into account both location and temporal data. The fusion is achieved by

Table 2 Overview of multimodal event detection techniques using data fusion

		Technique	Modality							Data representation				Fusion mechanism		
			Time	Text	Image	Audio	Video	Interactions	Location	Time series	Graph	Embed.	Table	Charact.	Transform.	Dec.
Retrospective	Stat.	Cai et al.'15 [23]	✓	✓	✓				✓				✓		✓	
		Cecaj et al.'17 [24]	✓	✓					✓	✓				✓		
		Han et al.'19 [48]	✓	✓					✓	✓				✓		
		Yilmaz et al.'18 [132]		✓					✓				✓		✓	
	Class.	Brenner et al.'12 [22]	✓	✓	✓			✓	✓				✓		✓	
		Oh et al.'14 [84]			✓	✓	✓						✓			✓
		Peng et al.'19 [94]		✓				✓			✓			✓		
		Yang et al.'15 [131]	✓	✓	✓			✓	✓				✓	✓		
	Clust.	Ghaemi et al.'19 [45]		✓					✓				✓		✓	
		Ma et al.'15 [78]	✓	✓	✓			✓	✓				✓	✓		
		Petkos et al.'12 [97]	✓	✓	✓				✓				✓		✓	
		Petkos et al.'14 [99]	✓	✓	✓			✓	✓		✓				✓	
		Wang et al.'12 [122]	✓	✓	✓			✓	✓				✓		✓	
		Yang et al.'17 [130]	✓	✓	✓			✓			✓				✓	
	Embed.	Elhoseiny et al.'16 [38]		✓		✓	✓					✓			✓	
		Wu et al.'14 [125]		✓		✓	✓					✓				✓
		Younessian et al.'12 [133]		✓		✓	✓						✓			✓
Real-time	Sequence mod.	Banerjee et al.'18 [9]	✓	✓			✓		✓	✓				✓		
		Chen et al.'21 [28]	✓	✓					✓			✓			✓	
		Khadanga et al.'19 [63]	✓	✓								✓			✓	
		Ould et al.'11 [86]	✓						✓	✓						✓
		Reuter et al.'12 [104]	✓	✓					✓				✓	✓		
	Graph	Chen et al.'14 [27]	✓	✓				✓	✓		✓				✓	
		Pan et al.'13 [89]	✓	✓					✓		✓				✓	
		Peng et al.'21 [95]	✓	✓				✓	✓		✓				✓	
Prospective	ML	Chen et al.'18 [26]	✓	✓									✓	✓		
		Horng et al.'17 [53]	✓	✓									✓	✓		
	DL	Rodrigues et al.'19 [106]	✓	✓								✓			✓	
		Yang et al.'21 [128]	✓	✓								✓			✓	

fragmenting the observed area into non-overlapping geographic regions. Within each region, cell phone activity and the number of social media posts are counted, producing time series sampled at one-hour intervals.

Their proposed event detection process consists of establishing baselines of activity within each region when no event is happening. Seasonality effects are accounted for by computing multiple baselines per region (e.g., for weekdays and weekends). The final event detection flags statistical outliers in the time series of each region, based on a significant deviation from the expected baseline.

Han et al. [48] address detecting events in collections of social media posts, taking into account temporal, location, and textual data. Their approach fuses the modalities into internal time series representations and detects events by verifying whether or not they follow a power law distribution. The time series are obtained by counting the number of messages published in close geographic proximity, on similar topics, and during specific time intervals.

The authors use a characterization-based fusion mechanism, extracting information from each modality in three stages. First, the authors handle the textual information by extracting keywords and phrases from Twitter messages. This text is embedded with a popular pre-trained text embedding technique called fastText [21]. The tweets are then aggregated into groups of similar tweets based on their embeddings, using BIRCH [137] as a clustering technique. Second, within each group, the geographic location is considered by splitting the tweets into quad-trees [18]–a recursive decomposition of the space into regions. Finally, the publication time is included for each region by counting the number of tweets published therein. This yields a fused latent representation with a time series for every region. On the obtained time series, the event detection resorts to verifying whether any events follow a power law distribution. To verify this condition, the authors fit a power-law model to each time series and apply the Kolmogorov Smirnov (KS) Test [30]. An event is detected whenever the score is above a user-defined *p*-value threshold of the KS test.

Yilmaz et al. [132] detect events by fusing textual features and geographical coordinates in Twitter networks. They group messages into event clusters using a generative latent variable model. Specifically, for each hashtag, the tweets mentioning it are aggregated to form the set of words used in all these tweets and as a set of geolocation coordinates (whenever available).

The authors then assume that every event spawns multiple hashtags, and the technique learns the underlying data-generating probability distributions of every modality that led to the observed hashtags. An Expectation Minimization (EM) [85] approach is used to find these statistical parameters by modeling the word (and location) distribution of every event over the whole vocabulary (respectively over the possible locations). This optimization framework allows the extraction of the mixture coefficients for every hashtag. Events are finally described as clusters of hashtags, obtained using the popular *k*-means clustering algorithm [81].

Cai et al. [23] highlight the lack of attention given to images in existing event detection approaches for social media. They propose an event detection technique fusing images with textual features, geographical coordinates, and temporal information. The authors propose STM-TwitterLDA, a multimodal extension of Latent Dirichlet Allocation (LDA) [136]. This approach allows LDA to target spatiotemporal multimodal data from social media.

To include the image data in the proposed LDA model, visual features are extracted from the images using Convolutional Neural Networks (CNNs) [70]. These embeddings are then fused with other modalities by modeling each topic as a mixture of four distributions: hashtags, text, timestamps, and image representations. Events are detected among the collection of tweets by modeling topics according to each modality with this LDA, parametrized with Expectation Maximization (EM) parameter estimation [5], and finally labeling the tweets with this topic model.

4.1.2 Classification-based techniques

Classification-based event detection techniques use supervised and semi-supervised Machine Learning (ML) algorithms. They assume a dataset in which instances are labeled with event classes, and the techniques are trained to assign event labels to unseen instances.

Oh et al. [84] address the task of detecting events in video collections, fusing information from the audio and video streams. Their event detection assigns an event class from a predefined list to each video. The data fusion intervenes at the decision level by first extracting features, detecting events on each modality separately, and combining the prediction scores. The authors achieve the fusion mechanism by extracting and handling features from the video's audio and image streams at different levels of abstraction.

Audio features, starting at the level of spectrum representations (such as MFCCs [13]), are extracted and represented as bag-of-words. At a higher level, acoustic segment models extract encodings of representative audio segments for the various types of events using Hidden Markov Models (HMM) [102]. Low-level visual features, such as color and geometry histograms, and mid-level features, such as detected objects (e.g., "sailboat"), are represented as sets of bag-of-words. In addition, higher-level concepts, such as recognized scene elements in sequences of video frames (e.g., "flooded streets") are learned in an unsupervised manner with a novel Latent SVM approach. Those features are then used by SVMs to produce an intermediate event classification score for each input modality. These individual scores are combined to obtain a final event classification by considering a maximal figure-of-merit approach and local expert forest models.

Peng et al. [94] address the problem of detecting events in collections of textual social media messages. Their technique fuses textual content with user interaction information and assigns event labels to instances. Data fusion is achieved at the feature level by extracting entities and relationships from the messages to build a heterogeneous graph that takes into account all modalities. The authors first extract so-called event elements (keywords, topics, concepts, users) to create different types of nodes.

To form the graph, nodes of the same type are connected by edges according to the following criteria. Keyword nodes are connected whenever two words are synonymous. Nodes representing topics are connected whenever topics are related according to the topic model hierarchical LDA (hLDA) [20]. Concepts are identified and retrieved from an external knowledge base and connected according to their relationships (e.g., "located-in", "attribute-of", etc.). User nodes are connected if a relationship exists between them. These entities are then fused by constructing a heterogeneous information network (HIN) [116] with the addition of hyper-edges that connect event elements of all types whenever they co-occur in the same message.

Using the HIN, the proposed event detection technique proceeds following a two-step approach. First, a newly defined similarity metric is computed between all messages based on the co-occurrence of event elements and their paths in the HIN. This results in a weighted adjacency matrix, upon which a Graph Convolutional Network (GCN) [60] is trained for node classification, where each class corresponds to a detected event.

Brenner et al. [22] detect events in collaborative image collections by combining images with textual metadata (e.g., descriptions, keywords, titles, etc.), timestamps, and geolocation. They use a classification model to determine whether an instance corresponds to an event class, defined by a textual description of the event. The provided event descriptions are enhanced with external information: topic and keywords are used as queries on knowledge bases (WordNet [40] and DBpedia [7]) and location names are converted with geocoding services (GeoNames and Google Geocoding API).

The images are filtered according to their textual annotations by applying a binary linear SVM to determine whether they pertain to an event described in the textual query. Low-level visual features (color and edge) are extracted from the remaining images to obtain their vector representations. In the final step, these representations and the textual event descriptions are passed to a binary classifier, determining whether the image corresponds to the described event.

In between the supervised and the fully unsupervised setting, Yang et al. [131] propose a semi-supervised approach to mine events from collaborative image collections. Their approach fuses images, time, geolocation, tags, and user information. In this semi-supervised setting, only a subset of the instances are labeled with an event class. The authors use the information in the partially labeled subset to guide the event detection clustering process for the remaining instances.

The data fusion process occurs at the feature level by computing similarity scores between labeled and unlabelled instances. To characterize unlabeled instances, the authors first select a labeled instance of each known event class to serve as a representative of the future event clusters. Unlabeled instances are then compared to the representative of each cluster, according to each modality. The similarity for the temporal modality is obtained by comparing publication timestamps. Location similarity is computed with the Haversine distance [1] of their geocoordinates. The textual information is compared with the Jaccard index [41] between sets of their annotations tags. A binary score indicates whether the same user uploaded the image. The visual similarity between images is computed by comparing perceptual hash values [135].

The fusion is achieved by concatenating the similarity values for each modality into a latent vector. The event detection on the fused representation is finally achieved with the unsupervised clustering technique DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [42] to obtain event cluster assignments for the unseen images.

4.1.3 Clustering-based techniques

Clustering-based techniques use unsupervised data mining techniques to detect events by aggregating instances into event clusters. There exist two main approaches that have been proposed in this context. The first type integrates existing clustering and community detection algorithms, such as spectral clustering, into their event detection technique. The second type of techniques proposes modifying existing clustering techniques to handle multimodal event detection.

Wang et al. [122] investigate the importance of social interactions for event detection in image collections, fusing images, text, time, location, and social interaction. Their approach learns similarity scores between instances and uses them to detect clusters of events. The authors build a social interaction graph with three node types (tags, images, users) and directed relationships. Edges exist between images and associated tags, between similar tags, between users and images they interact with, and between befriended users.

The social similarity between images is computed using an extension of Random Walk, called Random Walk with Restart (RWR) [74] between all the nodes that are active during the same time window. Feature similarity is obtained by considering the time difference between images, their geographical distance, the number of common tags, and the cosine similarity between the term frequency vectors of their textual descriptions. The proposed event detection technique learns a combined similarity score between images, which fuses these social and feature similarities using SVMs. This score is applied by an incremental clustering technique to group all images into event clusters.

Petkos et al. [97] aim to detect events from collaborative image collections. The authors fuse images, their geolocation, publication times, textual descriptions, and titles to cluster instances into events. The approach integrates a supervisory signal proceeding in two stages. First, they use labeled instances (i.e., images with a known event cluster assignment) to train classifiers and subsequently use the classification results to cluster unknown instances. Specifically, the authors compute pairwise similarities between all instances. They do so for each modality and concatenate the similarity matrices to obtain a feature vector for each pair of instances. A binary SVM classifier is trained on each pair of instances to predict whether they belong to the same event.

The second stage consists of fusing modalities at the representation level of the instances for which no label is provided. Similarly to the previous stage, the authors start by computing pairwise similarities between all unlabeled instances and concatenate the similarity scores of all modalities. The aforementioned binary classifier is then applied to each pair of instances, indicating whether the instances should be assigned to the same cluster for each pair. Concatenating all the predictions obtained for an individual instance yields a binary representation vector for each, which takes into account all modalities. The event detection finally applies a spectral clustering approach to the learned representation vectors to assign each instance to an event cluster.

In a follow-up work [99], Petkos et al. extend the event detection technique [97] introduced above. The authors improve their approach for detecting clusters of events among a collection of annotated images by fusing their geolocation, publication time, and textual descriptions and titles. Similarly to the original work, pairs of instances are first compared along each modality, and a binary classifier is trained on labeled data to learn whether pairs of instances should be assigned to the same cluster. Unlike the original work, the prediction for each pair of instances is used as an adjacency matrix. This way, the modalities are fused by building a graph where instances are connected if the binary classifier predicts that a pair should belong to the same cluster. A community detection algorithm is applied to this graph, yielding the final event clusters.

In [78], Ma et al. propose SVDMC, a multimodal extension of the matrix decomposition technique Singular Value Decomposition (SVD) [65] for detecting events in collaborative image collections by assigning each image to an event cluster. The authors consider images from Flickr, provided with user information, timestamps, geolocations, and textual tags. The modalities are fused at the feature level after computing similarities between pairs of instances within each modality.

Specifically, the proposed approach encodes each modality into a distinct binary adjacency matrix, where the rows and columns correspond to the images in the dataset. A positive entry indicates a similarity above a fixed threshold between the corresponding images using, for example, cosine similarity between pieces of text. The adjacency matrices of all modalities are fused using a logical addition operation, and SVD is finally applied to obtain a low-dimensional, fused feature representation. On these vector representations, events are detected using k-means by grouping messages into event clusters. The approach is compared against other clustering algorithms (such as [10, 32, 99] and [134]).

Yang et al. [130] process collections of images with metadata, such as textual descriptions, user identification, and geolocation. The authors propose a novel combination of DBSCAN and k-means to detect events. The data fusion mechanism merges the modalities at the feature level by computing similarities between instances. The authors use an exponentially decaying function (Heat Kernel Weighting [68]) for time differences, the Jaccard index for tag overlaps, and the Haversine distance for coordinates. The similarities are computed between instances and the dictionary entry of each modality (derived from a subset of labeled examples). This results in a similarity vector per instance for each modality. Those representations are combined using a multimodal soft-voting strategy, selecting the most salient features across all modalities and producing a fused feature matrix.

In the second step, the authors build graph representations based on the fused feature matrix. They propose a novel dual graph regularized non-negative matrix factorization approach, which learns a dense graph representation of each instance. Upon these representations, their novel hybrid clustering algorithm is applied to produce the final event clusters.

Ghaemi et al. [45] detect events in collections of social media messages. The authors highlight that the standard density clustering algorithm DBSCAN suffers from reduced clustering quality when dealing with spatial heterogeneity, i.e., when the density of the phenomenon significantly fluctuates across the observed regions. Instead, they propose an extension of DBSCAN called Varied Density-based spatial Clustering for Twitter data (VDCT). The authors address this limitation by clustering messages according to geographic proximity and content similarity. To compute location similarity, they consider the Euclidean distance between two tweets' locations and use exponential spline interpolation to determine the search radiuses for the clustering; for text similarity, they use the cosine similarity between bag-of-words representations of their content.

4.1.4 Embedding-based techniques

Embedding-based techniques handle the task of detecting events related to user-provided event description queries. They operate by embedding textual queries with the training instances in a common latent space. The representations are then compared, and the instances most similar to the query are retrieved. Existing techniques in this context handle mostly video content by fusing visual and audio channels with text queries.

In [125], Wu et al. detect events of interest in video collections. The events are described with a textual query that allows the proposed technique to retrieve the videos most similar to the prompt. Since the search queries are not provided during training, the event detection task can be conceived as a zero-shot classification problem, where every description of an event of interest is considered as a separate class, and instances are to be matched to the class. The data fusion is achieved at the decision level by combining the prediction scores obtained from each modality.

The authors first extract two sets of features from the audio and visual channels. On the former, they apply automatic speech recognition for spoken words in the videos and low-level structural audio feature extraction. Three off-the-shelf concept detectors are applied to the latter's frames, and an optical character recognition system detects written words. Furthermore, low-level visual features, such as motion boundary histograms, are extracted. To handle the event retrieval task, each feature set and the event query are embedded into their own feature space and fused by projecting them into a common high-dimensional lexicon space. Subsequently, the learned feature representations of each modality are compared to an embedding of the textual input query, yielding a similarity measure. The similarity measures are finally aggregated with a linear weighted sum to obtain the final fused similarity score that determines the event classification.

Elhoseiny et al. [38] address the same task of detecting events described by user-provided queries, combining text, video, and audio. They embed videos and textual event description queries into the same distributional semantic space and match similar items. The authors first split the visual and audio channels from the videos into two distinct data streams. They apply optical character recognition on the former and automatic speech recognition on the latter, thereby encoding both channels into textual content. Additionally, they extract visual concepts, such as scenes or actions, from the videos. The modalities are fused at the representation level by jointly learning the embeddings of the detected objects and extracted text. The embedding allows event detection to retrieve a ranked list of instances most similar to the provided event description query.

For the same task, Younessian et al. [133] extract low-level acoustic and visual features, as well as higher-level information, such as textual content on the one hand, namely spoken words using automatic speech recognition, and acoustic scenes, such as "engine noise" or "laugh", on the other. The authors also apply visual concept detectors to the video's image channel, establishing a visual concept signature for each event. Similarity metrics are computed between the provided event description and each representation. These scores are finally fused using a learnable linear combination to obtain the event detection score.

4.2 Real-time event detection

This section encompasses streaming techniques to detect events on-the-fly. We distinguish two classes of approaches: (i) sequential modeling and (ii) approaches relying on graph processing. Techniques from the first category model sequences based on prior data and detect an event whenever new instances exhibit unexpected patterns. Approaches relying on graph analysis detect events by identifying anomalous subgraphs or using graph machine learning methods.

4.2.1 Sequential modeling techniques

Sequential modeling-based methods learn a temporary representation of a data stream, which is updated incrementally as new instances come in. Event detection techniques use sequence models to learn expected baselines and flag deviations from the baseline as a new event. The complexity of the proposed models ranges from simple moving averages to approaches using HMM and Recurrent Neural Networks (RNN).

In [86], Ould-Ahmed-Vall et al. develop an approach for detecting anomalous behavior in a region covered by distributed wireless sensor networks. They integrate localization information into collections of sensor time series to enhance the reliability of the event detection mechanism. The fusion of the temporal and location data occurs at the decision stage. This is achieved by computing the event detection for each time series individually and taking into account the relative location of the sensors in a weighted aggregation score.

The detection procedure works as follows. Each sensor produces a time series of measurements, upon which statistical anomaly detection is applied to determine whether an event occurred in the first step. Next, the local time series are fused with the localization data using a geographically weighted voting scheme that considers the event detection response of the neighboring nodes, weighted by the relative location of a sensor in the network. Thus, a geographical threshold distribution is derived around each node, which is finally used to detect the occurrence of local events in the sensor networks. The proposed approach is assessed on a simulated synthetic network of sensors to detect sensor faults.

Banerjee et al. [9] tackle the problem of detecting events in an urban setting. They combine video (surveillance footage) with streams of social media data messages (from Twitter and Instagram) to detect anomalous events. Data fusion occurs at the feature level by counting entities in each modality to obtain a fused time series. Specifically, the authors apply a pre-trained Convolutional Neural Network (CNN) to identify and count objects, such as vehicles and people, appearing in the video streams sampled every second. Social media posts are divided into regions according to their publication location and counted at one-second intervals. The internal representation is obtained by summing up the total counts of vehicles, persons, and posts every second. Upon fusing time series, the authors propose an approach relying on a hidden Markov model (partially observable Markov decision processes) to trigger the detection of anomalous count sequences.

In [104], Reuter et al. propose a technique to detect new events in streams of images annotated with textual descriptions, titles, timestamps, and tags. The event detection component maintains an updated list of known events and compares incoming instances with these events. During image stream monitoring, an incoming image is sequentially assigned either to a known event or triggers the detection of a new one. Data fusion is achieved at the feature level by comparing a new image to ten representative images from each of the known event classes. For each class, the authors compute similarity scores within each modality: the time differences (image creation timestamp, upload timestamp), the geographic distances, and textual differences (cosine similarity of tf-idf vectors of descriptions and titles). The similarity scores are scaled and fused by concatenating them into a representation vector.

Event detection decides whether the new image is assigned to one of the known event classes or if it constitutes a new event. Given the fused similarity vectors above, an SVM is trained for each known event to compute the probability that the new image pertains to that event. Combining these prediction scores of each intermediate SVM into a new feature vector allows the final event detection SVM to predict whether the image should be assigned to one of the known events or constitutes a new event.

Chen et al. [28] propose a semi-supervised technique to detect events in road traffic networks. The proposed approach fuses sequential sensor data streams and social media content to identify anomalous events. To achieve this task, the authors use a neural network architecture based on Long Short-Term Memory (LSTM) [11, 52], combined with CNNs and a Generative Adversarial Network (GAN) [39, 67, 92].

In the first stage, embeddings of physical sensor time series (measuring traffic flows) are learned incrementally using two subsequent LSTM networks to represent the sensor values. The textual content of social media messages (published in the relevant regions) is encoded using pre-initialized word embeddings. These embeddings are fed into a CNN to learn an intermediate text representation. In parallel, a GAN is trained in a semi-supervised manner to produce synthetic sensor readings and text messages, both of which are embedded with the computed encoders.

The data fusion mechanism occurs at the representation level in the second stage. The authors combine the above time series embeddings and text embeddings by concatenating them to obtain a fused latent representation. Finally, a fully connected neural network is trained on the fused representation for the event detection classification task. To evaluate this approach, the authors combine traffic sensor data from the Caltrans Performance Measurement System and simultaneous geotagged social media messages originating from the same region.

In the clinical setting, Khadanga et al. [63] propose a deep learning-based technique to detect medical decompensation events for patients in an intensive care unit. They continuously monitor a patient's physiological state and combine it with sequences of textual data provided in clinical notes. The physiological data comprises various health measurements sampled at regular intervals. The authors propose using an LSTM network to learn the time series' embedding. The textual medical notes are written intermittently at irregular intervals. A CNN feature extractor is first applied to each note to embed all previously written notes. To prioritize recent medical notes, the learned text encodings are combined with a weighted sum. The latter exponentially decays with the time duration since the notes' creation to obtain a single embedding.

The data fusion mechanism intervenes at the representation level by concatenating the time series and clinical notes' embeddings. Event detection relies on a fully connected neural network layer. It takes the fused latent representation as input and produces a binary classification, assessing whether the patient's health is rapidly declining every hour.

4.2.2 Graph processing techniques

Multimodal event detection applies graph processing techniques to encode the multimodal datasets into a graph representation. It uses either statistics to detect anomalous subgraphs or graph neural networks to label nodes into event classes.

Chen et al. [27] propose a technique called Non-Parametric Heterogeneous Graph Scan (NPHGS) for early event detection in streams of social media messages. Their approach extracts a heterogeneous graph from social media data and detects events by fusing textual content, time information, geolocation, and social relationships. Specifically, the authors represent entities, such as users, messages, links, hashtags, and geolocation, as nodes of different types in a heterogeneous graph. Each node is additionally mapped to a feature vector at every timestamp (e.g., number of followers, number of retweets, etc.). Relationships between entities can also be of different types (e.g., retweets, followers, actions).

To detect events, the heterogeneous graph is mapped into a sensor network, where each node tracks the changes in its features and the ones of its neighbors. When no event occurs, statistical baselines are stored for each node at multiple levels of temporal granularity. An empirical *p*-value test is then computed to quantify how anomalous the feature values of each node are. A non-parametric scan statistic then uses the value for each node to identify anomalous subgraphs, which are, in turn, considered detected events.

Using graph-based statistical anomaly detection, Pan et al. [89] detect traffic events by combining location and mobility data, as well as textual social media posts. A preliminary mining step takes place offline and allows the establishment of historical baselines. This step derives usual routing trajectories, velocity, and temporal road segment utilization from GPS location data. A baseline of the usual terms used in social media posts is also determined. The observed road network is then modeled as a graph, and deviations from the expected traffic flow are used to identify anomalous subgraphs during runtime. Finally, social media messages published from these regions are used to detect the type of traffic event that occurred (e.g., accidents, protests, disasters, etc.).

In [95], Peng et al. extend their work on retrospective event detection (see [94] in Sect. 4.1.2) to handle streaming applications. Similarly to their original work, the authors model the semantic relatedness of social events with a heterogeneous graph. In this extension, they apply graph ML to group incoming instances into event clusters. An internal representation in the form of a heterogeneous network is built, where users, time, keywords, topics, and concepts are extracted from text messages. The extracted entities are enriched with information from an external knowledge base and are considered nodes of different types. The data fusion occurs at the representation level. The authors propose a novel Graph Convolutional Network (GCN) architecture called Pairwise Popularity GCN (PP-GCN) to learn pairwise similarities between nodes of different modalities in the heterogeneous graph.

The event detection operates on the similarity scores between nodes learned by the PP-GCN algorithm. The incoming stream of messages is partitioned into fixed time-frames. Events are detected during each time window using a modified version of DBSCAN for heterogeneous data, and the instances are clustered according to their similarity scores. In addition to clustering the incoming messages, a similarity threshold between messages is learned to detect whether an instance should be assigned to an existing or to a new event cluster.

4.3 Predictive event detection

This section discusses techniques aiming at detecting impending events. Typically, those techniques regularly evaluate whether recent observations can indicate future events and are thus generally formulated as classification tasks. The techniques for future event detection can be classified into two broad classes: (i) classical statistical and machine learning approaches and (ii) more nascent deep neural network-based approaches.

4.3.1 Classical machine learning-based techniques

Classical statistical and ML approaches often rely on statistical correlations to produce binary predictions about whether events are imminent.

In the clinical domain, Chen et al. [26] study the problem of forecasting acute health events for patients undergoing surgery. The proposed solution predicts the risk of postoperative bleeding based on physiological parameters, patient-provided information, and textual clinical notes written between the time of admission and the surgery. The authors fuse time-varying information (e.g., time series of physiological measurements) with categorical patient data (e.g., demographic characteristics, test results) and text (e.g., medical notes) at the feature level. The time series are converted into vectors of summary statistics (i.e., minimum, mean, and maximum values), and the categorical values are concatenated into a vector. To do so, the authors apply an off-the-shelf natural language processing tool (MedTagger) [75] to extract relevant medical information from the clinical notes. The three resulting vectors are concatenated into a fused representation vector, considering all modalities. Event prediction uses these vector representations to train a Gradient Boosting Machine (GBM) model [59], which is used to predict the risk of future health events.

Similarly, Horng et al. [53] propose a system to monitor patient health status and automatically predict imminent acute health events. Specifically, patient demographic characteristics such as age and gender are taken into account and combined with vital sign measurements, such as heart rate and blood pressure, as well as clinical notes describing the patient's state and issues. The former are discretized when necessary and converted to categorical variables. The authors apply bigram detection to the latter and represent notes either as term frequencies or as topics, which are further derived using topic models. The authors use an SVM on the combined vector representation for event prediction to detect imminent sepsis events and alert healthcare professionals of further triage and intervention.

Table 3 Datasets description

Dataset	Source	Modalities	Size	Domain
`MIMIC-III` [63, 128]	[61]	Numerical, text	50'000	Medical
`TRECVID 2011` [133]	[88]	Video, text	40'000	Multimedia
`TRECVID 2013` [38, 84, 125]	[87]	Video, text	98'000	Multimedia
`MediaEval 2011` [22, 97, 122]	[91]	Images, text, time	73'645	Social networks
`MediaEval 2012` [78, 99, 131]	[90]	Images, text, time	166'332	Social networks
`MediaEval 2014` [130]	[98]	Images, text, time	362'578	Social networks
`Telecom Big Data challenge 2014` [24]	[56]	Location, time, text	–	Mobility & communication

4.3.2 Deep learning-based techniques

Deep learning-based approaches using CNNs and LSTMs embed the input modalities before training event detectors on these learned embeddings. Because of their expensive training time, these models are often partially pre-trained, especially for text analysis, where model parameters are fitted to large datasets beforehand, and pre-trained word embeddings are used for inference.

In [106], Rodrigues et al. propose two deep learning architectures leveraging word embeddings, convolutional layers, and attention mechanisms to combine text information with time-series data and predict mobility demand in eventful urban areas. Their main hypothesis is that text often contains contextual cues for many of the events and patterns that can be observed in temporal data and, as such, is instrumental in predicting time series. The authors represent text data using well-known GloVe [96] embeddings fed into convolution filters, max-pooling layers, and finally, an attention layer. The time series data are fed either to an LSTM or a stack of fully connected layers and combined with a binary vector indicating whether or not the corresponding value (i.e., mobility demand) at a given timestamp in the series occurred during the appearance of an event.

The two latent representations, text and time series, are then combined using a dense layer to compute a prediction for the next time interval. Using time series data corresponding to taxi pickups in a given area and event information extracted from the Web, the authors show that their cross-modal data fusion technique significantly reduces forecasting error. Furthermore, their results show that using event information extracted from the web helps improve the quality of the predictions dramatically, significantly outperforming popular time-series forecasting methods.

In [128], Yang et al. predict mortality events. To exploit the multimodal nature of electronic medical records, the authors propose a deep learning-based pipeline that combines patients' health measurements with clinical notes. The prediction of mortality events consists of a binary prediction of whether a patient is expected to die during the next two days of their stay in the intensive care unit. The proposed approach starts by learning representations of the two input modalities distinctly. Physiological measurements are provided as time series with 17 features and are embedded using LSTM networks. The clinical notes describe patient symptoms, clinical histories, and medical reports.

The content is encoded using the popular word embedding technique Word2vec [80], trained on a medical citation dataset. Next, these encodings are passed through a convolutional neural network layer to produce text embeddings. The data fusion of the two modalities occurs at the level of their representation, i.e., the embeddings produced by the above two neural network models are concatenated to produce a fused latent representation for each patient. Event detection is achieved with a final fully connected layer, which takes as input the aforementioned fused representations.

5 Reported results

In this section, we introduce the most common datasets and metrics used to evaluate multimodal event detection techniques. Then, we discuss their reported results and the applicability of the proposed techniques.

5.1 Datasets

Our analysis identified three popular datasets in multimodal event detection. Some variations of those datasets exist, with yearly competitions extending the number of data instances in every iteration. Table 3 summarizes the main properties of existing datasets.

The MIMIC-III (Medical Information Mart for Intensive Care) dataset [61] consists of 50'000 patient records collected over 11 years in an intensive care unit. It provides demographic information, as well as measurements of vital signs, medical interventions, etc., for each patient. Specifically, the dataset comprises time-series data (e.g., bedside monitoring), free text (e.g., clinical notes), patient outcome metrics (e.g., length-of-stay), and categorical features (e.g., clinical diagnostics codes). This dataset is widely used in evaluating techniques in the medical domain, particularly for our survey, by the techniques proposed in [63, 128]. These techniques focus on fusing time series of patients' physiological measurements with textual clinical notes and patient demographic information.

The TRECVID datasets are provided yearly as a challenge with multiple subgoals, one of which is a multimedia event detection task. The datasets comprise annotated videos from various sources (namely, BBC, Internet Archive, surveillance video, and Flickr). Importantly, the dataset includes a list of pre-defined events with textual descriptions and illustrative video sequences of the event. Event detection techniques focusing on fusing text with audio and visual channels such as [38, 84, 125, 133] address the multimedia event detection task, wherein events described by a textual description are to be detected in the dataset.

The MediaEval datasets are a benchmarking reference for detecting social events in annotated image collections. The 2011 MediaEval dataset [91] comprises 73'645 images obtained through the Flickr API, originating from Amsterdam, Barcelona, London, Paris, and Rome. The 2012 dataset [90] contains 166'332 images published from Barcelona, Madrid, Hamburg, Cologne, and Hannover. In the 2014 version [98], the dataset was composed of two collections, one of 362'578 images, grouped into 17'834 events, and the second of 110'541 images, provided without event labels. All the images are annotated with metadata, namely the uploader's username, publication timestamps, titles, textual descriptions, tags, and geolocation data. The MediaEval Social Event Detection challenge addressed by [22, 78, 97, 99, 122, 130, 131] consists of detecting and describing events in these datasets.

Telecom Italia provides a dataset for event detection in telecommunications. For a data analytics competition (*Telecom Big Data Challenge 2014*), millions of data points about SMS, calls, internet connections, energy consumption, tweets, weather, and mobility were recorded. The dataset covers two Italian cities (Milan and Trento) in November and December 2013. It was used in [14, 24] to detect city life events.

5.2 Metrics

The chosen evaluation metric depends on the event detection task. In the case of anomaly detection tasks, authors assess the ability of a given technique to recognize eventful instances and how well it discards non-eventful instances. Event detection tasks aggregating instances into event clusters need to assess how well-separated the clusters are.

To evaluate anomaly identification and classification tasks, techniques are often compared according to precision–the proportion of correctly detected events among all instances positively flagged–as such by [24, 28, 48, 104]. Recall–the proportion of correctly detected events out of the total number of ground truth events–is evaluated by [24, 28, 48, 104]. Combining precision and recall in a summary statistic with the harmonic mean, the F1-score is reported in [94, 95, 104, 122]. Less frequently, the mean average precision, the probability of missed event detection, and the probability of false tweet alarm errors are computed to evaluate the event detection probability technique proposed by [23].

If class probabilities (or event detection probabilities) are produced instead of class prediction, the subsequent question will be how to select a threshold to discriminate between a detected and a rejected event. The receiver operating characteristic curve (ROC) formalizes this selection. Given a predicted probability for each candidate instance in the test set, the ROC varies the threshold to obtain a binary classification. For each threshold, it computes the classifier's specificity and recall and plots the true positive rate against the false positive rate. Such a curve represents the number of true positive examples that are missed as non-events and the number of non-events that are misdetected as events for the threshold values of the class probabilities.

Computing the area under this curve yields the AUC-ROC, a performance indicator between 0 and 1. This measure is reported by [28, 38, 53, 63, 106]. The PR curve plots the precision against the recall scores obtained in the same manner as above by varying the binary decision threshold. The area under this curve (AUC-PR) yields another metric, as reported in [26, 63, 106, 128].

In the clustering task, cluster quality is evaluated either externally or internally [103]. In the former case, ground truth labels are used as external information. When no additional information is available, internal cluster validation methods compare the similarities inside and across clusters.

For external cluster validation, most event detection techniques are evaluated with the Normalized Mutual Information (NMI), as in [78, 95, 97, 99, 130, 131]. The NMI score evaluates clustering by computing the amount of information shared when instances are split according to their label and when instances are split according to their predicted cluster. The mutual information score is obtained with Eq. 1, where X stands for the clusters and Y for the class labels:

$$MI(X;Y) = \sum_{y \in Y} \sum_{x \in X} P_{X,Y}(x,y) \log \left(\frac{P_{X,Y}(x,y)}{P_X(x) P_Y(y)} \right) \quad (1)$$

MI can be further normalized proportionally to the marginal entropies of the classes and clusters to obtain a score between 0 and 1. Alternatively, the MI can be adjusted to take random clusterings as the baseline with the Adjusted Mutual Information [120].

In internal cluster validation, the quality is measured regarding compactness and separation. The former indicates how similar instances within a cluster are, while the latter defines how dissimilar instances within a cluster are from the ones of another cluster [138]. For instance, the authors in [45] use the Dunn index [34], Davies-Bouldin index [33], and Silhouette index [108]. They aim to measure how tightly knit and how well-separated the various predicted clusters are.

The Dunn index identifies the sets of clusters with a small variance between members of the cluster and where the means of different clusters are sufficiently far apart. It captures compactness as the maximal distance between instances of the same cluster and separation as the minimal distance between clusters. The index is then computed by dividing the separation by the compactness, i.e.,

$$DI = \frac{\min\limits_{1 \leq i < j \leq m} \delta(C_i, C_j)}{\max\limits_{1 \leq k \leq m} \Delta_k} \tag{2}$$

Given m clusters, where Δ_i corresponds to the maximal distance between instances of the same cluster and $\delta(C_i, C_j)$ the distance between the closest instances in clusters C_i and C_j. The core limitations of the Dunn index, as highlighted by Halkidi et al. [47], are its high computational complexity and its sensitivity to noise in the dataset, which might significantly affect the distance between instances within the same cluster Δ_i.

The Davies-Bouldin index measures the ratio between the within-cluster distances and the between-cluster distances. It computes the compactness by each instance's distance to its cluster's centroid and separation by the maximal similarity between instances of different clusters. The score is computed as follows:

$$DB = \frac{1}{c} \sum_{i=1}^{c} Max_{i \neq j} \left\{ \frac{d(X_i) + d(X_j)}{d(c_i, c_j)} \right\} \tag{3}$$

with $d(X_i)$ being the distance of all instances in cluster i to its centroid, $d(c_i, c_j)$ the distance between centroids of clusters i and i, and c the number of clusters.

The Davies-Bouldin index assumes clusters with similar size and density [107]. In case the clusters contain outliers and noise, the results of this index do not reflect the clustering quality of the detection.

The Silhouette index captures compactness and separation by computing how confident the cluster assignment of each instance is. It compares how similar each instance (x) is, on average, to the other instances in its cluster $(a(x))$ with how similar, on average, it is to the instances in other clusters $(b(x))$, averaged over all instances, i.e.,

$$s(x) = \frac{b(x) - a(x)}{\max\,(a(x), b(x))} \tag{4}$$

The Silhouette index is appropriate when the event clusters are far apart [100]. If the clusters are close to each other or follow a non-conventional form, this index becomes non-indicative of the performance of the event detection algorithm.

Only a handful of authors evaluate the efficiency of their proposed algorithm. Elhoseiny et al. [38] do so by measuring the runtime of their approach and comparing it against their baselines. Other indicators of event detection during runtime are the average lead and lag times, as reported by Chen et al. [27].

5.3 Analysis of Results

Various authors have compared the performance of their techniques against other baselines. Table 4 describes the results originally reported by the authors of the papers we include in this survey. We omit papers with no specified numerical results (e.g., only showing precision-recall curves).

The reported results point to a couple of trends. Whenever ground truth labels are available, authors can report the precision, recall, F1, accuracy (Acc.), ROC-AUC, ROC-PR, mean average precision (MAP), and normalized mutual information (NMI). Otherwise, authors rely on clustering quality metrics such as Dunn, Davies-Bouldin (D-B), and Silhouette (Sil.) indices.

We also observe that the performance of the techniques in terms of Recall and Precision is always higher than 0.7. Furthermore, all techniques reporting both metrics achieve a higher Precision than Recall. This indicates that it is more difficult to find the relevant events in the dataset (lower recall) and easier to reject false positives (higher precision).

When comparing cluster quality, the most popular metric is NMI. Results range from 0.54 to almost perfect, with Yang et al. [130] reaching 0.99 (out of 1). The authors achieve this score by infusing their clustering mechanism with labeled data at the initialization (i.e., by using a semi-supervised approach). The high AUC-ROC results reported by Chen et al. [28] in the detection of events in roadway traffic (fusing road sensor and social media data) is achieved using a semi-supervised architecture, stressing the high potential of such techniques for event detection.

Some datasets are inherently more difficult than others. With an increasing number of modalities, techniques need to be increasingly sophisticated to extract relevant information from each source. For example, results (regarding AUC-ROC) on the MIMIC-III dataset, consisting of time series, text, and categorical variables, are generally higher than those on the TRECVID dataset. Furthermore, we notice that techniques evaluated on social networks (e.g., using data from Twitter, Sina Weibo, or Tencent) generally achieve high results, indicating that the modalities in social media streams (mainly text, temporal information, and social interactions)

Table 4 Reported results

Dataset	Techniques	Metrics											Category		
		Recall	Precision	F1	Acc	AUC-ROC	AUC-PR	MAP	NMI	D-B	Dunn	Sil	RP	RT	PR
MIMIC-III	Khadanga et al. [63]	–	–	–	–	0.91	0.35	–	–	–	–	–	–	✓	–
	Yang et al. [128]	–	–	–	–	0.86	0.56	–	–	–	–	–	–	–	✓
TRECVID 2013	Wu et al. [125]	–	–	–	–	0.73	–	0.13	–	–	–	–	✓	–	–
	Elhoseiny et al. [38]	–	–	–	–	0.83	–	0.13	–	–	–	–	✓	–	–
Upcoming	Wang et al. [122]	–	–	0.79	–	–	–	–	–	–	–	–	✓	–	–
Twitter data	Han et al. [48]	0.75	0.80	–	–	–	–	–	–	–	–	–	✓	–	–
Flickr & last.fm data	Reuter et al. [104]	0.70	0.80	0.74	–	–	–	–	–	–	–	–	–	✓	–
Traffic & social media data	Chen et al. [28]	0.87	0.87	0.87	0.87	0.93	–	–	–	–	–	–	–	✓	–
Mayo Clinic patients data	Chen et al.[26]	–	–	–	–	0.80	0.42	–	–	–	–	–	–	–	✓
Sina Weibo data	Peng et al. [94]	–	–	0.81	0.80	–	–	–	0.78	–	–	–	✓	–	–
	Peng et al. [95]	–	–	0.93	0.92	–	–	–	0.93	–	–	–	–	✓	–
Tencent data	Peng et al. [94]	–	–	0.92	0.93	–	–	–	0.90	–	–	–	✓	–	–
MediaEval SED 2011	Brenner et al. [22]	0.71	0.72	–	–	–	–	–	0.54	–	–	–	✓	–	–
	Wang et al. [122]	–	–	0.64	–	–	–	–	–	–	–	–	✓	–	–
	Petkos et al. [97]	–	–	–	–	–	–	–	0.69	–	–	–	✓	–	–
MediaEval SED 2012	Yang et al. [131]	–	–	–	–	–	–	–	0.76	–	–	–	✓	–	–
	Petkos et al. [99]	–	–	–	–	–	–	–	0.93	–	–	–	✓	–	–
	Ma et al. [78]	–	–	–	–	–	–	–	0.77	–	–	–	✓	–	–
MediaEval SED 2014	Yang et al. [130]	–	–	–	–	–	–	–	0.99	–	–	–	✓	–	–
Twitter data	Ghaemi et al. [45]	–	–	–	–	–	–	–	–	212.89	0.72	0.64	✓	–	–

Underlined names refer to new datasets introduced by the authors
RP, RetrosPective; RT, real-time; PR, PRospective; respectively

are prone to easier fusion. Another striking pattern is the improvement in performance for the yearly MediaEval challenges, indicative that the increasing size of the dataset allows for the development of more refined techniques.

6 Application areas

As stated earlier, techniques for multimodal event detection have applications in several domains. We describe those applications and highlight common use cases for detecting social, medical, and multimedia events.

6.1 Social networks

Detecting events on social media is one of the more widespread event detection applications. On micro-blogging platforms (e.g., Twitter) events such as breaking news or election results are often discussed in real-time as they unfold. Additionally, collecting social media content generates inherently multimodal datasets, combining timestamped textual messages with social interaction data, popularity metrics, and multimedia attachments (e.g., images, video clips, etc.).

We distinguish two types of applications for multimodal event detection using social media data. The first type of techniques assigns messages into groups (event clusters or event classes), either retrospectively on past data (e.g., [23, 45, 94, 132]) or in real-time streams of messages (e.g., [95, 104]). The second set of approaches identifies anomalous instances pertaining to events retrospectively with [48] and during runtime with [27].

Social media data can also be combined with external sources to detect real-world events. Combined with physical sensor and mobility data, the authors in [28, 89, 106] use multimodal event detection to detect traffic events. To identify anomalous behavior in urban spaces, the authors in [9, 24] merge social media streams with surveillance video and telecommunication data.

6.2 Medical applications

In the medical domain, patients are treated in an intensive care unit and attached to multiple physiological sensors. Their medical record is composed of prior health assessments, patient demographic data, as well as past and current textual clinical notes. Considering historical health data in combination with up-to-date physiological information can allow for the prediction of imminent, acute health events and alert medical staff.

Data fusion is becoming increasingly prevalent thanks to the increasing availability of physiological sensor data collected during medical procedures. This prevalence is also explained by the high-stakes need for early intervention and prophylactic treatment. Data fusion becomes a necessary procedure for reliable event detection with the multimodal encoding of sensor readings, medical imagery, case history, diagnostics, medical history, and demographic information. For instance, the technique proposed in [63] is applied to monitor patient status in real-time and detect acute health deterioration. The authors consider both medical sensor readings and the clinical notes produced intermittently by the medical staff.

Another example of potential clinical applications is presented in [26, 53, 128], where the authors predict imminent health events by infusing demographic and health data in textual and numerical form.

6.3 Multimedia events

Multimodal event detection is also applied to mining events from multimedia archives, such as video and image collections. To handle the task of detecting events in videos, the proposed techniques in [38, 84, 125, 133] first split the audio track and image streams to extract meaningful representations such as low-level visual and audio patterns, or higher-level optical character and speech recognition. Event detection models are then trained on these representations. Alternatively, the authors in [22, 78, 97, 99, 110, 122, 130, 131] apply multimodal event detection to detect images fused with their metadata.

7 Research opportunities

In the following, we describe a number of directions that could improve the research related to multimodal detection. The directions we describe below aim to foster the reproducibility of results in this field and to adapt multimodal detection to nascent fields.

Benchmark for multimodal event detection. Our survey highlights the difficulty in establishing meaningful performance rankings of multimodal event detection techniques. Authors frequently collect their own datasets and evaluate their techniques on them without making them publicly available. Furthermore, even when public datasets are used, authors perform experiments on different subsets of the data, augment the datasets with external information, and report different performance metrics on a reduced set of modalities.

A potential remedy for the results' inconsistencies would be to introduce a new test bed for comparing a curated list of event detection techniques. The benchmark should ideally implement a variety of metrics and include a data generator able to create new synthetic datasets with real-like modalities. Similar benchmarks have been introduced in other fields such as anomaly detection [111], imputation of missing values [66], or similarity search [35].

Fusion at the hardware level. As described in Sect. 4.2, many data fusion techniques target scenarios where events must be detected in real-time. For such scenarios, porting the fusion algorithm to various hardware accelerators might drastically reduce event detection latency. GPUs are prime candidates in this context, as they are known to significantly accelerate many machine learning tasks (both for training and inference [114]). In [4], for instance, researchers present a work-in-progress paper describing how GPUs were leveraged to accelerate the detection of roadside incidents by fusing three different data sources. Many related works could benefit from such acceleration using GPUs to run their fusion models.

Beyond GPUs, further accelerators could be used to speed up data fusion and event detection techniques. Next-generation networking and storage components–perhaps surprisingly–can today be used to accelerate tasks in large-scale systems since some of them feature considerable computational capabilities [73]. We identify smart SSDs, smart NICs, and smart routers as particularly promising in our context, as they have all recently been shown to substantially accelerate a variety of data-intensive tasks [55, 58, 72].

Fusion using IoT edge devices. Another exciting development is the wide use of edge devices to power IoT applications. The deployment of heterogeneous devices at the edge of the network is significant in our context for two main reasons: first, the fact that sets of devices running at the edge take over entire tasks that were traditionally solved centrally (either on a central server or in the cloud) prompts for a new kind of distributed computation called *edge computing* [36]. In edge computing, processing capabilities are pushed to relatively light devices such as sensors or mobile devices. While response time might be improved using nearby devices, this also changes how algorithms (e.g., event detection) are run, as none of the edge devices has full knowledge or control of the situation at hand. This opens lots of opportunities in terms of distributing or decentralizing event detection techniques, leveraging a set of less-capable and self-organizing nodes in a dynamic fashion.

Second, the wide heterogeneity of the devices used at the edge makes data fusion a requirement in many scenarios. In that context, many research opportunities are emerging, with new devices (e.g., wearables [93]) appearing along with new modalities requiring a whole new family of data fusion techniques. One particularly compelling application in this context is *urban edge analytics* [29], where mobile devices (e.g., autonomous cars) collaboratively detect events with the help of base stations (e.g., smart traffic lights and roadside units).

8 Conclusion

In this paper, we presented a survey of multimodal event detection techniques and introduced a new taxonomy of the field. We distinguish techniques according to their temporal axis and classify them into three broad categories: retrospective, real-time, and forecasting techniques. Our survey covers a wide range of detection algorithms, from simple statistical methods to more nascent deep neural network techniques. Additionally, we discussed the data fusion adopted by each multimodal event detection technique. We identified three main data fusion mechanisms based on data characterizations, transformed representations, or occurring before decision-making. We explained the main differences between those mechanisms by highlighting their peculiarities along the event detection pipeline.

Finally, we explored the experimental setup used by the various techniques we surveyed. We discussed standard evaluation metrics and reviewed recurring benchmarking datasets. We also reported the performance of the proposed techniques and uncovered inconsistencies in the benchmarking setup. This analysis helped us identify limitations of previous work's applicability and propose potential future research endeavors in the multimodal event detection field.

Acknowledgements This work has received funding from ArmaSuisse S+T and from the Swiss State Secretariat for Education (SERI) in the context of the SmartEdge EU project (Grant agreement No. 101092908).

Funding Open access funding provided by University of Fribourg.

References

1. Al Hasan Haldar, N., Li, J., Reynolds, M., Sellis, T., Yu, J.X.: Location prediction in large-scale social networks: an in-depth benchmarking study. VLDB J. **28**, 623–648 (2019)
2. Ali, S.M.F., Wrembel, R.: From conceptual design to performance optimization of ETL workflows: current state of research and open problems. VLDB J. **26**(6), 777–801 (2017)
3. Allan, J., Carbonell, J.G., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study final report (1998)
4. Aqib, M., Mehmood, R., Alzahrani, A., Katib, I., Albeshri, A., Altowaijri, S.M.: Smarter traffic prediction using big data, in-

memory computing, deep learning and GPUs. Sensors **19**(9), 2206 (2019)

5. Arous, I., Yang, J., Khayati, M., Cudré-Mauroux, P.: Opencrowd: a human-AI collaborative approach for finding social influencers via open-ended answers aggregation. In: Y. Huang, I. King, T. Liu, M. van Steen (eds.) In: WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, pp. 1851-1862. ACM / IW3C2 (2020). 10.1145/3366423.3380254
6. Atefeh, F., Khreich, W.: A survey of techniques for event detection in twitter. Comput. Intell. **31**(1), 132–164 (2015)
7. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings, pp. 722–735. Springer (2007)
8. Babaee, E., Anuar, N.B., Abdul Wahab, A.W., Shamshirband, S., Chronopoulos, A.T.: An overview of audio event detection methods from feature extraction to classification. Appl. Artif. Intell. **31**(9–10), 661–714 (2017)
9. Banerjee, T., Whipps, G., Gurram, P., Tarokh, V.: Sequential event detection using multimodal data in nonstationary environments. In: 2018 21st International conference on information fusion (FUSION), pp. 1940–1947. IEEE (2018)
10. Bao, B.K., Min, W., Lu, K., Xu, C.: Social event detection with robust high-order co-clustering. In: Proceedings of the 3rd ACM conference on International conference on multimedia retrieval, pp. 135-142 (2013)
11. Baytas, I.M., Xiao, C., Zhang, X., Wang, F., Jain, A.K., Zhou, J.: Patient subtyping via time-aware LSTM networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 65–74 (2017)
12. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: real-world event identification on twitter. In: Proceedings of the international AAAI conference on web and social media, vol. 5, pp. 438–441 (2011)
13. Begum, N., Keogh, E.: Rare time series motif discovery from unbounded streams. Proc. VLDB Endow. **8**(2), 149–160 (2014)
14. Bendre, M.R., Thool, V.R.: Analytics, challenges and applications in big data environment: a survey. J. Manag. Anal. **3**(3), 206–239 (2016). https://doi.org/10.1080/23270012.2016.1186578
15. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
16. Benson, E., Haghighi, A., Barzilay, R.: Event discovery in social media feeds. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pp. 389-398 (2011)
17. Bhardwaj, A., Yang, J., Cudré-Mauroux, P.: A human-AI loop approach for joint keyword discovery and expectation estimation in micropost event detection. In: AAAI Conference on Artificial Intelligence (AAAI'20). New York, USA (2020)
18. Bishnu, P.S., Bhattacherjee, V.: Software fault prediction using quad tree-based k-means clustering algorithm. IEEE Trans. Knowl. Data Eng. **24**(6), 1146–1150 (2011)
19. Blauensteiner, P., Kampel, M., Musik, C., Vogtenhuber, S.: A socio-technical approach for event detection in security critical infrastructure. In: 2010 IEEE computer society conference on computer vision and pattern recognition-workshops, pp. 23-30. IEEE (2010)
20. Blei, D.M., Griffiths, T.L., Jordan, M.I.: The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. J. ACM (JACM) **57**(2), 1–30 (2010)
21. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
22. Brenner, M., Izquierdo, E.: Social event detection and retrieval in collaborative photo collections. In: proceedings of the 2nd ACM International Conference on Multimedia Retrieval, pp. 1-8 (2012)
23. Cai, H., Yang, Y., Li, X., Huang, Z.: What are popular: exploring twitter features for event detection, tracking and visualization. In: Proceedings of the 23rd ACM international conference on Multimedia, pp. 89-98 (2015)
24. Cecaj, A., Mamei, M.: Data fusion for city life event detection. J. Amb. Intell. Human. Comput. **8**(1), 117–131 (2017)
25. Chan, T.K., Chin, C.S.: A comprehensive review of polyphonic sound event detection. IEEE Access **8**, 339–373 (2020)
26. Chen, D., Afzal, N., Sohn, S., Habermann, E.B., Naessens, J.M., Larson, D.W., Liu, H.: Postoperative bleeding risk prediction for patients undergoing colorectal surgery. Surgery **164**(6), 1209–1216 (2018)
27. Chen, F., Neill, D.B.: Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1166-1175 (2014)
28. Chen, Q., Wang, W., Huang, K., De, S., Coenen, F.: Multi-modal generative adversarial networks for traffic event detection in smart cities. Expert Syst. Appl. **177**, 114939 (2021)
29. Chowdhery, A., Levorato, M., Burago, I., Baidya, S.: Urban IoT edge analytics, pp. 101–120. Springer International Publishing, Cham (2018)
30. Cong, Z., Chu, L., Yang, Y., Pei, J.: Comprehensible counterfactual explanation on Kolmogorov-Smirnov test. Proc. VLDB Endow. **14**(9), 1583–1596 (2021). https://doi.org/10.14778/3461535.3461546
31. Cordeiro, M., Gama, J.: Online social networks event detection: a survey. In: Michaelis, S., Piatkowski, N., Stolpe, M. (eds.) Solving large scale learning tasks. Lecture notes in computer science. Springer, Cham (2016)
32. Daras, P., Manolopoulou, S., Axenopoulos, A.: Search and retrieval of rich media objects supporting multiple multimodal queries. IEEE Trans. Multimed. **14**(3), 734–746 (2011)
33. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **2**, 224–227 (1979)
34. Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. J. Cybern. **4**(1), 95–104 (1974)
35. Echihabi, K., Zoumpatianos, K., Palpanas, T., Benbrahim, H.: The Lernaean hydra of data series similarity search: an experimental evaluation of the state of the art. Proc. VLDB Endow. **12**(2), 112–127 (2018). https://doi.org/10.14778/3282495.3282498
36. El-Sayed, H., Sankar, S., Prasad, M., Puthal, D., Gupta, A., Mohanty, M., Lin, C.: Edge of things: the big picture on the integration of edge, IoT and the cloud in a distributed computing environment. IEEE Access **6**, 1706–1717 (2018)
37. Elgamal, T., Yabandeh, M., Aboulnaga, A., Mustafa, W., Hefeeda, M.: SPCA: scalable principal component analysis for big data on distributed platforms. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 79-91 (2015)
38. Elhoseiny, M., Liu, J., Cheng, H., Sawhney, H., Elgammal, A.: Zero-shot event detection by multimodal distributional semantic embedding of videos. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
39. Fan, J., Liu, T., Li, G., Chen, J., Shen, Y., Du, X.: Relational data synthesis using generative adversarial networks: a design space exploration. Proc. VLDB Endow. **13**(11), 1962–1975 (2020)
40. Fellbaum, C.: WordNet: an electronic lexical database. MIT press, Cambridge (1998)
41. Fernandez, R.C., Min, J., Nava, D., Madden, S.: Lazo: A cardinality-based method for coupled estimation of jaccard simi-

larity and containment. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 1190-1201. IEEE (2019)

42. Gan, J., Tao, Y.: DBSCAN revisited: MIS-claim, un-fixability, and approximation. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data, pp. 519-530 (2015)
43. Gao, J., Li, P., Chen, Z., Zhang, J.: A survey on deep learning for multimodal data fusion. Neural Comput. **32**(5), 829–864 (2020)
44. Garg, M., Kumar, M.: Review on event detection techniques in social multimedia. Online Inform. Rev. **40**(3), 347–361 (2016)
45. Ghaemi, Z., Farnaghi, M.: A varied density-based clustering approach for event detection from heterogeneous twitter data. ISPRS Int. J. Geo-Inform. **8**(2), 82 (2019)
46. Goswami, A., Kumar, A.: A survey of event detection techniques in online social networks. Soc. Netw. Anal. Mining **6**, 1–25 (2016)
47. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering validity checking methods: Part II. ACM Sigmod Record **31**(3), 19–27 (2002)
48. Han, Y., Karunasekera, S., Leckie, C., Harwood, A.: Multi-spatial scale event detection from geo-tagged tweet streams via power-law verification. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 1131-1136. IEEE (2019)
49. Hanskamp-Sebregts, M., Zegers, M., Vincent, C., van Gurp, P.J., de Vet, H.C., Wollersheim, H.: Measurement of patient safety: a systematic review of the reliability and validity of adverse event detection with record review. BMJ Open **6**(8), e011078 (2016)
50. Harrison, D.C., Seah, W.K., Rayudu, R.: Rare event detection and propagation in wireless sensor networks. ACM Comput. Surv. (CSUR) **48**(4), 1–22 (2016)
51. Hasan, M., Orgun, M.A., Schwitter, R.: A survey on real-time event detection from the twitter data stream. J. Inform. Sci. **44**(4), 443–463 (2018)
52. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
53. Horng, S., Sontag, D.A., Halpern, Y., Jernite, Y., Shapiro, N.I., Nathanson, L.A.: Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning. PloS one **12**(4), e0174708 (2017)
54. Hu, X., Ma, W., Chen, C., Wen, S., Zhang, J., Xiang, Y., Fei, G.: Event detection in online social network: methodologies, state-of-art, and evolution. Comput. Sci. Rev. **46**, 100500 (2022)
55. Hussein, R., Lerner, A., Ryser, A., Bürgi, L.D., Blarer, A., Cudré-Mauroux, P.: Graphinc: graph pattern mining at network speed. Proc. ACM Manag. Data **1**(2), 1–28 (2023)
56. Italia, T.: Telecom Italia big data challenge. URL https://dandelion.eu/datamine/open-big-data (2015)
57. Jagannatha, A.N., Yu, H.: Bidirectional RNN for medical event detection in electronic health records. In: Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting, vol. 2016, p. 473. NIH Public Access (2016)
58. Jepsen, T., Lerner, A., Pedone, F., Soulé, R., Cudré-Mauroux, P.: In-network support for transaction triaging. Proc. VLDB Endow. **14**(9), 1626–1639 (2021)
59. Jiang, J., Jiang, J., Cui, B., Zhang, C.: Tencentboost: A gradient boosting tree system with parameter server. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pp. 281-284. IEEE (2017)
60. Jin, D., Yu, Z., He, D., Yang, C., Philip, S.Y., Han, J.: GCN for HIN via implicit utilization of attention and meta-paths. IEEE Trans. Knowl. Data Eng. **35**(4), 3925–3937 (2021)
61. Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.W.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G.: Mimic-III, a freely accessible critical care database. Sci. data **3**(1), 1–9 (2016)
62. Karæz, Y., Ivanova, M., Zhang, Y., Manegold, S., Kersten, M.L.: Lazy ETL in action: ETL technology dates scientific data. Proc. VLDB Endow. **6**(12), 1286–1289 (2013). https://doi.org/10.14778/2536274.2536297
63. Khadanga, S., Aggarwal, K., Joty, S., Srivastava, J.: Using clinical notes with time series data for icu management. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6432-6437 (2019)
64. Khalifa, Y., Mandic, D., Sejdić, E.: A review of hidden Markov models and recurrent neural networks for event detection and localization in biomedical signals. Inform. Fusion **69**, 52–72 (2021)
65. Khayati, M., Cudré-Mauroux, P., Böhlen, M.H.: Using lowly correlated time series to recover missing values in time series: a comparison between svd and cd. In: Advances in Spatial and Temporal Databases - 15th International Symposium, SSTD 2015, Seoul, South Korea, August 26-28, 2015. Proceedings (2015)
66. Khayati, M., Lerner, A., Tymchenko, Z., Cudré-Mauroux, P.: Mind the gap: an experimental evaluation of imputation of missing values techniques in time series. Proc. VLDB Endow. **13**(5), 768–782 (2020). https://doi.org/10.14778/3377369.3377383
67. Khelifati, A., Khayati, M., DignÃs, A., Difallah, D., CudrÃ-Mauroux, P.: TSM-bench: benchmarking time series database systems for monitoring applications. Proc. VLDB Endow. **16**(11), 3363–3376 (2023)
68. Kloster, K., Gleich, D.F.: Heat kernel based community detection. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1386-1395 (2014)
69. Lavin, A., Ahmad, S.: Evaluating real-time anomaly detection algorithms-the numenta anomaly benchmark. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA), pp. 38-44. IEEE (2015)
70. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L.: Handwritten digit recognition with a back-propagation network. Adv. Neural Inform. Process. Syst. **2** (1989)
71. Lee, R., Sumiya, K.: Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In: Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks, pp. 1-10 (2010)
72. Lee, S., Lerner, A., Ryser, A., Park, K., Jeon, C., Park, J., Song, Y.H., Cudré-Mauroux, P.: X-SSD: A storage system with native support for database logging and replication. In: Z.G. Ives, A. Bonifati, A.E. Abbadi (eds.) SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022, pp. 988-1002. ACM (2022)
73. Lerner, A., Hussein, R., Ryser, A., Lee, S., Cudré-Mauroux, P.: Networking and storage: the next computing elements in exascale systems? IEEE Data Eng. Bull. **43**(1), 60–71 (2020)
74. Lin, D., Wong, R.C.W., Xie, M., Wei, V.J.: Index-free approach with theoretical guarantee for efficient random walk with restart query. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 913-924. IEEE (2020)
75. Liu, H., Bielinski, S.J., Sohn, S., Murphy, S., Wagholikar, K.B., Jonnalagadda, S.R., Ravikumar, K., Wu, S.T., Kullo, I.J., Chute, C.G.: An information extraction framework for cohort identification using electronic health records. AMIA Summits Transl. Sci. Proc. **2013**, 149 (2013)
76. Liu, Y., Sarabi, A., Zhang, J., Naghizadeh, P., Karir, M., Bailey, M., Liu, M.: Cloudy with a chance of breach: Forecasting cyber security incidents. In: 24th USENIX Security Symposium (USENIX Security 15), pp. 1009-1024 (2015)
77. Long, R., Wang, H., Chen, Y., Jin, O., Yu, Y.: Towards effective event detection, tracking and summarization on microblog data. In: Web-Age Information Management: 12th International Con-

ference, WAIM 2011, Wuhan, China, September 14-16, 2011. Proceedings 12, pp. 652-663. Springer (2011)

78. Ma, Y., Li, Q., Yang, Z., Lu, Z., Pan, H., Chan, A.B.: An svd-based multimodal clustering method for social event detection. In: 2015 31st IEEE International Conference on Data Engineering Workshops, pp. 202-209. IEEE (2015)
79. Madani, A., Boussaid, O., Zegour, D.E., et al.: What's happening: a survey of tweets event detection. In: Proc. Intl. Conf. on Communications, Computation, Networks and Technologies (INNOV), pp. 16-22 (2014)
80. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. (2013) arXiv:1301.3781
81. Mortensen, K.O., Zardbani, F., Haque, M.A., Agustsson, S.Y., Mottin, D., Hofmann, P., Karras, P.: Marigold: efficient k-means clustering in high dimensions. Proc. VLDB Endow. **16**(7), 1740–1748 (2023)
82. Nasridinov, A., Ihm, S.Y., Jeong, Y.S., Park, Y.H.: Event detection in wireless sensor networks: Survey and challenges. In: Mobile, Ubiquitous, and Intelligent Computing: MUSIC 2013, pp. 585-590. Springer (2014)
83. Nurwidyantoro, A., Winarko, E.: Event detection in social media: a survey. In: International Conference on ICT for Smart Society, pp. 1-5. IEEE (2013)
84. Oh, S., McCloskey, S., Kim, I., Vahdat, A., Cannons, K.J., Hajimirsadeghi, H., Mori, G., Perera, A., Pandey, M., Corso, J.J.: Multimedia event detection with multimodal feature fusion and temporal concept localization. Mach. Vision Appl. **25**(1), 49–69 (2014)
85. Ordonez, C., Cereghini, P.: SQLEM: fast clustering in SQL using the EM algorithm. ACM Sigmod Record **29**(2), 559–570 (2000)
86. Ould-Ahmed-Vall, E., Ferri, B.H., Riley, G.F.: Distributed fault-tolerance for event detection using heterogeneous wireless sensor networks. IEEE Trans. Mobile Comput. **11**(12), 1994–2007 (2011)
87. Over, P.: Trecvid 2013-an overview of the goals, tasks, data, evaluation mechanisms and metrics (2013)
88. Over, P., Awad, G., Fiscus, J., Antonishek, B., Michel, M., Smeaton, A., Kraaij, W., Quenot, G.: Trecvid 2011 - an overview of the goals, tasks, data, evaluation mechanisms, and metrics (2012)
89. Pan, B., Zheng, Y., Wilkie, D., Shahabi, C.: Crowd sensing of traffic anomalies based on human mobility and social media. In: Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems, pp. 344-353 (2013)
90. Papadopoulos, S., Schinas, E., Mezaris, V., Troncy, R., Kompatsiaris, I.: The 2012 social event detection dataset. In: Proceedings of the 4th ACM Multimedia Systems Conference, pp. 102-107 (2013)
91. Papadopoulos, S., Troncy, R., Mezaris, V., Huet, B., Kompatsiaris, I.: Social event detection at mediaeval 2011: Challenges, dataset and evaluation. In: MediaEval (2011)
92. Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., Kim, Y.: Data synthesis based on generative adversarial networks. Proc. VLDB Endow. **11**(10), 1071–1083 (2018). https://doi.org/10.14778/3231751.3231757
93. Park, S., Jayaraman, S.: Wearables: fundamentals, advancements, and a roadmap for the future. In: Sazonov, E. (ed.) Wearable sensors, 2nd edn., pp. 3–27. Academic Press, Oxford (2021)
94. Peng, H., Li, J., Gong, Q., Song, Y., Ning, Y., Lai, K., Yu, P.S.: Fine-grained event categorization with heterogeneous graph convolutional networks. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 3238-3245 (2019)
95. Peng, H., Li, J., Song, Y., Yang, R., Ranjan, R., Yu, P.S., He, L.: Streaming social event detection and evolution discovery in heterogeneous information networks. ACM Trans. Knowl. Discov. Data (TKDD) **15**(5), 1–33 (2021)
96. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543 (2014)
97. Petkos, G., Papadopoulos, S., Kompatsiaris, Y.: Social event detection using multimodal clustering and integrating supervisory signals. In: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, pp. 1-8 (2012)
98. Petkos, G., Papadopoulos, S., Mezaris, V., Kompatsiaris, Y.: Social event detection at mediaeval 2014: Challenges, datasets, and evaluation. In: MediaEval. Citeseer (2014)
99. Petkos, G., Papadopoulos, S., Schinas, E., Kompatsiaris, Y.: Graph-based multimodal clustering for social event detection in large collections of images. In: International Conference on Multimedia Modeling, pp. 146-158. Springer (2014)
100. Petrovic, S.: A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters. In: Proceedings of the 11th Nordic workshop of secure IT systems, vol. 2006, pp. 53-64. Citeseer (2006)
101. Porumb, M., Stranges, S., Pescapè, A., Pecchia, L.: Precision medicine and artificial intelligence: a pilot study on deep learning for hypoglycemic events detection based on ecg. Sci. Rep. **10**(1), 170 (2020)
102. Qiao, M., Bian, W., Da Xu, R.Y., Tao, D.: Diversified hidden Markov models for sequential labeling. IEEE Trans. Knowl. Data Eng. **27**(11), 2947–2960 (2015)
103. Rendón, E., Abundez, I.M., Gutierrez, C., Zagal, S.D., Arizmendi, A., Quiroz, E.M., Arzate, H.E.: A comparison of internal and external cluster validation indexes. In: Proceedings of the 2011 American Conference, San Francisco, CA, USA, vol. 29, pp. 1-10 (2011)
104. Reuter, T., Cimiano, P.: Event-based classification of social media streams. In: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, pp. 1-8 (2012)
105. Ritter, A., Mausam, Etzioni, O., Clark, S.: Open domain event extraction from twitter. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1104-1112 (2012)
106. Rodrigues, F., Markou, I., Pereira, F.C.: Combining time-series and textual data for taxi demand prediction in event areas: a deep learning approach. Inform. Fusion **49**, 120–129 (2019)
107. Ros, F., Riad, R., Guillaume, S.: PDBI: a partitioning Davies-Bouldin index for clustering evaluation. Neurocomputing **528**, 178–199 (2023)
108. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
109. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web, pp. 851-860 (2010)
110. Schinas, M., Papadopoulos, S., Petkos, G., Kompatsiaris, Y., Mitkas, P.A.: Multimodal graph-based event detection and summarization in social media streams. In: Proceedings of the 23rd ACM international conference on Multimedia, pp. 189-192 (2015)
111. Schmidl, S., Wenig, P., Papenbrock, T.: Anomaly detection in time series: a comprehensive evaluation. Proc. VLDB Endow. **15**(9), 1779–1797 (2022). https://doi.org/10.14778/3538598.3538602
112. Serizel, R., Turpault, N.: Sound event detection from partially annotated data: trends and challenges. In: IcETRAN conference (2019)
113. Simitsis, A., Vassiliadis, P., Sellis, T.K.: Optimizing ETL processes in data warehouses. In: K. Aberer, M.J. Franklin, S. Nishio

(eds.) Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan, pp. 564-575. IEEE Computer Society (2005). 10.1109/ICDE.2005.103

114. Steinkrau, D., Simard, P.Y., Buck, I.: Using gpus for machine learning algorithms. In: Eighth International Conference on Document Analysis and Recognition (ICDAR 2005), pp. 1115-1119. IEEE Computer Society (2005)
115. Sun, N., Zhang, J., Rimba, P., Gao, S., Zhang, L.Y., Xiang, Y.: Data-driven cybersecurity incident prediction: a survey. IEEE Commun. Surv. Tutor. **21**(2), 1744–1772 (2018)
116. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Heterogeneous information networks: the past, the present, and the future. Proc. VLDB Endow. **15**(12), 3807–3811 (2022)
117. Tata, S., Patel, J.M.: Estimating the selectivity of TF-IDF based cosine similarity predicates. ACM Sigmod Record **36**(2), 7–12 (2007)
118. Tonon, A., Cudré-Mauroux, P., Blarer, A., Lenders, V., Motik, B.: Armatweet: detecting events by semantic tweet analysis. In: The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28-June 1, 2017, Proceedings, Part II 14, pp. 138-153. Springer (2017)
119. Troncy, R., Malocha, B., Fialho, A.T.: Linking events with media. In: Proceedings of the 6th international conference on semantic systems, pp. 1-4 (2010)
120. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: Proceedings of the 26th annual international conference on machine learning, pp. 1073-1080 (2009)
121. Wang, W., Zhang, M., Chen, G., Jagadish, H., Ooi, B.C., Tan, K.L.: Database meets deep learning: challenges and opportunities. ACM Sigmod Record **45**(2), 17–22 (2016)
122. Wang, Y., Sundaram, H., Xie, L.: Social event detection with interaction graph modeling. In: Proceedings of the 20th ACM international conference on Multimedia, pp. 865-868 (2012)
123. Weiler, A., Grossniklaus, M., Scholl, M.H.: Survey and experimental analysis of event detection techniques for twitter. Comput. J. **60**(3), 329–346 (2017)
124. Weng, J., Lee, B.S.: Event detection in twitter. In: Proceedings of the international AAAI conference on web and social media, vol. 5, pp. 401-408 (2011)
125. Wu, S., Bondugula, S., Luisier, F., Zhuang, X., Natarajan, P.: Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2665-2672 (2014)
126. Xiao, K., Qian, Z., Qin, B.: A survey of data representation for multi-modality event detection and evolution. Appl. Sci. **12**(4), 2204 (2022)
127. Xie, L., Sundaram, H., Campbell, M.: Event mining in multimedia streams. Proc. IEEE **96**(4), 623–647 (2008)
128. Yang, H., Kuang, L., Xia, F.: Multimodal temporal-clinical note network for mortality prediction. J. Biomed. Seman. **12**(1), 1–14 (2021)
129. Yang, Y., Pierce, T., Carbonell, J.: A study of retrospective and on-line event detection. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 28-36 (1998)
130. Yang, Z., Li, Q., Liu, W., Ma, Y., Cheng, M.: Dual graph regularized NMF model for social event detection from Flickr data. World Wide Web **20**(5), 995–1015 (2017)
131. Yang, Z., Li, Q., Lu, Z., Ma, Y., Gong, Z., Pan, H.: Semi-supervised multimodal clustering algorithm integrating label signals for social event detection. In: 2015 IEEE International Conference on Multimedia Big Data, pp. 32-39. IEEE (2015)
132. Yılmaz, Y., Hero, A.O.: Multimodal event detection in twitter hashtag networks. J. Signal Process. Syst. **90**(2), 185–200 (2018)
133. Younessian, E., Mitamura, T., Hauptmann, A.: Multimodal knowledge-based analysis in multimedia event detection. In: Proceedings of the 2nd ACM International conference on multimedia retrieval, pp. 1-8 (2012)
134. Zaharieva, M., Zeppelzauer, M., Breiteneder, C.: Automated social event detection in large photo collections. In: Proceedings of the 3rd ACM conference on International conference on multimedia retrieval, pp. 167-174 (2013)
135. Zauner, C.: Implementation and benchmarking of perceptual image hash functions (2010)
136. Zhang, B., Peng, B., Qiu, J.: Model-centric computation abstractions in machine learning applications. In: Proceedings of the 3rd ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond, pp. 1-4 (2016)
137. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. ACM Sigmod Record **25**(2), 103–114 (1996)
138. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: Proceedings of the eleventh international conference on Information and knowledge management, pp. 515-524 (2002)
139. Zheng, Y.: Methodologies for cross-domain data fusion: an overview. IEEE Trans. Big Data **1**(1), 16–34 (2015)