electronics

Article

Multisource Information-Based Social Event Detection Using Integrated Deep Neural Architectures and Hybrid Text Representation

Mudasir Ahmad Wani [1], Sarang Shaikh [2], Kashish Ara Shakil [3,*] and Abdulaziz Alsahli [1]

[1] Department of Computer Sciences, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318, Saudi Arabia; mawani@imamu.edu.sa (M.A.W.); aaalsahli@imamu.edu.sa (A.A.)
[2] Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), 2815 Gjøvik, Norway; sarang.shaikh@ntnu.no
[3] Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P. O. Box 84428, Riyadh 11671, Saudi Arabia
* Correspondence: kashakil@pnu.edu.sa

Abstract

Event detection is crucial for timely crisis response, public safety, and combating misinformation by extracting critical signals from real-time social data. The user-generated content on social media platforms has created a critical need for intelligent systems capable of accurately detecting and classifying real-world events across diverse domains. This paper presents a fusion-driven framework for social event detection, leveraging textual data from diverse social media sources and a suite of deep neural architectures to achieve state-of-the-art performance in classifying diverse event categories from noisy, unstructured social media streams. The proposed system collects information from heterogeneous domains, including medical conditions, natural disasters, and sports events, to improve both feature richness and contextual awareness. Our method applies a hybridized deep learning strategy, specifically a customized layered CNN+LSTM architecture. This fusion enables the model to learn spatial and temporal dependencies simultaneously, thus enhancing classification performance. To benchmark effectiveness, we evaluated standalone architectures including CNN, RNN, LSTM, BiLSTM, and BiRNN. Among individual models, LSTM and BiLSTM yielded the top performance, achieving up to 95.3% and 95.8% precision, 95.5% and 96.1% recall, 95.4% and 95.9% accuracy, 95.4% and 96.0% F1-score, 95.6% and 96.2% F1.5, and 95.5% and 96.1% F2-scores, respectively. However, these models still showed variability across event types due to limited contextual generalization from a single-layered perspective. In contrast, the proposed fused CNN and LSTM model recorded promising performance with superior F1-scores on 14 independent classes, achieving 100% on Acne, Earthquake, Landslide, Ringworm, and Vitiligo, 99% on Depression, 98% on Badminton and Tennis, 97% on Cricket, and 96% on FIFA (2022). These results demonstrate the model's robustness across both high- and low-frequency classes, as well as its ability to generalize across varying linguistic and topical contexts. The collection of data from multiple heterogeneous sources not only mitigates the limitations posed by domain-specific biases and class imbalance but also enriches semantic representation and improves the model's adaptability to emerging events. This research thus establishes a scalable foundation for intelligent, real-time event classification systems in domains such as public safety, health surveillance, and crisis management.

Check for updates

Academic Editor: Jichai Jeong

Received: 26 July 2026
Revised: 13 October 2026
Accepted: 17 October 2026
Published: 27 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: multisource fusion; generative AI; Natural Language Processing (NLP); event detection; Artificial Intelligence (AI); deep learning; Convolutional Neural Network (CNN); Long Short-Term Memory (LSTM)

1. Introduction

Event detection, the automated process of identifying and characterizing occurrences from Online Social Network (OSN) data, has evolved into a critical computational discipline [1]. It aims to construct a coherent narrative of an incident by answering the fundamental questions of what happened, when, where, to whom, and why [2]. This capability transcends mere data analysis, offering transformative potential for real-time decision-making across an expansive array of sectors. The imperative for such systems is clear: in an increasingly dynamic world, the speed and accuracy of event awareness directly impact outcomes in public safety, health, economic strategy, and social discourse.

The advent of social media has fundamentally altered the information landscape, serving as a global, distributed sensor network [3]. Platforms like Twitter (X), Facebook, and Instagram generate a continuous, high-volume stream of user-generated content comprising text, images, videos, and metadata. These data embody immediate, ground-level perspectives that are often absent from traditional, institutionally curated sources like news wires or official reports, which can suffer from inherent delays and filtering. Social media's real-time nature and democratized access provide an unparalleled resource for detecting events at their incipient stages, enabling responses that are more agile and informed by grassroots reporting [4].

The computational challenge lies in distilling signal from noise within this vast, unstructured, and often noisy data corpus. This is where deep learning, a subfield of machine learning inspired by artificial neural networks, has become indispensable. Unlike traditional methods reliant on manually engineered features, deep learning models autonomously learn hierarchical representations and complex patterns directly from raw data. Architectures such as Convolutional Neural Networks (CNNs) excel at processing spatial data like images to identify disaster damage or sports event scenes [5–7]. Recurrent Neural Networks (RNNs), and particularly their advanced variants like Long Short-Term Memory (LSTM) networks, are adept at modeling sequential text and temporal dynamics in event narratives [8–10]. More recently, transformer-based models (e.g., BERT, GPT) have set new benchmarks by leveraging self-attention mechanisms to capture nuanced contextual relationships and semantic meaning within text, proving exceptionally powerful for understanding social media linguistics [11].

Natural Language Processing (NLP) operates as the essential interpreter within this framework, transforming unstructured text into structured, actionable insights [12]. Through a suite of techniques—including named entity recognition (NER), sentiment analysis, topic modeling, and geolocation extraction—NLP algorithms parse colloquial language, slang, and multilingual posts to identify keywords, emotional tones, and contextual patterns indicative of specific events. For instance, during earthquakes, a surge in geotagged posts containing terms like "tremor," "earthquake," or "shaking" can be triangulated to estimate epicenters and assess intensity in near real time, as demonstrated in studies of Twitter activity during seismic events [13]. Similarly, during floods or hurricanes, NLP can filter posts referencing "water level," "evacuation route," or "storm surge," while computer vision models can analyze shared images to classify damage severity, providing critical situational awareness for emergency responders [14].

The application domain extends far beyond disasters. In the realm of sports events, event detection systems monitor viral spikes in conversation around tournaments like the FIFA World Cup or grand slams in tennis. Analyzing sentiment and volume around specific matches or players allows broadcasters and marketers to gauge public engagement in real time and tailor content accordingly [15]. Perhaps one of the most socially impactful applications is in public health surveillance for disease events. By analyzing anonymized, population-level trends, researchers can detect early signals of outbreaks or track the psychosocial impact of diseases. Discussions about symptom management for depression or anxiety on social platforms have been shown to correlate with clinical trends, offering a complementary tool for mental health monitoring [16]. Similarly, public concerns and information-seeking behaviors related to dermatological conditions like acne, eczema, ringworm, or vitiligo can be mapped, identifying communities with high discussion volume that may indicate unmet needs or misinformation spread [17].

The technical pipeline for social media event detection follows a structured, though iterative, sequence. It begins with data acquisition, collecting streams via platform APIs or utilizing existing curated datasets. This is followed by intensive data preprocessing—cleaning text, removing duplicates and bots, tokenizing, and normalizing. The feature extraction phase then converts these raw data into numerical representations, such as word embeddings (Word2Vec, GloVe, BERT embeddings) or multimodal features combining text with image or network data. In the modeling phase, deep learning architectures are trained on labeled datasets to classify posts or clusters into event categories. Finally, the system undergoes rigorous evaluation using metrics like precision, recall, and F1-score, with results informing model refinement. A significant advantage of this pipeline is its adaptability; a well-designed system can be fine-tuned with new data to detect emerging event types, from a novel disease outbreak to a new social trend.

Despite its promise, the field grapples with challenges including data veracity (misinformation and rumours), platform bias, multilingual processing, and ethical concerns regarding privacy and surveillance. Addressing these requires a multidisciplinary approach, combining technical innovation with social science insights.

In this context, this study proposes an AI-driven framework based on multisource information for social media event classification. The framework is designed to accommodate diverse event categories, including natural disasters, public health crises, and major social and sporting events. By collecting heterogeneous social media data through a unified preprocessing pipeline and employing complementary textual representations, the framework aims to provide accurate and scalable event classification across different event domains. The proposed approach further incorporates neural architectures to effectively learn discriminative patterns from social media text and support reliable identification of event-related content.

To leverage multisource information in the proposed study, the social media datasets from different sources and event domains are first standardized and harmonized using a common preprocessing pipeline. The resulting textual data are then represented using TF-IDF and Word2Vec, capturing complementary lexical and semantic characteristics, respectively. These representations are combined within the proposed CNN–LSTM architecture to enable joint learning from term-level and semantic information. This design provides a unified learning framework for exploiting heterogeneous social media content while maintaining a consistent representation and classification process across diverse event categories.

Contributions

The major contributions of this study are as follows:

1. **Comprehensive Deep Neural Architecture Ensemble:** The research implements and rigorously evaluates a suite of deep learning models including CNN, RNN, LSTM, BiLSTM, and BiRNN within a unified experimental pipeline. This is extended by the design and implementation of a Customized Hybrid CNN+LSTM Model, which architecturally combines the local feature extraction prowess of CNNs with the sequential context modeling of LSTMs for superior pattern recognition in event-related text.
2. **Strategic Data Augmentation with Generative Language Models:** To address the critical challenges of data scarcity and severe class imbalance inherent in social media event datasets, this work employs advanced generative language models for synthetic data generation. The generated samples undergo a rigorous human-in-the-loop validation process with domain experts, ensuring semantic fidelity and contextual relevance, thereby creating balanced and high-quality training corpora.
3. **Systematic Embedding Technique Analysis:** The study conducts a meticulous comparative analysis of fundamental feature embedding techniques, namely, Word2Vec and TF-IDF. This investigation reveals the distinct performance characteristics of each method across different model architectures and event types, providing empirical guidance for selecting optimal text representation strategies in social media analysis tasks.
4. **Extensive Experimental Protocol and Granular Performance Analysis:** The work establishes a comprehensive experimental protocol, meticulously recording and analyzing Training and Validation Performance per Epoch to diagnose model behavior, prevent overfitting, and ensure optimal convergence. Furthermore, we conducted detailed class-wise performance experiments across three major event domains, including Natural Disasters (Earthquake, Flood, Landslide, Hurricane), Sports Events (Badminton, Cricket, FIFA, Tennis), and Disease Events (Acne, Depression, Eczema, Ringworm, Vitiligo, COVID), for models including CNN, RNN, and LSTM. This granular analysis provides critical insights into model efficacy for specific event categories.

The present study extends our previous work [18], which focused exclusively on two event categories: COVID-19 and floods. In contrast, this study broadens the scope of social media-based event detection to 14 categories across three domains: natural disasters (earthquake, flood, landslide, and hurricane), sports events (badminton, cricket, FIFA, and tennis), and disease-related events (acne, depression, eczema, ringworm, vitiligo, and COVID-19). Furthermore, while our previous study employed an integrated CNN–LSTM architecture, the present study conducts a broader comparative evaluation of six deep learning architectures: CNN, RNN, LSTM, BiRNN, BiLSTM, and the hybrid CNN–LSTM model. This expanded investigation provides a more comprehensive assessment of their effectiveness in detecting diverse event categories from noisy social media text.

The rest of the paper is organized as follows: in Section 2, we perform a background study about event detection on different social media platforms. We perform event data collection and preprocessing in Section 3. Section 4 discusses feature engineering. In Section 5, we discuss the proposed architectural framework and the experimental setup for deep learning toward designing the detection system. In Section 6, we present the experimental results and provide a detailed analysis of the outcomes. Section 7 provides a comparative study with existing studies and methods. Finally, a set of observations and the conclusion of our work are presented in Section 8.

2. Background Study

The use of social networking platforms, particularly Twitter, for event detection and analysis has received significant attention in the literature. An event is typically characterized by the 4Ws: who, what, when, and where [19]. Event-related tweets often contain terms that reflect these aspects, categorized into semantic types such as entity names (who),

hashtags (what), temporal expressions (when), locations (where), and mentions (who). These semantic types vary in their importance for event description, influencing detection algorithms. However, many studies on social media event detection overlook this classification. Researchers have explored a range of approaches and techniques to leverage the rich data available on these platforms, with a growing focus on event detection. In addition to applications like fake news detection [20], identifying depression in online users [21,22], spam identification [23], stance detection [24], and behavior analysis [25], significant attention is now being directed towards event detection.

This literature review provides an overview of key studies focusing on event detection using social networking platforms. A study [13] introduced a system called "EARWORM" that utilized Twitter data to detect earthquakes in Japan. The authors proposed a method to filter and analyze tweets in real time to identify earthquake-related information. They used keyword-based filtering and applied a machine learning algorithm to classify tweets as earthquake-related or not. The study demonstrated the potential of Twitter as a valuable source for detecting natural disasters. The authors in [12] conducted a comprehensive survey on processing social media messages in the context of mass emergencies. The authors reviewed various techniques and frameworks used for event detection and classification from Twitter data. They highlighted the challenges and opportunities in utilizing social media data for real-time event monitoring and discussed the role of machine learning and NLP techniques in event detection. The authors in [26] proposed a probabilistic model that identified bursts of tweets related to specific events and tracked the evolution of those events over time. The study demonstrated the effectiveness of the bursty topic model in capturing event-related discussions on Twitter. Similarly, the authors in [27] proposed TwitterMonitor, a system for trend detection over the Twitter stream. The authors focused on identifying bursts of tweets associated with specific topics or events. They introduced the concept of burstiness and content similarity to detect and track emerging events in real time. The study demonstrated the effectiveness of Twitter as a source of real-time event information.

A recent survey [28] on Twitter streams categorizes existing event detection techniques based on event type (specified or unspecified), detection task (retrospective or new event detection), detection methods (supervised or unsupervised), and their applications. A study [29] used clustering methods to detect events in news streams, leveraging temporal and content features for improved event clustering. Another study [30] extended this work to event tracking using supervised learning, employing k-nearest neighbor and decision-tree methods to efficiently label incoming news with minimal training data. Another study [31] conducted a user study exploring the integration of news articles and user-generated content for event-related information retrieval, assessing how combining these sources meets user needs. Similarly, the authors in [32] compared news streams and tweets in event reporting, using discrete dynamic topic modeling and a Hidden Markov Model for event detection, followed by clustering news documents and analyzing the timeliness of tweets and articles in reporting events.

Towards an early alert of an outbreak, the authors in study [33] propose an early warning system for meningitis or COVID-19 epidemics using Twitter data. They employ NLP techniques and domain ontologies to classify tweets, followed by geolocation and timestamp extraction from metadata. Real-time data collection is implemented via Kafka, and tweet identification is performed using SVM and CNN models, with the CNN model achieving 0.99 accuracy. The approach integrates ontology, machine learning, and Kafka for efficient epidemic monitoring. One more study [34] used space-time scan statistics (STSS) to detect events on Twitter without predefined keywords. Applied to the 2013 London helicopter crash, STSS identified a short-lived but informative cluster of tweets. The method also detected other events like football matches and transportation delays, showcasing its

effectiveness for spatio-temporal event analysis. In another similar study [35] the authors propose event detection algorithms for short- and long-term Twitter trends, using dynamic metrics like time-sensitive IDF and fuzzy representations to capture temporal shifts. Events are detected via multi-assignment spectral graph partitioning, enabling words to belong to multiple clusters. Similarly, a study [2] proposes a novel framework to detect composite social events by integrating content, time, and location dimensions using a location-time-constrained topic (LTT) model. This model represents social messages as topic distributions and measures similarity via distribution distances. Events are identified through efficient similarity joins, accelerated by a variable-dimensional extendible hash. Extensive experiments demonstrate the framework's effectiveness and efficiency in analyzing social media streams for crisis management and decision-making. The authors in [36] presented Iktishaf, a Spark-based tool for detecting traffic events in Arabic tweets, using machine learning to classify eight event types. With 2.5 million tweets, it identifies events like the KSA National Day and Al-Haramain train launch, combining AI, big data, and human cognition for Arabic event detection. Further discussing deep machine learning, the authors in [37] proposed a Dual-CNN, a semantically enhanced deep learning model for crisis event detection from social media, achieving over 79% F-measure for event types but 61% for fine-grained details, competing with traditional models like SVM. Similarly, another study [38] uses word embeddings to map tweets into numerical vectors and classify them into non-traffic, traffic incidents, or traffic information using CNN and RNN models. A dataset of 51,100 tweets is collected, labeled, and publicly released. The approach demonstrates superior performance through extensive evaluation. Similarly, another study [18] proposed an optimized CNN–LSTM framework for crisis event detection in noisy social media streams. The present study extends this work by broadening the scope of event detection to encompass a wider range of event categories using textual data from social media.

These studies demonstrate the diverse range of social networking platforms that can be leveraged for event detection. By analyzing the content, interactions, and temporal patterns present on these platforms, researchers have showcased the effectiveness of utilizing social networking data beyond Twitter for event detection.

2.1. Areas of Event Detection

Event detection refers to the process of identifying and categorizing specific occurrences or incidents from a given data source. In the context of social media, event detection involves extracting meaningful events from the vast amount of user-generated content. Figure 1 presents some examples of events that can be detected using social media data:

2.1.1. Natural Disasters

Event detection can identify and track natural disasters such as earthquakes, hurricanes, floods, or wildfires by monitoring social media posts that mention relevant keywords, locations, or include images/videos related to the event. This information can be crucial for disaster response and emergency management.

2.1.2. Social Unrest

Event detection can identify instances of protests, riots, or political rallies by analyzing social media posts that contain hashtags, mentions, or sentiment analysis related to social or political movements. This can provide insights into public sentiment and social dynamics.

2.1.3. Sports Events

Event detection can identify and track sports events such as football matches, basketball games, or major tournaments by monitoring social media conversations, hashtags, or user

check-ins at stadiums. This can help gauge fan engagement and sentiment during live sporting events.

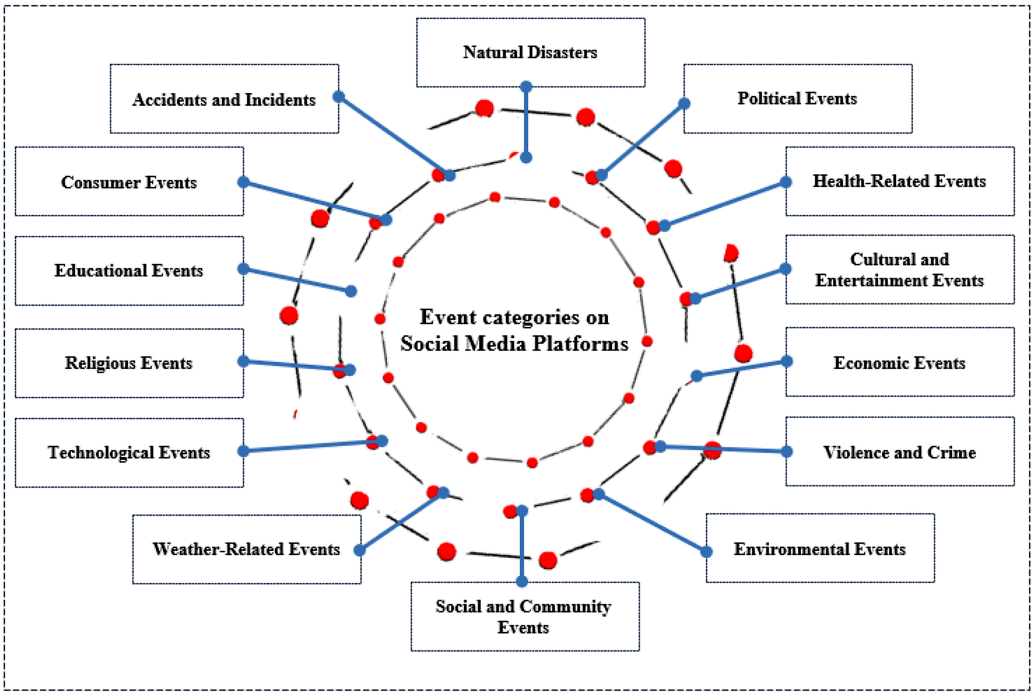


Figure 1. Different event categories on social media platforms.

2.1.4. Product Launches

Event detection can identify new product launches, releases, or major announcements by monitoring social media platforms for posts, reviews, or discussions related to specific brands, products, or industries. This information can be valuable for market analysis and brand reputation management.

2.1.5. Entertainment Events

Event detection can identify events related to movies, music concerts, festivals, or cultural shows by monitoring social media conversations, ticket sales, or posts from attendees. This can provide insights into popular entertainment trends and audience preferences.

2.1.6. Disease Outbreaks

Event detection can identify and track disease outbreaks by monitoring social media posts for symptoms, mentions of illnesses, or discussions related to public health issues. This can aid in early detection and response to potential epidemics.

2.1.7. Traffic Incidents

Event detection can identify traffic accidents, road closures, or congestion by analyzing social media posts that mention traffic-related keywords, locations, or real-time updates from users on the road. This information can help in route planning and traffic management.

3. Data Collection and Preprocessing

In event classification using textual data, data collection and preprocessing are essential steps that directly impact the model's accuracy and effectiveness. Textual data are often unstructured, containing noise, redundant information, and varying formats, making it challenging for models to interpret without proper preparation.

Effective data collection ensures that the dataset is comprehensive and representative of real-world scenarios, covering a wide range of event types and linguistic variations. Preprocessing, which includes steps like tokenization, stop-word removal, stemming, and lemmatization, standardizes the text, reducing complexity while retaining important information. Together, data collection and preprocessing lay a solid foundation, enabling models to focus on meaningful patterns within the text and ultimately improving the classification of events with higher precision and reliability. In the following subsections, we will briefly discuss the data collection and preprocessing procedures employed for the proposed study.

Data are a fundamental component of any experimental research process, and their quality and diversity directly influence the effectiveness and generalizability of the developed models. In this study, we constructed a comprehensive multisource event detection dataset by integrating publicly available datasets from three major event categories: Natural Disasters, Sports Events, and Disease-Related Events. The distribution of the primary datasets used in this work is presented in Table 1.

For the Natural Disaster category, we collected data from multiple disaster-related datasets, including the Earthquake dataset [13], CrisisNLP dataset (landslides) [39], Hurricane dataset [40], and additional Flood-related datasets obtained from previous studies conducted in Pakistan and Sri Lanka [41,42]. These datasets contain social media posts associated with earthquakes, floods, hurricanes, and other crises, providing a diverse representation of disaster-related events and emergency responses.

The Sports category consists of datasets covering various sporting events and competitions. Specifically, we utilized the Sports datasets Badminton [43], Tennis [44], Cricket [45], and FIFA [46]. These datasets include social media discussions generated during sporting activities, tournaments, and international competitions, offering rich information regarding event dynamics and public engagement.

For the Disease category, we integrated several health-related datasets, including the Health dataset [17], Depression dataset [16], Dermatology dataset [47], and multiple COVID-19 datasets collected from prior research efforts [21,48]. These datasets encompass discussions related to public health concerns, infectious diseases, mental health conditions, dermatological disorders, and the COVID-19 pandemic, thereby providing broad coverage of disease-related events.

The primary objective of utilizing these heterogeneous datasets was to develop a unified multisource dataset capable of capturing diverse real-world events across multiple domains. By aggregating datasets from different studies and event categories, we aimed to improve dataset diversity, enhance model robustness, and facilitate the development of a generalized social event detection framework. In total, approximately 1.6 million samples were collected from the primary datasets listed in Table 1, while additional COVID-19 and Flood-related datasets were incorporated to enrich the Disease and Natural Disaster categories and improve event diversity.

Before incorporating any sample into the Initial Main Dataset (IMD), a rigorous validation and filtering procedure was performed. First, duplicate entries were identified and removed to ensure uniqueness. Subsequently, each sample was examined to verify its relevance to one of the target event categories: Natural Disaster, Sports, or Disease. Samples that were duplicated, incomplete, corrupted, or unrelated to the target categories were discarded. This process ensured that only relevant and high-quality data were retained.

Following data collection and validation, several preprocessing operations were applied to prepare the dataset for experimental analysis. These operations included text cleaning, normalization, tokenization, stop-word removal, stemming, and lemmatization using NLTK-based preprocessing techniques. The resulting dataset was then analyzed to identify category-level and class-level imbalances.

Table 1. Event detection dataset distribution.

Dataset/Event Type	Samples	Category	Description
Earthquake [13]	~170 K	Natural Disaster	Earthquake event posts.
CrisisNLP (Landslide) [39]	~60 K	Natural Disaster	Landslide-related tweets.
Hurricane Dataset [40]	~3.5 M	Natural Disaster	Hurricane event posts.
Flood Dataset [41,42]	~458 K	Natural Disaster	Flood-related posts.
Badminton [43]	~50 K	Sports-Events	Badminton-related posts.
Tennis [44]	~124 K	Sports-Events	Tennis-related tweets.
Cricket Dataset [45]	~150 K	Sports-Events	Cricket event posts.
FIFA Dataset [46]	~3 M	Sports-Events	FIFA event posts.
Health Dataset [17]	~1.6 M	Health-Crisis	Health-related posts.
COVID-19 Dataset [21,48]	~259 K	Health-Crisis	COVID-19 posts.
Depression Dataset [16]	~40 K	Health-Crisis	Depression-related posts.
Dermatology Dataset [47]	~200 K	Health-Crisis	Skin disease-related posts.
Total Samples	∼9.61 M	—	Multisource event detection dataset.

To mitigate data imbalance and improve representation across different event classes, Generative Language Models (GLMs) were employed to generate synthetic samples for underrepresented classes where necessary. The generated samples were carefully validated and incorporated into the dataset to improve diversity and balance. After augmentation, a final duplicate removal stage was conducted to ensure data quality and uniqueness.

The resulting Final Experimental Dataset (FED) represents a large-scale, balanced, and curated collection of multisource event data covering Natural Disaster events (Earthquake, Flood, Hurricane, and Crisis-related events), Sports events (Sports, Cricket, FIFA, and related competitions), and Disease-related events (COVID-19, Depression, Dermatology, and Health-related discussions). This dataset serves as the foundation for training and evaluating the proposed social event detection framework.

3.1. Multisource Data Acquisition

The complete workflow of the data collection, integration, preprocessing, augmentation, and validation stages is illustrated in Figure 2. Furthermore, Algorithm 1 presents the pseudocode of the proposed Data Acquisition and Curation Algorithm (DACA), which systematically aggregates multisource datasets, performs preprocessing and balancing operations, removes duplicate records, and produces the Final Experimental Dataset (FED) for subsequent experimentation.

Algorithm 1 Data Acquisition and Curation Algorithm (**DACA**)

BEGIN
Inputs:
Natural Disaster Datasets {*Flood*, *Landslides*, *Earthquake*, *Hurricane*},
Health-Crisis Datasets {*Acne, Depression, Covid, Vitligo, Eczema, Ringworm*},
Sports Datasets {*Badminton*, *Cricket*, *FIFA*, *Tennis*},
Generated Flood Data Samples (*gfds*)
Output: Final Experimental Dataset (*FED*)
Initialize:
imd ← [] Initial main dataset
pd ← [] Preprocessed dataset
FED ← [] Final Experimental Dataset

Algorithm 1 *Cont.*

gfds ← [] Generated flood data samples
for each instance i in $\{CD_1, CD_2, ..., CD_n\}$ **do**
    $imd[i] \leftarrow CD_i$
end for
for each instance j in $\{FD_1, FD_2, ..., FD_n\}$ **do**
    $imd[i] \leftarrow FD_j$
end for
for each k in imd **do**
    $pd[k] \leftarrow$ **perform_NLTK-Tasks**$(imd[k])$
end for
$nfs \leftarrow$ **Count**(flood instances in pd)
$cfs \leftarrow$ **Count**(covid instances in pd)
if $cfs > nfs$ **then**
    $amount_samples \leftarrow cfs - nfs$
    $pd \leftarrow pd \cup gfds[amount_samples]$
else
    $amount_samples \leftarrow nfs - cfs$
    $pd \leftarrow pd \cup$ generated_covid_samples[**amount_samples**]
end if
for each **sample** s in pd **do**
    $FED[s] \leftarrow pd[s]$
    if $FED[s] == FED[s+1]$ **then**
        Discard$(FED[s+1])$
    end if
end for
END

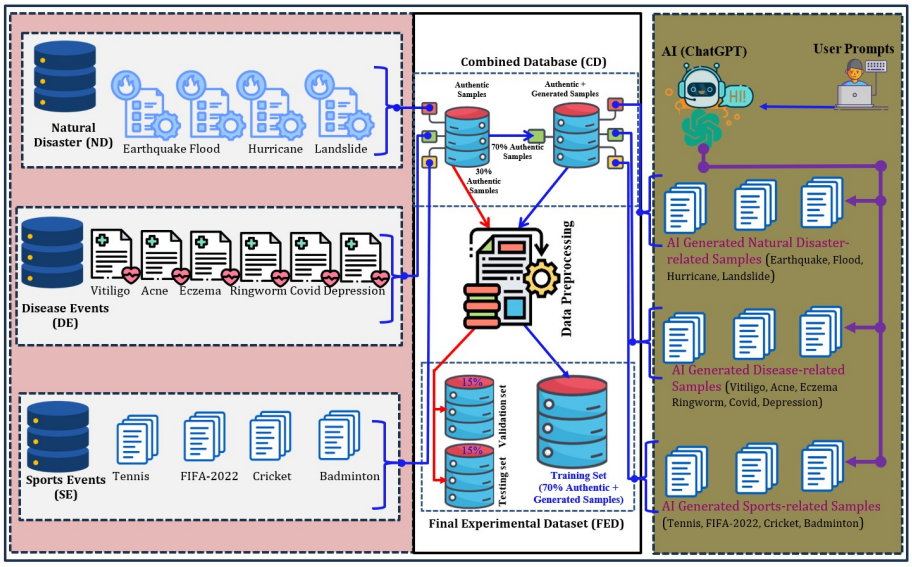


Figure 2. Detailed data collection procedure framework.

3.2. Data Preprocessing and Preparation

Once you have the data, it is essential to preprocess them before feeding them into the deep learning model. This preprocessing step involves cleaning the text by removing noise, such as URLs, hashtags, mentions, and emojis. Additionally, you may want to tokenize the text into individual words or use more advanced techniques like word embeddings to capture semantic meaning. Preprocessing is an essential step in natural language processing (NLP) tasks, and the Natural Language Toolkit (NLTK) [49] provides several useful

functionalities for this purpose. Here are the basic steps we applied in preprocessing text using NLTK:

- ***Tokenization:*** Tokenization is the process of splitting text into individual tokens, which are typically words or sentences. NLTK provides various tokenizers, such as word_tokenize for word-level tokenization and sent_tokenize for sentence-level tokenization. In a formal way, we can say that tokenization in NLTK splits text S (a string) into a sequence of tokens as follows:
  $T(S) = \{t_1, t_2, \ldots, t_n\}$, where t_i are the extracted tokens.
  Example Sentence S**:** "*Depression is a silent battle. Seeking help is a sign of strength! "*
  For word tokenization,

$$T(S) = Word_Tokenizer(S),\ T\big(\text{“}Depression\ is\ a\ silent\ battle.\ Seeking\ help\ is\ a\ sign\ of\ strength!\text{”}\big)$$
$$\Rightarrow \{\text{“}Depression\text{”},\ \text{“}is\text{”},\ \text{“}a\text{”},\ \text{“}silent\text{”},\ \text{“}battle\text{”},\ \text{“}.\text{”}, \ldots\}$$

  For sentence tokenization,
  $\mathbf{T_s(S)}$ = Sentence_Tokenizer(S),
  where $\mathbf{S}$ = "Depression is a silent battle. Seeking help is a sign of strength!"
  ⇒ {"Depression is a silent battle.","Seeking help is a sign of strength!"}
- ***Case Conversion:*** Text may contain a mixture of uppercase and lowercase characters. NLTK provides methods like lower() and upper() to convert text to lowercase or uppercase, respectively. This step helps to normalize the text and avoid duplication of words due to case differences.
  The formal representation of transforming a string S by applying a uniform case format (lowercase or uppercase) to each token is as follows:
  For lowercase conversion,
  $\mathbf{C(S)} = \{t_1, t_2, \ldots, t_n\}$
  $t_i = Lowercase(t_i), \forall t_i \in \mathbf{T(S)}$
  Example:
  $\mathbf{C}(\{\text{“}This\text{”},\text{“}\ is\text{”},\text{“}\ a\text{”},\text{“}\ Test\text{”}\})$
  $=> \{\text{“}this\text{”},\text{“}\ is\text{”},\text{“}\ a\text{”},\text{“}\ test\text{”}\}$
  For uppercase conversion,
  $\mathbf{C(S)} = \{t_1, t_2, \ldots, t_n\}$
  $t_i = Uppercase(t_i), \forall t_i \in \mathbf{T(S)}$
  Example:
  $\mathbf{C}(\{\text{“}This\text{”},\text{“}\ is\text{”},\text{“}\ a\text{”},\text{“}\ Test\text{”}\})$
  $=> \{\text{“}THIS\text{”},\text{“}\ IS\text{”},\text{“}\ A\text{”},\text{“}\ TEST\text{”}\}$
- ***Stop-Word Removal:*** Stop words are commonly occurring words that usually do not carry much meaningful information, such as "a," "the," and "is." NLTK provides a predefined list of stop words for multiple languages. By removing these stop words using the stop-word module, we can reduce noise and focus on important content words. The mathematical representation of stop-word removal is as follows:
  $R(S) = \{t_i \in S | t_i \notin SW\}$
  where $S = t_1, t_2, \ldots, t_n$ are tokenized input text and SW is a predefined set of stop words.
  Example:
  Input tokens*:* S = {"This", "is", "a", "sample", "text"}
  Stop words*:* SW = {"is", "a"}
  Output after stop-word removal*:*
  $R(S)$ = {"This", "sample", "text"}
- ***Punctuation Removal:*** Punctuation marks like periods, commas, and exclamation points do not usually contribute to the meaning of the text in many NLP tasks. NLTK provides methods like string punctuation to obtain a list of common punctuation

symbols, which can be removed using string manipulation techniques or regular expressions. The mathematical representation for the removal of punctuation marks will be somewhat similar to that for removing stop words from the text.

- ***Lemmatization/Stemming:*** Lemmatization and stemming are techniques to reduce words to their base or root form. NLTK offers several stemmers, such as the PorterStemmer and SnowballStemmer, and also provides lemmatization capabilities through the WordNetLemmatizer. These techniques help reduce inflected forms and normalize vocabulary. Mathematically, we can represent lemmatization as follows:
  $\mathbf{L(S)} = \{\mathbf{L}(t_i) \mid t_i \in \mathbf{S}\}$
  where $S = t_1, t_2, ..., t_n$ are tokenized input text and L is a lemmatization function that maps a word to its lemma.
  Example:
  Input tokens: $S = \{$"*running*","*better*","*cats*"$\}$
  Lemmatization: L("running")="run",L("better")="good", L("cats")="cat"
  Output: L(S) ={"run", "good", "cat"}
- ***Spell Checking:*** NLTK provides modules for spell checking and correction, such as the enchant module, which interfaces with various spell-checking libraries. These functionalities can help identify and correct misspelt words. The formal way of writing spell checking is as follows:
  $\mathbf{C(T)} = \{\mathbf{C}(t_i) \mid t_i \in \mathbf{T}\}$
  where $T = t_1, t_2, ..., t_n$ are tokenized input text and $C(t_i)$ is a function that returns the corrected word for token t_i

Starting with the Initial Raw Dataset (IRD), approximately 1.6 million social media posts were collected from multiple publicly available sources covering a diverse range of events, including natural disasters, sports events, and health-related crises. To ensure data quality and consistency across heterogeneous sources, a comprehensive preprocessing pipeline was applied. During this stage, duplicate, noisy, incomplete, non-English, and irrelevant records were identified and removed, resulting in a cleaner and more reliable corpus for subsequent analysis.

To provide greater transparency regarding the preprocessing workflow, Algorithm 2 and Figure 3 illustrate the Natural Language Toolkit (NLTK)-based preprocessing pipeline adopted in this study. The algorithm processes raw event-related social media posts and produces a normalized textual corpus suitable for machine learning and deep learning applications.

The preprocessing process begins by initializing the required NLP resources, including a tokenizer, stop-word dictionary, punctuation set, lemmatizer, and spell-checking module. Each textual instance then undergoes a sequence of preprocessing operations. First, the text is tokenized into individual words and converted to lowercase to ensure consistency throughout the corpus. Subsequently, stop words and punctuation symbols are removed to eliminate non-informative content. Lemmatization is then applied to transform words into their canonical forms, thereby reducing vocabulary sparsity and improving semantic consistency. A spell-correction stage further enhances textual quality by correcting typographical and spelling errors. Finally, the cleaned tokens are reconstructed into normalized textual instances and stored in the processed dataset.

Although the preprocessing stage improved data quality, the resulting dataset remained imbalanced, with considerable variation in the number of instances across event classes. Such class imbalance can negatively impact the learning process by biasing classification models toward majority classes while reducing their ability to accurately identify minority classes. To mitigate this challenge, a Large Language Model (LLM)-based data augmentation framework was employed using ChatGPT (GPT-6).

Algorithm 2 Data preprocessing with the **NLTK** toolkit

```
BEGIN
Input: Initial Main Dataset (IMD)
Output: Processed Tweets (Processed_Tweets)
Required Packages:
    lemmatizer ← WordNetLemmatizer()
    spell_checker ← SpellChecker()
    stop_words ← Set(stopwords.words("english"))
    punctuation_set ← Set(string.punctuation)
Initialize:
    processed_tweets ← []
    temp_tokens ← []
for each tweet t in IMD do
    temp_tokens ← []
    for each word i in word_tokenize(t) do
        temp_tokens.append(i)
    end for
    # Convert to lowercase
    temp_tokens_lower ← []
    for each word j in temp_tokens do
        temp_tokens_lower.append(word.lower(j))
    end for
    # Remove stop words
    tokens_no_stopwords ← []
    for each word k in temp_tokens_lower do
        if k ∉ stop_words then
            tokens_no_stopwords.append(k)
        end if
    end for
    # Remove punctuation
    tokens_no_punctuation ← []
    for each word l in tokens_no_stopwords do
        if l ∉ punctuation_set then
            tokens_no_punctuation.append(l)
        end if
    end for
    # Lemmatization
    lemmatized_tokens ← []
    for each word m in tokens_no_punctuation do
        lematized_tokens.apend(lematizer.lemmatize(m))
    end for
    # Spell Checking
    corrected_tokens ← []
    for each word n in lemmatized_tokens do
        corrected_tokens.append(spell_checker.correction(n))
    end for
    # Combine tokens into a cleaned tweet
    cleaned_tweet ← " ".join(corrected_tokens)
    # Append the cleaned tweet to the processed list
    processed_tweets.append(cleaned_tweet)
end for
Return processed_tweets
END
```

Specifically, prompt-engineering strategies were developed to generate semantically meaningful and contextually relevant synthetic samples for each event class. The prompts incorporated domain-specific terminology, event descriptions, contextual information, and lin-

guistic patterns extracted from the original datasets. Through iterative prompt refinement, ChatGPT generated diverse event-related textual instances that preserved the semantic characteristics and contextual properties of their corresponding classes while introducing additional lexical and syntactic variability.

The Generative Language Model (ChatGPT) is utilized exclusively as an offline data augmentation mechanism during the dataset construction and preparation stage. Specifically, it is employed to synthesize additional event-specific textual instances to mitigate class imbalance and enhance the lexical and contextual diversity of the training corpus. The generated samples subsequently undergo the same preprocessing and quality-control procedures applied to the original data, including duplicate detection, text normalization, token standardization, tokenization, stop-word removal, punctuation handling, lemmatization, and linguistic validation. These procedures ensure consistency between the authentic and synthetic portions of the dataset while minimizing the introduction of artificial noise. The validated synthetic instances are incorporated only into the training partition, thereby preventing potential information leakage into the validation and test sets. It should be noted that the Generative Language Model does not form part of the proposed event detection architecture and is not involved in feature extraction, feature fusion, classification, or inference. Following the offline data augmentation stage, the proposed detection architecture operates independently of the Generative Language Model, with the augmented training corpus subsequently processed through the proposed text representation and CNN–LSTM-based event detection framework.

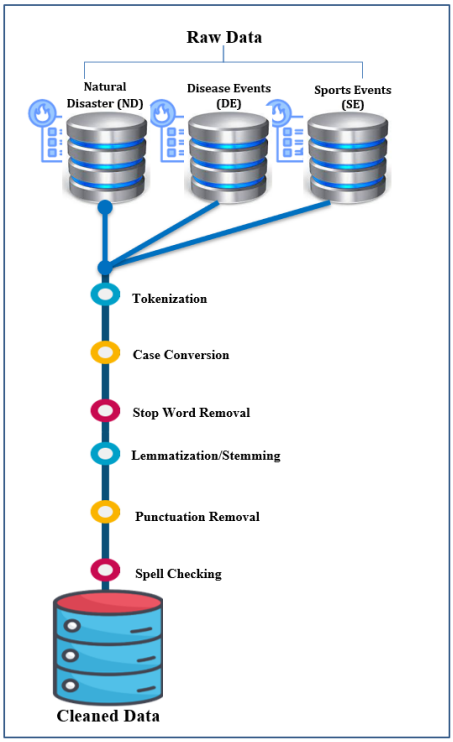


Figure 3. Converting raw data into cleaned data using the NLTK toolkit.

As shown in Table 2, approximately 1.52 million AI-generated samples were produced and integrated with the original 1.6 million instances. The augmentation process was performed across all event classes, with the number of generated samples being approximately proportional to the original class sizes. Consequently, the final balanced Final Experimental Dataset (FED) comprised approximately 3.12 million instances spanning 14 event categories. This augmentation strategy not only increased the volume of training data but also enhanced

intra-class diversity, improved the statistical representativeness of individual event classes, and reduced the adverse effects of class imbalance.

Table 2. Balanced Final Experimental Dataset (FED) after AI-generated data augmentation.

Sample/Instance Type	Original Samples	AI-Generated Samples	Total Samples
Earthquake Instances	27,930	25,500	53,430
Landslide Instances	9858	9000	18,858
Hurricane Instances	575,161	550,000	1,125,161
Flood Instances	75,249	70,000	145,249
Badminton Instances	25,125	30,000	55,125
Tennis Instances	24,170	15,000	39,170
Cricket Instances	24,645	22,000	46,645
FIFA Instances	493,011	470,000	963,011
Acne Instances	79,520	80,000	159,520
COVID-19 Instances	42,544	40,000	82,544
Depression Instances	6572	6000	12,572
Eczema Instances	73,935	80,000	153,935
Ringworm Instances	57,335	50,000	107,335
Vitiligo Instances	84,950	70,000	154,950
Total	1,600,005	1,517,500	3,117,505

The resulting FED provides a large-scale and diverse benchmark for independent event classification. By combining authentic social media posts with carefully validated AI-generated samples, the dataset enables the development of more robust, generalizable, and bias-resistant machine learning and deep learning models for event detection and classification. It is to be noted that, in this study, the term "balanced dataset" refers specifically to the balance between event-related and non-event samples within each individual event-classification experiment, rather than to an equal number of samples across all 14 event categories. Each event category was treated as an independent binary classification task under the same experimental configuration. For example, the COVID-19 experiment distinguishes COVID-19 from non-COVID-19 samples, while the Flood and Hurricane experiments distinguish their respective event-related samples from non-event samples. Therefore, the variation in total sample sizes across categories reflects the availability of category-specific data and the applied AI-based data augmentation rather than class imbalance within an individual experiment. The AI-generated samples also contributed to within-category balancing by providing additional event and non-event instances, including non-event samples containing event-related keywords used in different contexts, thereby increasing contextual diversity and reducing reliance on simple keyword-based associations.

As we can observe in Table 2, event datasets vary considerably in size due to differences in the availability of social media data for each event. For example, the Hurricane dataset contains substantially more samples than the Depression dataset. However, each of the 14 event datasets was processed and evaluated independently. Within each dataset, event-related and non-event samples were balanced before training. Thus, balancing refers to the class distribution within each individual dataset, not to equal sample sizes across all event categories.

It can now be concluded that the initial data collection comprised approximately 9.61 million raw samples across the 14 event categories as presented in Table 1. These data

were subsequently subjected to preprocessing and quality-control procedures, including text normalization, duplicate removal, and filtering of irrelevant, redundant, and low-quality records. Following these procedures, 1,600,005 authentic samples were retained for the final experiments, as reported in Table 2. These authentic samples were subsequently augmented with 1,517,500 AI-generated samples, resulting in a final experimental dataset of 3,117,505 samples. Thus, the difference between the raw and final sample counts reflects the data-cleaning and filtering process.

It is further noted that URLs, hashtags, user mentions, and emojis were removed during preprocessing primarily to reduce non-linguistic noise, sparsity, and source- or interaction-specific artifacts and to obtain a consistent textual representation across the heterogeneous datasets. While URLs and user mentions may primarily represent external links or interaction patterns rather than the underlying event semantics, hashtags and emojis may contain valuable semantic, contextual, or affective information. Their removal therefore represents a trade-off between noise reduction and information preservation and may potentially result in the loss of useful social-media-specific cues. In the present study, the primary focus was on evaluating the overall performance of the proposed Event Detection framework and its integrated deep learning models rather than isolating the individual contribution of each preprocessing element. Accordingly, the potential impact of retaining these features is acknowledged and identified as an important direction for future research through controlled comparative and ablation experiments.

4. Feature Engineering

In feature engineering, we aim to identify the most salient and informative characteristics from raw data. Its primary objective is to improve the effectiveness of deep learning models [50]. This is achieved by transforming raw data into more meaningful features that can better represent the underlying patterns and relationships. Careful selection and construction of features can greatly impact the performance of learning models and is often a vital stage in machine/deep learning.

To identify and interpret the patterns from the textual content, we employed two basic word embedding techniques, viz. TF-IDF [51], and Word2Vec [52], to represent collected data to the deep learning models. These two embedding methods are briefly discussed in the following subsections.

4.1. Word2Vec-Based Feature Representations

The Word2Vec model is employed to extract semantic relationships between words in corpora, enabling the detection of contextual and relational patterns critical for social event detection. It leverages Neural Network (NN) architectures to generate word embeddings through a three-layer network comprising an input layer, a hidden layer, and an output layer. Given a text corpus as input, Word2Vec outputs high-dimensional vectors where semantically similar words are mapped to the same region in the embedding space.

The model operates by performing sophisticated mathematical transformations on word tokens, capturing latent relationships such as similarity and analogy. This is particularly advantageous for social event detection, as it enables the identification of contextual clues and implicit connections between event-related keywords. Word2Vec supports two primary training algorithms: Continuous Bag of Words (CBoW) and Skip-Gram [53]. The former uses the context to predict a target word (w) and later uses a word (w) to predict the target context, i.e., neighbors. In other words, CBoW predicts a word based on its surrounding context, and Skip-Gram predicts the context given a specific word. These techniques allow the model to generalize word usage patterns effectively, facilitating ro-

bust event detection in large-scale textual datasets. In mathematical terms, we can assert the following:

$$\text{if } w_{i-1},\ w_{i-2}, w_{i+1}, w_{i+2} \text{ are given words} \\ -\text{the Word2Vec CBoW will provide } w_i \ (A)$$

Similarly, the word2Vec skip-gram model can be mathematically stated as follows:

$$\text{if } w_i \text{ is given word the Word2Vec Skip} - \text{Gram} \\ -\text{will provide } w_{i-1},\ w_{i-2}, w_{i+1}, w_{i+2} \ (B)$$

The output of the Word2vec model is a set of word vectors where vectors close to each other in the vector space have a similar meaning, and the vectors away from each other have different meanings. For example, in the context of a flood-related event, words like *evacuation* and *rescue* will be positioned close to each other in the vector space, reflecting their semantic similarity, while words such as *rainfall* and *vaccine* will appear farther apart, as they belong to different contexts.

Figure 4 presents a visualization of the Word2Vec model, showcasing its architecture with the input layer, hidden layer, and output layer, as well as the embedding space where semantically similar words are grouped closely. As shown in the figure, there are two matrices shown, embedding (m) and vector (n) matrices. The embedding matrix stores the vector representations (embeddings) of all words in the vocabulary, where each row corresponds to a word and its learned representation. These embeddings capture the semantic relationships between words in a continuous vector space.

The context matrix is used during training to predict context words (surrounding words) for a given target word, particularly in the Skip-Gram approach. It stores the embeddings of context words and is optimized to reflect word associations based on their proximity in the text.

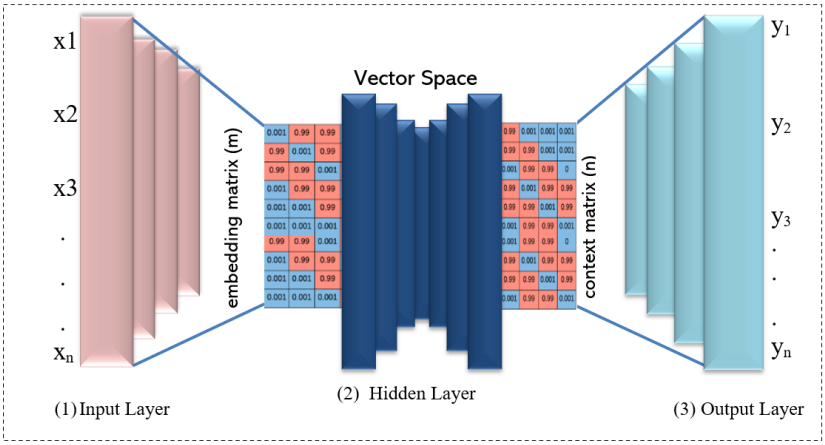


Figure 4. Word2vec embedding architectural concept [18].

4.2. Term Frequency—Inverse Document Frequency (TF-IDF)-Based Feature Representations

TF-IDF embedding has also been utilized to derive the distinguishing patterns between COVID-19 and flood-related instances. TF-IDF is a statistical measure to estimate how relevant a word is to a document in a collection of documents. It is achieved by performing the product operation on two matrices: how many times a particular word (w) is present in a document (TF) and the number of documents ($|D|$) in which the word appears (IDF).

In formal terms, we can state the following:

Let us say T_f represents the number of repetitions of a given word w in a document d divided by the total number of unique words in the document nd. Similarly, ID_f represents

the number of total documents D divided by the number of documents containing word (nd). More formally, it can be represented as follows:

$$\mathrm{T_f} = \frac{\mathrm{w(d)}}{\mathrm{nd}} \; and \; \mathrm{ID_f} = \frac{\mathrm{D}}{\mathrm{nd} \in \mathrm{D}} \tag{1}$$

To sum up the TF-IDF model more formally and understandably, the TF-IDF value of a word w in a document d from a set of documents D can be calculated as follows:

$$T_f - ID_f(w, d, D) = T_f(w, d) \times ID_f(w, D)$$

where

$$T_f(w, d) = log(1 + frequency(w, d))$$

and

$$ID_f(w, D) := log(\frac{|D|}{|d \in D : w \in d|}) \tag{2}$$

In the context of social event detection, we applied the Term Frequency-Inverse Document Frequency (TF-IDF) technique to each tweet to identify and weight words most relevant to specific events. TF-IDF assigns higher importance to words that appear frequently in a particular document (tweet) but are relatively rare across the corpus, making them indicative of the tweet's unique context. For instance, in flood-related tweets, terms like evacuate, floodwater, and damage often emerge as high-scoring tokens, while in COVID-19-related tweets, terms such as quarantine, vaccine, and pandemic frequently stand out.

By leveraging techniques like TF-IDF, key terms associated with a particular social event can be identified, enabling clear distinctions between different event types. Therefore, techniques like TF-IDF can learn keywords used in one class to better distinguish it from other classes.

Figure 5 presents a simple diagram illustrating the TF-IDF architecture. It shows the flow from the document corpus through Term Frequency (TF) and Inverse Document Frequency (IDF) calculations to the final TF-IDF matrix. As shown in the figure, the TF-IDF architecture evaluates word importance in a document relative to a corpus. It calculates Term Frequency (TF) to measure how often a word appears in a document and Inverse Document Frequency (IDF) to reduce the weight of common words across documents. Multiplying TF and IDF produces a TF-IDF matrix, where rows represent documents, columns represent words, and values indicate word significance.

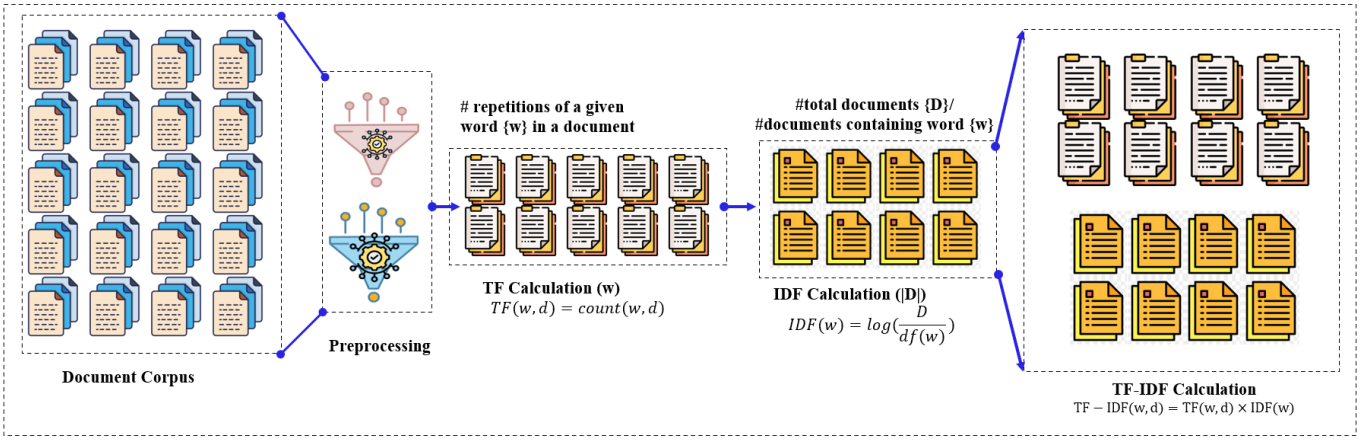


Figure 5. Term Frequency-Inverse Document Frequency (TF-IDF) embedding architectural concept.

5. Methodology and Experimental Configuration

The experimental configuration of the proposed social event detection framework follows a systematic pipeline encompassing data preprocessing, manual annotation and

filtering, dataset partitioning, text representation, and deep-learning-based event classification. The overall architecture of the proposed framework is illustrated in Figure 6.

Initially, the collected raw and unstructured social media data are preprocessed using the Natural Language Toolkit (NLTK) with Python 3.4 [49]. As mentioned before, the preprocessing pipeline comprises several essential operations, including tokenization, case normalization, stop-word and punctuation removal, lemmatization, and spell checking. These operations convert the raw textual content into a standardized and machine-readable form, thereby making it more suitable for subsequent machine learning and deep learning analysis. Following preprocessing, the resulting data undergo a rigorous manual annotation and filtering process. Three trained human annotators independently examine and label the samples to ensure accurate and consistent class assignment. Ambiguous, irrelevant, and potentially misleading instances are identified and removed during this stage. This multi-annotator procedure improves the quality and reliability of the dataset while reducing annotation-related noise that could adversely affect the performance of the classification models. The resulting curated dataset constitutes the Final Experimental Dataset (FED), as presented in Table 2. Before model development, the FED is systematically partitioned into training, validation, and test subsets while maintaining data integrity and minimizing the risk of data leakage. It is further noted here that the validation and test subsets were maintained exclusively from authentic data. Exact and near-duplicate tweets are first removed, after which a source-aware and user-aware splitting strategy is employed. Specifically, tweets originating from the same user or dataset source are restricted to a single partition, thereby reducing the possibility that highly similar or related instances occur across different subsets. The resulting dataset is divided into approximately 70% for training, 15% for validation, and 15% for testing.

We divided the 1,600,005 authentic samples into training, validation, and test sets at approximately a 70:15:15 ratio, corresponding to 1,120,004, 240,000, and 240,001 samples, respectively. We then added the 1,517,500 AI-generated samples exclusively to the training set to address data scarcity and class imbalance, resulting in approximately 2,637,504 training samples. No AI-generated samples were included in the validation or test sets, ensuring that model performance was evaluated on unseen authentic social media data. It is important to note that the study does not implement a unified 14-class multiclass classifier. Instead, the 14 event categories are evaluated through 14 independent binary event detection tasks, with each category treated as a separate binary classification problem. The overall dataset partitioning described above applies to the experimental data, while the classification task is defined and evaluated independently for each event category.

As illustrated in Figure 2, the collected authentic social media data are first partitioned into training, validation, and testing subsets. AI-generated samples are subsequently incorporated exclusively into the training subset for data augmentation. The validation and testing subsets remain strictly limited to authentic social media samples and are therefore excluded from the augmentation process. This data-construction strategy prevents potential synthetic-data leakage and ensures that model performance is evaluated on authentic, unseen social media content.

The partitioned textual data are subsequently transformed into numerical representations using two complementary text representation techniques: Term Frequency–Inverse Document Frequency (TF-IDF) and Word2Vec. TF-IDF captures the relative importance of individual terms within the corpus, whereas Word2Vec provides distributed representations that capture semantic and contextual relationships among words. These two representations provide complementary views of the textual information and are used as inputs to the selected deep learning architectures. The use of both representations enables

a systematic investigation of the influence of different feature representations on event classification performance.

As illustrated in Figure 6, the proposed Event Detection framework is evaluated across three major event domains: Natural Disasters, Sports Events, and Disease Events. The Natural Disaster domain comprises four event categories: Earthquake, Flood, Landslide, and Hurricane. The Sports Event domain includes Badminton, Cricket, FIFA, and Tennis, whereas the Disease Event domain consists of six categories: Acne, Depression, Eczema, Ringworm, Vitiligo, and COVID-19. Thus, the experimental dataset encompasses a total of 14 independent event-specific categories.

To evaluate the effectiveness of different deep learning architectures across these event categories, five models were selected, namely, CNN, RNN, LSTM, BiLSTM, and BiRNN. Each model is independently trained and evaluated for every event category. Consequently, the 4 Natural Disaster categories result in 20 experiments (4 categories × 5 models), the 4 Sports Event categories result in 20 experiments (4 categories × 5 models), and the 6 Disease Event categories result in 30 experiments (6 categories × 5 models).

Overall, the proposed framework involves 70 individual experiments (14 independent event categories × 5 models). This comprehensive experimental design enables systematic comparison of the selected deep learning architectures across heterogeneous event domains and alternative text representations.

Given the extensive scope of the experimental evaluation, the results are presented and analyzed in a structured manner. The outcomes of all experimental configurations are considered in the overall evaluation; however, presenting the complete training and validation process for all 70 experiments individually would result in substantial repetition and an unnecessarily lengthy presentation. Therefore, the subsequent analysis focuses on the most informative and best-performing models. The detailed training and validation behavior of the selected best-performing models is presented in Section 6.1, whereas the final performance obtained on the independent test dataset is reported and discussed in Section 6.3.

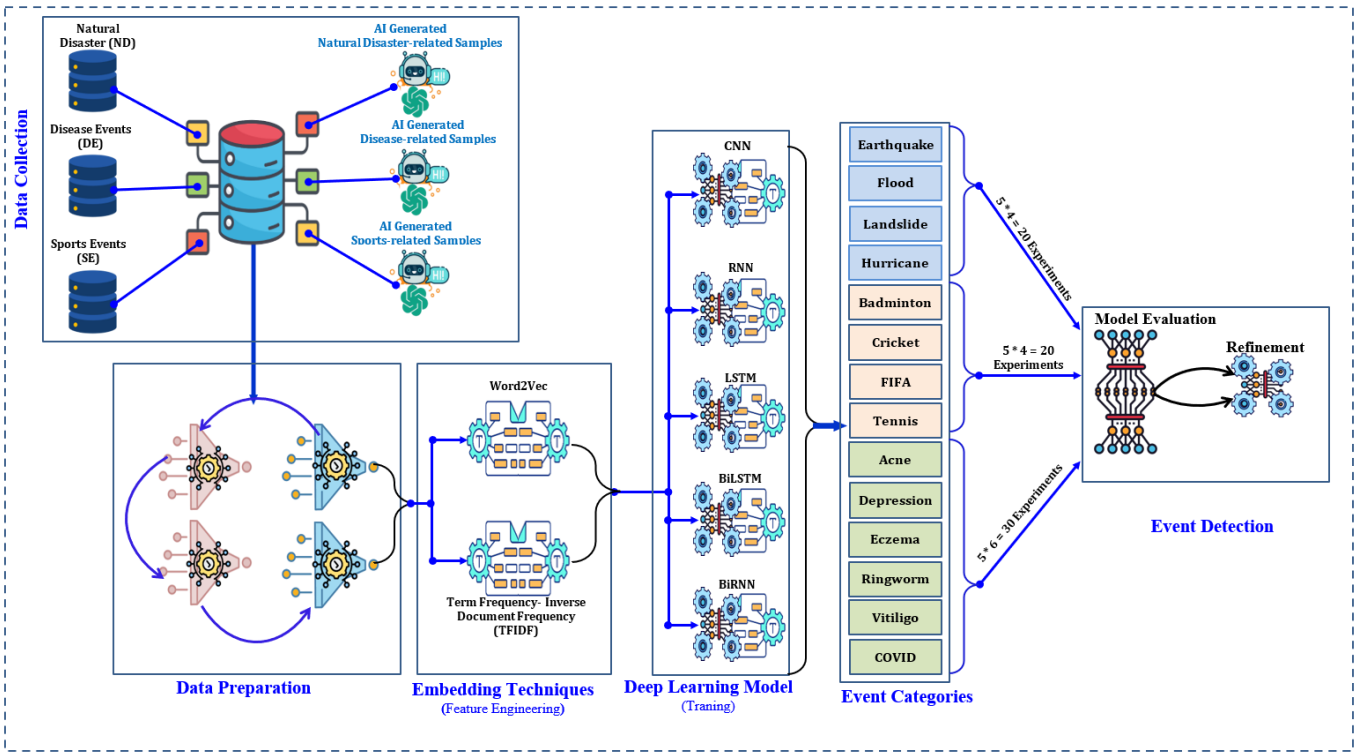


Figure 6. Architecture of the proposed social event detection framework.

AI-generated samples were used exclusively to augment the training set to address data scarcity and class imbalance. The validation and test sets contained only authentic data, ensuring that the reported performance reflects the model's ability to classify unseen authentic social media posts rather than recognize synthetic samples.

We evaluate model performance using accuracy, precision, recall, F1-score, and the Matthews Correlation Coefficient (MCC), based on the confusion matrix in Table 3. The formal representation for each of these metrics is presented as follows.

Table 3. Confusion matrix.

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positive (**TP**)	False Positive (**FP**)
	Negative	False Negative (**FN**)	True Negative (**TN**)

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{3}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{5}$$

To analyze the balanced performance using both precision and recall, we have used the F-score. The F-measure, or F-score, estimates the performance of the model on a given dataset by combining the precision and recall of the model and is defined as the harmonic mean of the model's precision and recall. The formula for calculating the F-measure is as follows.

$$F_\beta - score = (1 + \beta^2) \times \frac{\text{PR}}{\beta^2\text{P} + \text{R}} \tag{6}$$

The F_β-score is a generalization of the F-score which adds a configuration parameter called beta (β). The default value is 1.0, which is the same as the F-measure referred to as the F1-score. A smaller beta value (such as = 0.5) gives more weight to precision and less to recall, whereas a larger beta value (such as = 2.0) favors recall more than precision. It is helpful to use both precision and recall, but slightly more attention is needed on one or the other, such as when false negatives are more important than false positives or vice versa. The better way to calculate the F1-score for our study is mentioned in Equation (7).

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{7}$$

The Matthews Correlation Coefficient (MCC) in statistics, known as the phi coefficient, is a measure of association between two binary variables, while in machine learning, it is used as a measure of the quality of binary classifications. It is a more reliable statistical metric that produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (TP, FN, TN, and FP). MCC is preferred over the F1 score as it is a more balanced evaluation of classification models, no matter which class is positive. The mathematical way to calculate MCC is given in Equation (8).

$$\text{MCC} = \frac{\text{TN} \times \text{TP} - \text{FP} \times \text{FN}}{\sqrt{(\text{TN} + \text{FN})(\text{FP} + \text{TP})(\text{TN} + \text{FP})(\text{FN} + \text{TP})}} \tag{8}$$

It is to be noted that the proposed framework is evaluated across 14 independent binary event detection tasks spanning natural disasters, disease events, and sports events.

Design of Integrated CNN and LSTM Architecture for Event Detection

So far, we have observed that extracting meaningful event-related information from social media posts is challenging because such content is often noisy, informal, unstructured, and highly variable in linguistic form. Therefore, an effective event detection model needs to capture both discriminative local textual patterns and contextual dependencies within social media posts.

To address these requirements, an integrated Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architecture is employed. CNNs are effective in extracting local and discriminative features from textual sequences by identifying informative n-gram patterns, whereas LSTM networks are designed to capture sequential dependencies and contextual relationships across the input sequence. The integration of CNN and LSTM therefore enables the model to first identify salient local event-related patterns and subsequently learn their sequential and contextual relationships.

Figure 7 shows the design of the proposed integrated CNN and LSTM architecture for the detection of social media events. The proposed architecture consists of an Input Layer, Hybrid Embedding Layer, parallel one-dimensional Convolutional (Conv1D) Layers, feature concatenation, Batch Normalization, Max Pooling, Dropout, an LSTM Layer, a Flatten Layer, a Fully Connected (Dense) Layer, and an Output Layer.

The architecture begins with an Input Layer of shape (100, 300), where each social media post is represented as a sequence of up to 100 tokens, with each token associated with a 300-dimensional representation. Posts shorter than the predefined sequence length are padded, whereas longer posts are truncated to maintain a fixed input structure. This standardization allows the model to process social media posts of varying lengths efficiently while maintaining a consistent representation across all samples. Following the input stage, the posts are processed through a Hybrid Embedding Layer that combines Word2Vec and TF-IDF representations. Word2Vec provides dense semantic representations by learning relationships between words based on their contextual usage. Consequently, semantically related terms can obtain similar vector representations even when their exact lexical forms differ. In contrast, TF-IDF emphasizes terms according to their discriminative importance within the textual corpus, assigning greater importance to terms that are relatively informative for distinguishing documents or event categories. The integration of Word2Vec and TF-IDF is intended to combine semantic information with lexical term importance, providing a richer representation of social media content. The hybrid textual representation is subsequently passed through three parallel 1D Convolutional Layers with kernel sizes of 3, 5, and 7. These convolutional layers operate as local feature detectors and learn discriminative n-gram patterns from the input sequence. The kernel with a size of 3 captures local three-token patterns, while the kernels with sizes of 4 and 5 capture broader four-token and five-token patterns, respectively. The use of multiple kernel sizes enables the model to identify event-related expressions at different granularities. The feature maps generated by the three convolutional branches are then concatenated to form a unified feature representation. This feature fusion combines the information extracted using different kernel sizes and allows the subsequent layers to utilize complementary multi-scale features. The concatenated feature maps therefore provide a broader representation of local patterns than would be obtained from a single convolutional kernel size.

The merged representation is then processed through a sequence of Batch Normalization, Max Pooling, and Dropout operations. Batch Normalization normalizes the intermediate activations and helps stabilize the training process, potentially improving convergence. Max Pooling reduces the dimensionality of the convolutional feature maps while retaining the most prominent features within local regions. This operation reduces computational requirements and allows the network to focus on salient event-related patterns. Dropout is

subsequently used as a regularization mechanism by randomly deactivating a proportion of neurons during training.

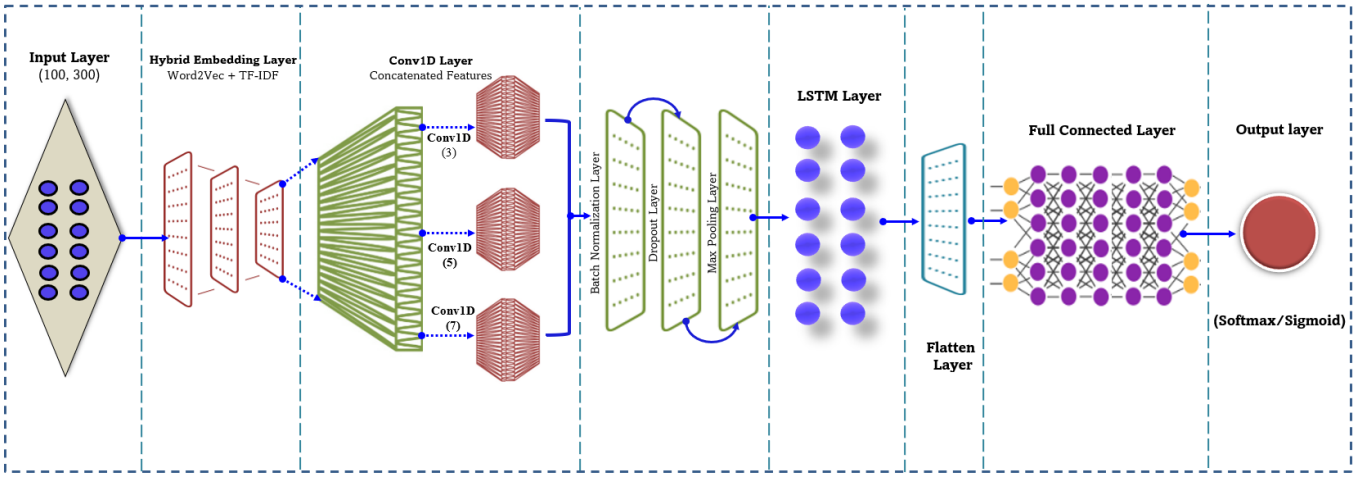


Figure 7. Proposed integrated CNN and LSTM architecture for event detection.

The resulting pooled feature sequence is then fed into a Long Short-Term Memory (LSTM) layer with 128 hidden units. The CNN component is primarily responsible for extracting local and position-invariant textual patterns, while the LSTM captures sequential dependencies and contextual relationships among the extracted features.

The output of the LSTM layer is subsequently transformed into a one-dimensional representation using a Flatten Layer, which prepares the learned sequential features for further processing by the Fully Connected (Dense) Layer and, ultimately, the Output Layer for event classification.

The flattened representation is subsequently passed to a Dense Layer with 128 units. This layer learns higher-level nonlinear combinations of the features extracted by the CNN and LSTM components. The 128-unit Dense Layer is followed by Batch Normalization and Dropout. Batch Normalization helps maintain stable activation distributions during training, while Dropout provides additional regularization and reduces the likelihood of overfitting. A second Dense Layer with 64 units further transforms the representation into a more compact feature space and refines the learned information before the final classification stage.

Finally, the processed representation is passed to the Output Layer, which generates the predicted event category. For binary event classification, a Sigmoid activation function can be used to produce the probability of the target class. For multiclass event detection, a Softmax activation function is employed to generate a probability distribution across the event categories. The class associated with the highest predicted probability is selected as the final event classification.

The proposed CNN–LSTM architecture is evaluated across diverse social media event domains, including Natural Disasters such as Earthquake, Flood, Landslide, and Hurricane, Sports Events such as Badminton, Cricket, FIFA, and Tennis, and Disease Events such as Acne, Depression, Eczema, Ringworm, Vitiligo, and COVID-19. The architecture is designed to exploit the complementary strengths of hybrid lexical-semantic representation, multi-scale convolutional feature extraction, and bidirectional sequential modeling. The Word2Vec component contributes semantic relationships, the TF-IDF component emphasizes discriminative terms, the CNN component extracts local event-specific patterns, and the LSTM component captures contextual dependencies among these features. The integration of these components provides a comprehensive representation of noisy social media posts and supports robust classification across heterogeneous event categories.

To improve the reproducibility of the proposed CNN-LSTM model, the principal architectural and training hyperparameters are summarized in Table 4. As mentioned before the model receives a 100-token sequence represented using the Word2Vec and TF-IDF-based hybrid representation with a feature dimension of 300. Three parallel one-dimensional convolutional branches with kernel sizes of 3, 5, and 7 are employed to capture local textual patterns at different contextual scales. The extracted features are normalized, regularized using dropout, and downsampled through max pooling before being concatenated and passed to a single LSTM layer with 128 units. The resulting sequential representation is subsequently processed by a fully connected layer and a sigmoid-based output layer. The table also specifies the optimization and training parameters, including the optimizer, learning rate, batch size, loss function, number of epochs, early-stopping criterion, and random seed.

Table 4. Architecture and training hyperparameters of the proposed CNN-LSTM model.

Parameter	Configuration
Model architecture	CNN-LSTM
Input sequence length	100 tokens
Feature representation	Word2Vec + TF-IDF
Input feature dimension	300
Number of CNN branches	3
Filters per CNN branch	128
CNN kernel sizes	3, 5, and 7
Convolution stride	1
Convolution padding	Same
CNN activation function	ReLU
Batch normalization	Applied after convolution
CNN dropout rate	0.30
Max-pooling size	2
Number of LSTM layers	1
LSTM units	128
LSTM return sequences	False
Fully connected layer	64 units
Dense-layer activation	ReLU
Output activation	Sigmoid
Loss function	Binary cross-entropy
Optimizer	Adam
Learning rate	0.001
Batch size	64
Maximum number of epochs	10
Early-stopping patience	5 epochs
Random seed	42

6. Results Analysis and Discussion

6.1. Individual Model Training and Validation

All models were trained for ten epochs (for convenience, we present five epochs in the table), and we meticulously recorded their training times. We halted training for each model

after reaching an epoch patience of 5. Table 5 provides a summary of the performance metrics for different neural network architectures in the context of fusion-driven social event detection. Each model is evaluated based on several key metrics, including the epoch time (in seconds), loss, accuracy, validation loss, and validation accuracy. A pictorial representation of all the selected models for model accuracy and model loss is presented in Figures 8–12.

Among all models, CNN demonstrates the fastest training time per step (14 ms) and rapid convergence, reaching a validation accuracy of 91.74% by the third epoch with consistently low validation loss (0.019). This suggests that CNN effectively captures relevant features early and offers an excellent trade-off between efficiency and performance, making it suitable for real-time or resource-constrained environments.

RNN, on the other hand, shows slower convergence and moderately lower performance. While it improves over epochs, its validation accuracy plateaus around 89.1% and loss remains higher (0.0307), indicating limitations in handling long-term dependencies. This is expected given RNNs' known vanishing gradient issues, especially when compared to gated architectures like LSTM.

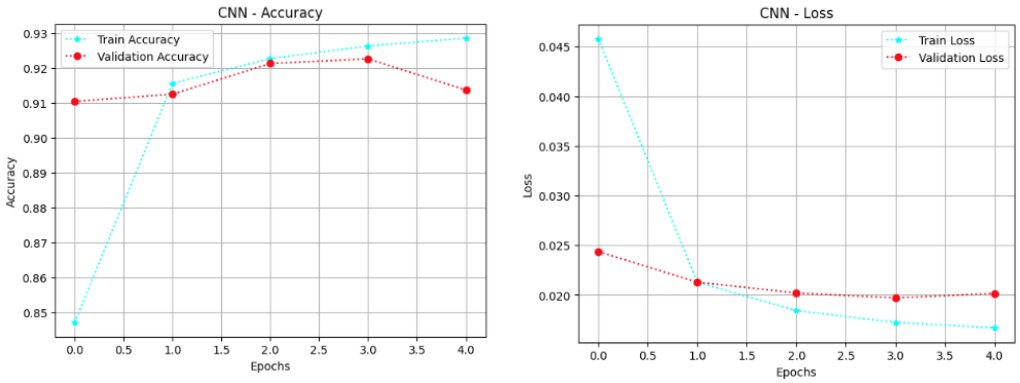
LSTM reported enhanced performance over RNN, achieving 91.4% validation accuracy and reduced validation loss (0.0195) by the fifth epoch. Despite a slower training speed (64–66 ms/step), LSTM provides a solid balance between sequence modeling capability and generalization. It clearly benefits from its gating mechanisms, which better preserve context across time steps.

Table 5. Training and validation performance per epoch.

Model	Ep.	Time	Train Acc	Train Loss	Val Acc	Val Loss
CNN	1	14 ms (49 s)	0.7242	0.0869	0.9064	0.0227
	2	14 ms (46 s)	0.9144	0.0204	0.9135	0.0196
	3	14 ms (47 s)	0.9236	0.0175	0.9174	0.0188
	4	14 ms (47 s)	0.9290	0.0160	0.9160	0.0190
	5	14 ms (47 s)	0.9283	0.0155	0.9152	0.0191
RNN	1	25 ms (88 s)	0.6605	0.1122	0.8767	0.0341
	2	24 ms (81 s)	0.8791	0.0355	0.8899	0.0319
	3	24 ms (82 s)	0.8952	0.0297	0.8988	0.0287
	4	24 ms (81 s)	0.9063	0.0245	0.9009	0.0284
	5	24 ms (80 s)	0.9103	0.0240	0.8912	0.0307
LSTM	1	64 ms (218 s)	0.6868	0.1052	0.8889	0.0287
	2	64 ms (217 s)	0.9069	0.0239	0.9085	0.0224
	3	64 ms (216 s)	0.9189	0.0200	0.9126	0.0210
	4	66 ms (222 s)	0.9241	0.0183	0.9173	0.0200
	5	66 ms (222 s)	0.9273	0.0174	0.9142	0.0195
BiLSTM	1	76 ms (262 s)	0.7143	0.0954	0.9001	0.0254
	2	80 ms (270 s)	0.9063	0.0242	0.9110	0.0221
	3	81 ms (274 s)	0.9182	0.0202	0.9187	0.0212
	4	92 ms (312 s)	0.9221	0.0185	0.9177	0.0205
	5	80 ms (269 s)	0.9249	0.0176	0.9212	0.0195
BiRNN	1	39 ms (134 s)	0.7070	0.0927	0.8890	0.0291
	2	38 ms (129 s)	0.8844	0.0312	0.8898	0.0297
	3	38 ms (129 s)	0.9001	0.0261	0.8997	0.0262
	4	39 ms (131 s)	0.9012	0.0253	0.9004	0.0233
	5	38 ms (130 s)	0.9107	0.0223	0.9097	0.0227

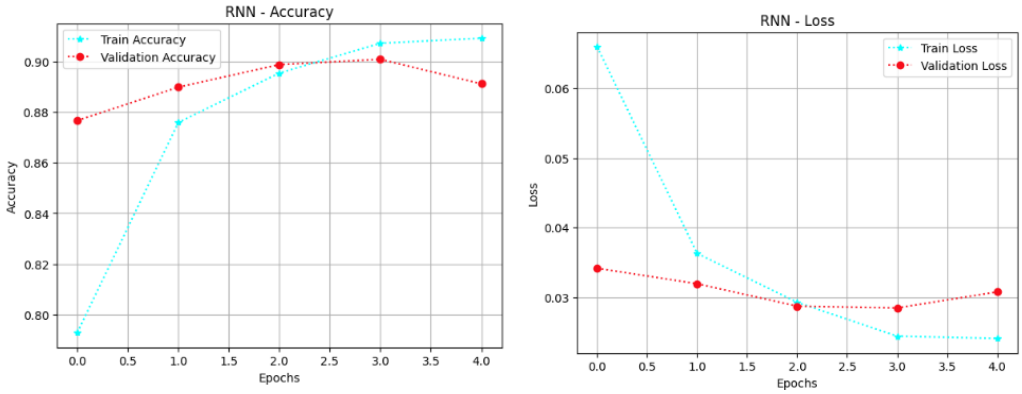
BiLSTM stands out as the most accurate model, attaining a validation accuracy of 92.12% and the lowest loss (0.0195). The bidirectional structure helps the model utilize

both past and future context, enhancing its understanding of complex sequential dependencies. However, this performance comes at a computational cost, with the highest training time (up to 92 ms/step), making it more suitable for scenarios where accuracy outweighs latency concerns.



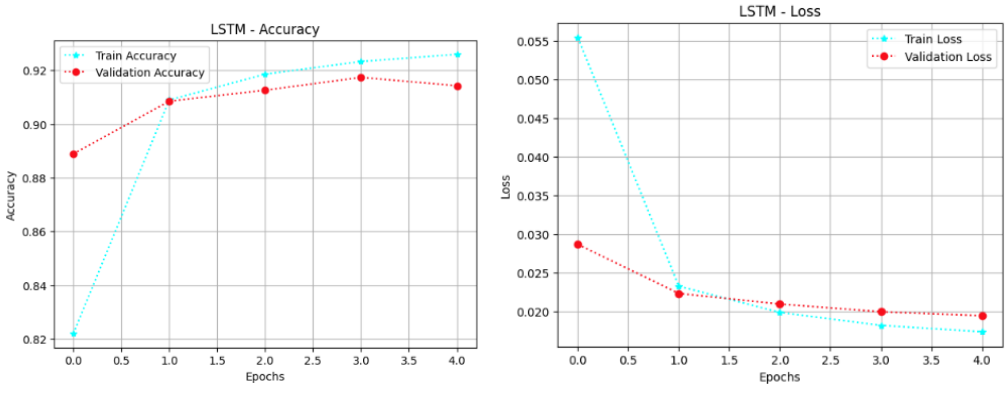
(**a**) (**b**)

Figure 8. Performance of the CNN model in terms of accuracy and loss on training and validation data. (**a**) CNN model accuracy. (**b**) CNN model loss.



(**a**) (**b**)

Figure 9. Performance of the RNN model in terms of accuracy and loss on training and validation data. (**a**) RNN model accuracy. (**b**) RNN model loss.

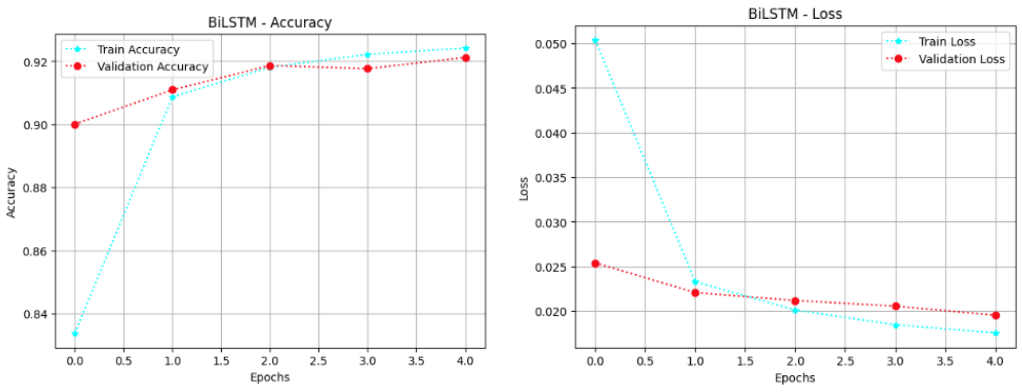


(**a**) (**b**)

Figure 10. Performance of the LSTM model in terms of accuracy and loss on training and validation data. (**a**) LSTM model accuracy. (**b**) LSTM model loss.

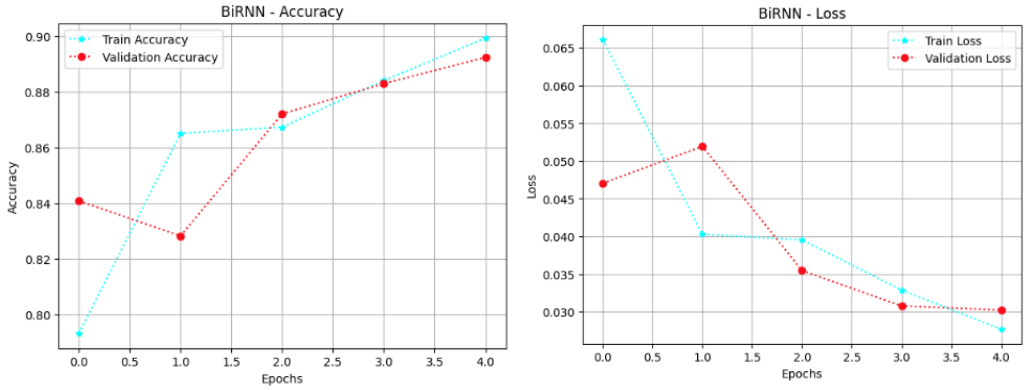
Lastly, BiRNN improves upon RNN and performs comparably to LSTM, achieving 90.97% validation accuracy and a loss of 0.0227. However, it still lags behind BiLSTM and

CNN, both in accuracy and training efficiency. Overall, BiLSTM emerges as the most effective model in terms of predictive performance, while CNN offers the best efficiency, making the choice model-dependent based on task requirements and computational constraints.



(**a**) (**b**)

Figure 11. Performance of the BiLSTM model in terms of accuracy and loss on training and validation data. (**a**) BiLSTM model accuracy. (**b**) BiLSTM model loss.



(**a**) (**b**)

Figure 12. Performance of the BiRNN model in terms of accuracy and loss on training and validation data. (**a**) BiRNN model accuracy. (**b**) BiRNN model loss.

6.2. Training and Validation of Customized CNN+LSTM Model

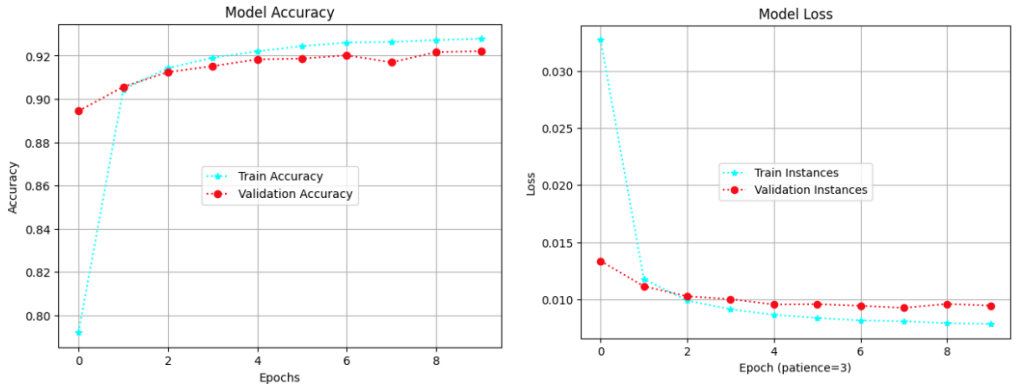
Table 6 presents the training and validation performance of the hybrid CNN+LSTM model also presented in Figure 13, which demonstrates a highly effective integration of convolutional and sequential learning paradigms, offering a balanced performance between training speed, convergence, and generalization. Compared to the standalone models, this hybrid approach consistently outperforms in terms of validation accuracy and loss minimization while maintaining moderate computational cost.

From Table 6, we observe that the validation accuracy improves from 89.44% to 92.21% over 10 epochs, with a notably low and stable validation loss (0.0095) by the final epoch. This indicates excellent generalization and learning stability. In contrast, while CNN was the fastest and had decent early convergence, it plateaued around 91.7% validation accuracy with higher loss values. LSTM and BiLSTM performed well too, but BiLSTM required more computation time (92 ms/step vs. 23 ms/step here), despite reaching a comparable validation accuracy (92.12%). Thus, CNN+LSTM achieves near state-of-the-art performance with substantially lower time costs.

Table 6. Training and validation metrics for the CNN+LSTM model.

Epoch	Time (s)	Time/Step	Train Acc.	Train Loss	Val. Acc.	Val. Loss
1/10	83	24 ms	0.6221	0.0688	0.8944	0.0134
2/10	81	24 ms	0.9012	0.0123	0.9056	0.0112
3/10	82	24 ms	0.9139	0.0100	0.9124	0.0103
4/10	79	23 ms	0.9196	0.0091	0.9152	0.0100
5/10	79	23 ms	0.9209	0.0087	0.9183	0.0096
6/10	80	23 ms	0.9242	0.0084	0.9187	0.0096
7/10	79	23 ms	0.9260	0.0081	0.9202	0.0094
8/10	79	23 ms	0.9279	0.0080	0.9170	0.0093
9/10	83	25 ms	0.9277	0.0078	0.9217	0.0096
10/10	79	23 ms	0.9284	0.0078	0.9221	0.0095

The technical justification for this improvement lies in the architectural synergy: CNN efficiently extracts spatial and local patterns from the input data, which are then passed to LSTM layers to model temporal dependencies and sequential relationships. This dual-layered representation helps the network capture both static and dynamic patterns more effectively than either model alone. Unlike RNN or BiRNN, which struggle with long-range dependencies and noisy gradients, LSTM provides stable memory control, and CNN preprocessing ensures more informative sequences are fed into it.



(**a**) (**b**)

Figure 13. Performance of the hybrid CNN+LSTM model in terms of accuracy and loss on training and validation data. (**a**) CNN+LSTM model accuracy. (**b**) CNN+LSTM model loss.

In comparison to BiLSTM, while both models excel in sequence modeling, CNN+LSTM outperforms BiLSTM in efficiency (23–25 ms/step vs. up to 92 ms/step), with nearly identical validation accuracy. This makes CNN+LSTM suitable for real-world applications requiring a trade-off between performance and latency. Moreover, the training curve for CNN+LSTM shows consistent convergence and minimal overfitting, with training and validation losses remaining close, which is a strong indicator of robustness.

The CNN-LSTM hybrid model demonstrates competitive predictive performance across the evaluated independent binary event detection tasks. Its combination of convolutional layers for extracting local textual patterns and an LSTM layer for capturing sequential dependencies provides an effective representation of the input data. Compared with the evaluated baseline architectures, the CNN-LSTM model achieves strong performance while maintaining a relatively streamlined architecture. These results indicate its suitability for binary event detection tasks involving diverse and heterogeneous social media content.

6.3. Model Testing: Results and Discussion

In this section, we present the critical phase of model testing to assess our proposed model's performance, accuracy, and reliability. As we observed in the model training phase, the ANN model performed poorly among all the selected deep learning architectures, therefore, we have decided not to take it any further for the testing phase. Tables 7 and 8 present the results of each model obtained on the test data for each class of the three categories. As shown in the table, all the models were evaluated for event detection across three categories, including Natural Disasters, Diseases, and Sports, using social media data. In the Natural Disaster category, the CNN model showed outstanding performance for Earthquake and Landslide, achieving perfect scores across all evaluation metrics such as precision, recall, F1-score, and accuracy, demonstrating the model's ability to detect these events flawlessly.

However, the model's performance dropped for Flood and Hurricane. While Flood had a high recall of 0.97, its precision was only 0.52, leading to a modest F1-score of 0.68 and an overall accuracy of 74.5%. This indicates that although the model detected most actual flood-related instances, it also produced a large number of false positives. The performance for Hurricane was the weakest among all classes, with a recall of just 0.17 and a low F1-score of 0.24, resulting in a poor accuracy of 31%. This suggests the model struggled considerably to identify hurricane-related posts, possibly due to insufficient or ambiguous data representations.

In the Disease category, the model excelled. It achieved perfect performance for Acne, Eczema, Ringworm, and Vitiligo, with all metrics scoring 1.0 and 100% accuracy. These results highlight the model's robustness in identifying and classifying content related to these diseases. Depression was also classified with near-perfect performance, yielding a precision and recall of 0.99 and an accuracy of 99%, showing that the model handled this class exceptionally well. The only slight performance dip was observed in the COVID-19 class, where the precision remained high at 0.99, but the recall dropped to 0.88, leading to an F1-score of 0.93 and an accuracy of 93.5%. This suggests that while the model was cautious in labeling posts as COVID-related, it may have missed some actual instances. In the Sports category, the model continued to perform strongly. It achieved high precision and recall for Badminton, Cricket, and Tennis, with F1-scores close to 0.98 and accuracies around 98.5% and 97.5%. These results demonstrate the model's ability to recognize sports-related content effectively. FIFA_2022 showed slightly lower recall at 0.93, resulting in an F1-score of 0.96 and accuracy of 96.5%, still reflecting robust performance overall. In summary, the CNN model performed best in detecting diseases and sports events, especially for classes like Acne, Vitiligo, Ringworm, and Tennis. While it was effective for some natural disasters like Earthquake and Landslide, it underperformed notably for Hurricane detection. This indicates a need for targeted improvements in the natural disaster category, particularly in handling underrepresented or more ambiguous classes.

The RNN model exhibited strong performance across most classes in all three categories, including Natural Disasters, Diseases, and Sports. Among disease-related classes, Acne achieved a perfect recall of 1 and an F1-score of 0.84, while Depression and Vitiligo showed near-perfect precision and recall, resulting in high F1-scores of 0.97 and 0.96, respectively. Eczema, although having high precision (0.99), had lower recall (0.7), leading to a moderate F1-score of 0.82. Ringworm stood out with an almost perfect score across all metrics.

Table 7. Class-wise performance of the CNN, RNN, and LSTM models.

Model	Class	Precision	Recall	F1-Score	F0.5-Score	F2-Score	Accuracy (%)
CNN	Acne	1.00	1.00	1.00	1.00	1.00	100
	Badminton	1.00	0.97	0.98	0.99	0.97	98.5
	COVID-19	0.99	0.88	0.93	0.96	0.89	93.5
	Cricket	0.99	0.96	0.97	0.98	0.96	97.5
	Depression	0.99	0.99	0.99	0.99	0.99	99
	Earthquake	1.00	1.00	1.00	1.00	1.00	100
	Eczema	1.00	1.00	1.00	1.00	1.00	100
	FIFA_2022	1.00	0.93	0.96	0.98	0.94	96.5
	Flood	0.52	0.97	0.68	0.56	0.83	74.5
	Hurricane	0.45	0.17	0.24	0.39	0.19	31
	Landslide	1.00	1.00	1.00	1.00	1.00	100
	Ringworm	1.00	1.00	1.00	1.00	1.00	100
	Tennis	1.00	0.97	0.98	0.99	0.97	98.5
	Vitiligo	1.00	1.00	1.00	1.00	1.00	100
RNN	Acne	0.73	1.00	0.84	0.78	0.94	86.5
	Badminton	1.00	0.96	0.98	0.99	0.96	98
	COVID-19	0.89	0.88	0.88	0.89	0.88	88.5
	Cricket	0.99	0.9	0.94	0.97	0.91	94.5
	Depression	0.98	0.95	0.97	0.98	0.96	96.5
	Earthquake	1.00	1.00	1.00	1.00	1.00	100
	Eczema	0.99	0.7	0.82	0.93	0.74	84.5
	FIFA_2022	0.99	0.92	0.96	0.98	0.93	95.5
	Flood	0.52	0.63	0.57	0.54	0.61	57.5
	Hurricane	0.49	0.46	0.47	0.48	0.46	47.5
	Landslide	1.00	1.00	1.00	1.00	1.00	100
	Ringworm	0.99	0.99	0.99	0.99	0.99	99
	Tennis	0.99	0.92	0.95	0.97	0.93	95.5
	Vitiligo	0.94	1.00	0.96	0.95	0.98	97
LSTM	Acne	1.00	1.00	1.00	1.00	1.00	100
	Badminton	0.99	0.97	0.98	0.99	0.97	98
	COVID-19	0.98	0.88	0.93	0.96	0.89	93
	Cricket	0.98	0.96	0.97	0.98	0.96	97
	Depression	0.99	0.99	0.99	0.99	0.99	99
	Earthquake	1.00	1.00	1.00	1.00	1.00	100
	Eczema	1.00	1.00	1.00	1.00	1.00	100
	FIFA_2022	1.00	0.93	0.96	0.98	0.94	96.5
	Flood	0.5	0.29	0.36	0.43	0.31	39.5
	Hurricane	0.48	0.62	0.54	0.5	0.6	55
	Landslide	1.00	1.00	1.00	1.00	1.00	100
	Ringworm	1.00	1.00	1.00	1.00	1.00	100
	Tennis	0.99	0.97	0.98	0.99	0.97	98
	Vitiligo	1.00	1.00	1.00	1.00	1.00	100

In the sports category, Badminton, Cricket, FIFA_2022, and Tennis were classified with high effectiveness. Badminton had a high precision of 1 and a recall of 0.96, leading to a

98% accuracy. Cricket and Tennis both exceeded 94% accuracy, with strong F1-scores of 0.94 and 0.95, respectively. FIFA_2022 achieved a balance of precision (0.99) and recall (0.92), resulting in 95.5% accuracy. For natural disasters, Earthquake and Landslide were perfectly classified with full scores across all metrics. Flood and Hurricane were more challenging for the model, with Flood achieving only 57.5% accuracy and an F1-score of 0.57, and Hurricane showing slightly weaker performance with 47.5% accuracy and an F1-score of 0.47.

Table 8. Class-wise performance of the BiLSTM and BiRNN models.

Model	Class	Precision	Recall	F1-Score	F0.5-Score	F2-Score	Accuracy (%)
BiLSTM	Acne	1.00	1.00	1.00	1.00	1.00	100
	Badminton	1.00	0.97	0.98	0.99	0.97	98.5
	COVID-19	0.98	0.88	0.93	0.96	0.89	93
	Cricket	0.99	0.95	0.97	0.98	0.96	97
	Depression	0.99	0.98	0.99	0.99	0.98	98.5
	Earthquake	1.00	1.00	1.00	1.00	1.00	100
	Eczema	1.00	1.00	1.00	1.00	1.00	100
	FIFA_2022	0.99	0.93	0.96	0.98	0.94	96
	Flood	0.52	0.93	0.67	0.57	0.86	72.5
	Hurricane	0.52	0.17	0.26	0.41	0.2	34.5
	Landslide	1.00	1.00	1.00	1.00	1.00	100
	Ringworm	1.00	1.00	1.00	1.00	1.00	100
	Tennis	0.99	0.96	0.98	0.99	0.97	98
	Vitiligo	1.00	1.00	1.00	1.00	1.00	100
BiRNN	Acne	0.97	0.97	0.97	0.97	0.97	94
	Badminton	0.99	0.96	0.97	0.98	0.96	95
	COVID-19	0.9	0.82	0.86	0.88	0.83	78
	Cricket	0.95	0.92	0.93	0.94	0.92	88
	Depression	0.98	0.94	0.96	0.97	0.95	92
	Earthquake	0.99	0.97	0.98	0.98	0.97	96
	Eczema	0.84	0.95	0.89	0.87	0.94	79
	FIFA_2022	1.00	0.91	0.95	0.98	0.92	91
	Flood	0.52	0.94	0.67	0.57	0.85	49
	Hurricane	0.5	0.13	0.21	0.37	0.16	13
	Landslide	1.00	1.00	1.00	1.00	1.00	100
	Ringworm	1.00	0.94	0.97	0.99	0.95	94
	Tennis	0.97	0.9	0.93	0.95	0.91	87
	Vitiligo	0.98	0.97	0.97	0.98	0.97	94

The LSTM model outperformed the RNN slightly in most cases. It achieved perfect classification for several disease classes, including Acne, Eczema, Ringworm, and Vitiligo. Depression was classified with extremely high accuracy (99%), and COVID-19 also performed well with an F1-score of 0.93. Among sports, all classes, including Badminton, Cricket, Tennis, and FIFA_2022, had high precision, recall, and accuracy above 96%. Natural

disasters like earthquakes and landslides were again perfectly classified, but Flood and Hurricane remained weak points. Flood had the lowest performance with 39.5% accuracy and an F1-score of 0.36, while Hurricane showed moderate improvement with 55% accuracy and an F1-score of 0.54.

In summary, both RNN and LSTM performed excellently across most classes, especially in sports and diseases. However, LSTM slightly outperformed RNN overall, particularly in classifying disease-related events and achieving perfect scores in multiple classes, while both models struggled with classes like Flood and Hurricane.

The BiLSTM model delivered excellent classification results across nearly all classes, particularly excelling in the disease and sports categories. For diseases, the model achieved perfect performance (precision, recall, F1-score, and accuracy all equal to 1) in Acne, Eczema, Ringworm, and Vitiligo. Depression was also handled effectively with an F1-score of 0.99 and 98.5% accuracy, while COVID-19 showed slightly lower recall (0.88) but still maintained a strong F1-score of 0.93 and 93% accuracy. In the sports category, BiLSTM performed consistently well. Badminton, Cricket, Tennis, and FIFA_2022 were all classified with high precision and recall, each scoring between 96% and 98.5% in accuracy, and F1-scores ranging from 0.96 to 0.98.

Natural disaster classes like Earthquake and Landslide were flawlessly classified with perfect scores. Flood showed improvement compared to other models with an F1-score of 0.67 and accuracy of 72.5%, despite a relatively low precision (0.52). However, Hurricane remained a challenging class with the weakest performance, only 34.5% accuracy and an F1-score of 0.26.

The BiRNN model demonstrated strong classification performance across most classes, especially in health-related and sports categories. It achieved high accuracy for Acne (94%), Ringworm (94%), Vitiligo (94%), and Depression (92%), with F1-scores around 0.96–0.97, indicating balanced precision and recall. Eczema showed a high recall (0.95) and a good F1-score (0.89), although with slightly lower precision (0.84). In sports categories, the model performed robustly. Badminton, Cricket, and Tennis had high F1-scores between 0.93 and 0.97, with accuracies ranging from 87% to 95%. FIFA_2022 also showed a solid performance with a 0.95 F1-score and 91% accuracy. Natural disasters like Earthquake and Landslide were handled well, with Earthquake showing a 0.98 F1-score and 96% accuracy, and Landslide achieving perfect scores. Flood classification improved with a 0.67 F1-score, though overall accuracy remained lower at 49%. Hurricane remained the most difficult class, scoring the lowest across all metrics with only 13% accuracy and a 0.21 F1-score.

Overall, the CNN model showed limited capability, performing well only on a few static feature classes such as Acne, Earthquake, and Vitiligo (100% accuracy), but underperforming on complex classes like Flood (74.5%) and Hurricane (31%). RNN records an improvement over CNN, achieving better F1-scores and accuracies for most classes, though it still struggled with Flood (57.5%) and Hurricane (47.5%). LSTM provided a notable performance boost, offering perfect accuracy (100%) for seven classes, including Acne, Earthquake, and Eczema. However, it showed reduced accuracy for Flood (39.5%) and moderate performance for Hurricane (55%). BiLSTM outperformed all models overall, with consistently high accuracy across classes and improved handling of challenging categories like Flood (72.5%), though it too struggled with Hurricane (34.5%). BiRNN offered mixed results, performing well in some classes (e.g., Landslide: 100%), but poorly in others, particularly Hurricane (13%) and COVID-19 (78%). In summary, BiLSTM emerged as the most effective model due to its robust and balanced performance, followed by LSTM and RNN. While CNN excelled in static feature extraction, it lacked generalization for sequential or ambiguous classes. BiRNN was the least consistent, particularly for disaster-related

events. The table below highlights the accuracy trends, indicating where each model excels or falls short.

As can be observed from Tables 7 and 8, the comparatively lower performance for the Flood and Hurricane categories is recorded. This is mainly due to the highly raw, heterogeneous, and information-poor nature of the corresponding social media posts. Many tweets contain limited contextual information, informal expressions, spelling variations, hashtags, abbreviations, and generic disaster-related terms, making it difficult to distinguish between the two event categories. For example, a Flood-related tweet such as "Heavy rain and strong winds have left the streets completely underwater" may be incorrectly classified as Hurricane because of the presence of "strong winds" and the implied storm context.

Similarly, a Hurricane-related tweet such as "Homes are underwater after the storm hit the area overnight" may be classified as Flood because the explicit "underwater" cue provides a stronger flood-related signal. Other examples, such as "People are evacuating as water levels continue to rise" or "Rainfall is getting worse, streets are filling with water", provide insufficient contextual information to clearly identify the underlying event. Furthermore, considerable lexical similarity exists between the two categories, with terms such as rain, storm, wind, water, damage, and evacuation frequently appearing in both. These characteristics make Flood and Hurricane tweets more challenging to classify reliably and consequently contribute to their comparatively lower F1-scores.

Some examples of misclassifications for Flood and Hurricane categories are presented in Table 9.

Table 9. Representative misclassification examples for the Flood and Hurricane categories.

Actual Category	Predicted Category	Representative Tweet
Flood	Hurricane	"Heavy rain and strong winds have left the streets completely underwater."
Hurricane	Flood	"Homes are underwater after the storm hit the area overnight."
Flood	Hurricane	"The storm has caused massive damage across the region."
Hurricane	Flood	"People are evacuating as water levels continue to rise."
Flood	Hurricane	"Rainfall is getting worse, streets are filling with water."
Hurricane	Flood	"Heavy rain is causing flooding and roads are becoming impossible to cross."

These examples illustrate the ambiguity arising from short, raw, and information-poor tweets, as well as the substantial lexical overlap between the two event categories.

Although several event categories achieved perfect precision, recall, F1-score, and accuracy, these results should not be interpreted as evidence that the proposed models achieve universally perfect generalization across real-world social media data. Rather, the observed 100% performance may be associated with the relatively high lexical and contextual separability of these particular categories within the datasets used in the experiments. Certain categories, such as Acne, Eczema, Ringworm, Vitiligo, Earthquake, and Landslide, may contain distinctive terminology, recurring contextual patterns, or more clearly defined event-related expressions that make them easier to distinguish from their corresponding non-event samples.

6.4. Hybrid Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) Model

In this section, we first present and evaluate the best-performing CNN and LSTM models using Word2Vec and TF-IDF representations individually and then analyze the performance of the CNN+LSTM model based on the combined Word2Vec + TF-IDF representation. This three-way analysis allows us to examine the individual behavior of the two feature representations and to determine whether their integration provides additional benefits for social media event detection.

Table 10 presents the class-wise performance of the CNN+LSTM model using Word2Vec-based feature representation. Overall, Word2Vec demonstrates strong performance across most event categories, particularly for classes characterized by distinctive semantic patterns. The highest performance is observed for Acne, Landslide, and Vitiligo, which achieve an F1-score of 1.00, an F2-score of 1.00, and an accuracy of 100%. These results indicate that the semantic relationships learned through Word2Vec, together with the local feature extraction and sequential modeling capabilities of CNN+LSTM, are highly effective for these relatively distinctive categories.

Several other categories also demonstrate consistently high performance. Earthquake and Eczema achieve an F1-score of 0.99, an F2-score of 0.99, and an accuracy of 99.5%, while Depression achieves an F1-score of 0.98, an F2-score of 0.98, and an accuracy of 98.5%. Among the Sports Event categories, Badminton and Tennis obtain F1-scores of 0.97, with corresponding F2-scores of 0.96 and 0.95, respectively. These results suggest that Word2Vec is particularly effective when the target event exhibits relatively stable semantic and contextual patterns.

Table 10. Class-wise performance of the CNNLSTM model using Word2Vec representation.

Class	Precision	Recall	F1-Score	F0.5-Score	F2-Score	Accuracy (%)
Acne	1.00	1.00	1.00	1.00	1.00	100
Badminton	0.99	0.96	0.97	0.98	0.96	98.0
COVID-19	0.93	0.90	0.91	0.92	0.90	91.5
Cricket	0.98	0.94	0.96	0.97	0.94	96.0
Depression	0.99	0.98	0.98	0.99	0.98	98.5
Earthquake	1.00	0.99	0.99	1.00	0.99	99.5
Eczema	1.00	0.99	0.99	1.00	0.99	99.5
FIFA_2022	0.99	0.91	0.95	0.97	0.92	95.0
Flood	0.48	0.62	0.54	0.51	0.60	55.0
Hurricane	0.49	0.81	0.61	0.53	0.74	66.0
Landslide	1.00	1.00	1.00	1.00	1.00	100
Ringworm	1.00	0.99	0.99	1.00	0.99	99.5
Tennis	0.99	0.95	0.97	0.98	0.95	97.0
Vitiligo	1.00	1.00	1.00	1.00	1.00	100

In contrast, Flood and Hurricane remain the most challenging categories, with F1-scores of 0.54 and 0.61, respectively. Their lower performance is consistent with the ambiguity and lexical overlap frequently observed in raw social media posts, where terms such as rain, storm, wind, water, and damage may occur across both event types. Nevertheless, Hurricane achieves a relatively high recall of 0.81, indicating that the model identifies a substantial proportion of relevant instances, although its lower precision indicates considerable

confusion with non-Hurricane content. Flood exhibits a lower F1-score primarily because of its relatively low precision and the resulting false-positive predictions.

Considering all 14 categories, the Word2Vec-based model achieves an average F1-score of approximately 91.86%, an average F2-score of 92.57%, and an average accuracy of 92.54%. Thus, despite the difficulties associated with a small number of ambiguous disaster categories, Word2Vec provides a strong overall semantic representation for the majority of the evaluated event classes.

Table 11 reports the corresponding results for the CNN+LSTM model using TF-IDF-based feature representation. Similar to Word2Vec, TF-IDF produces very strong results for event categories with distinctive lexical characteristics. Acne, Landslide, and Vitiligo again achieve an F1-score of 1.00, an F2-score of 1.00, and 100% accuracy. Earthquake, Eczema, and Ringworm also demonstrate near-perfect performance, with F1-scores of 0.99 and accuracies of 99.5%. These results indicate that event-specific discriminative terms can provide highly informative signals for classification when the vocabulary associated with an event is sufficiently distinctive. As with Word2Vec, the lowest performance is observed for Flood and Hurricane, with F1-scores of 0.55 and 0.60, respectively. This indicates that term-level weighting alone is insufficient to fully resolve the contextual ambiguity between these closely related disaster events. Nevertheless, Hurricane achieves a recall of 0.79, while Flood achieves a recall of 0.61, suggesting that both categories remain substantially more difficult than the semantically distinct event classes.

Across all 14 categories, the TF-IDF-based CNN+LSTM model obtains an average F1-score of approximately 91.71%, an average F2-score of 92.29%, and an average accuracy of 92.36%. Although these values are very close to those obtained with Word2Vec, Word2Vec shows marginally higher overall average results. This indicates that both representations provide strong but somewhat different information for event classification.

Table 11. Class-wise performance of the CNN+LSTM model using TF-IDF representation.

Class	Precision	Recall	F1-Score	F0.5-Score	F2-Score	Accuracy (%)
Acne	1.00	1.00	1.00	1.00	1.00	100
Badminton	0.99	0.95	0.97	0.98	0.96	97.5
COVID-19	0.94	0.89	0.91	0.93	0.90	91.5
Cricket	0.98	0.94	0.96	0.97	0.94	96.0
Depression	0.98	0.97	0.97	0.98	0.97	97.5
Earthquake	1.00	0.99	0.99	1.00	0.99	99.5
Eczema	1.00	0.99	0.99	1.00	0.99	99.5
FIFA_2022	0.98	0.91	0.94	0.96	0.92	94.5
Flood	0.50	0.61	0.55	0.52	0.59	55.5
Hurricane	0.49	0.79	0.60	0.52	0.71	65.0
Landslide	1.00	1.00	1.00	1.00	1.00	100
Ringworm	1.00	0.99	0.99	1.00	0.99	99.5
Tennis	0.99	0.95	0.97	0.98	0.96	97.0
Vitiligo	1.00	1.00	1.00	1.00	1.00	100

The individual experiments demonstrate that Word2Vec and TF-IDF exhibit complementary characteristics. Word2Vec is designed to capture semantic and contextual relationships among words, whereas TF-IDF emphasizes the relative importance and discriminative value of individual terms. The results show that both approaches perform

exceptionally well for highly distinctive categories such as Acne, Landslide, Vitiligo, Earthquake, and Eczema, while both experience greater difficulty with semantically overlapping categories such as Flood and Hurricane.

Therefore, the two representations were combined to provide the CNN+LSTM model with a richer feature space containing both semantic relationships and discriminative lexical information. The resulting performance is analyzed next using the integrated Word2Vec + TF-IDF CNN+LSTM model. Table 12 presents the results obtained by the hybrid CNN+LSTM model based on the integration of the Word2Vec and TFIDF representations.

As can be observed from Table 12 and Figures 14–16 the performance of the CNN+LSTM model across the various classes showcases its strengths and weaknesses, with several notable observations. The model achieved perfect Precision, Recall, F1-score, F0.5-score, F2-score, and 100% accuracy for Acne, Earthquake, Eczema, Landslide, Ringworm, and Vitiligo. These results suggest that these classes had distinct and easily identifiable features, enabling the model to classify them with absolute precision. This could also indicate the well-balanced representation of these classes in the training dataset.

Table 12. Class-wise performance of the integrated CNN+LSTM model based on Word2Vec+TFIDF representation.

Class	Precision	Recall	F1-Score	F0.5-Score	F2-Score	Accuracy (%)
Acne	1.00	1.00	1.00	1.00	1.00	100
Badminton	1.00	0.97	0.98	0.98	0.97	98.5
COVID-19	0.94	0.92	0.93	0.94	0.92	93
Cricket	0.99	0.95	0.97	0.98	0.96	97
Depression	0.99	0.99	0.99	0.99	0.99	99
Earthquake	1.00	1.00	1.00	1.00	1.00	100
Eczema	1.00	1.00	1.00	1.00	1.00	100
FIFA_2022	0.99	0.93	0.96	0.97	0.94	96
Flood	0.49	0.64	0.56	0.52	0.61	56.5
Hurricane	0.5	0.85	0.63	0.54	0.76	67.5
Landslide	1.00	1.00	1.00	1.00	1.00	100
Ringworm	1.00	1.00	1.00	1.00	1.00	100
Tennis	1.00	0.96	0.98	0.99	0.97	98
Vitiligo	1.00	1.00	1.00	1.00	1.00	100

For categories like Badminton, Cricket, and Tennis, the model achieved high Precision (1.00 or 0.99) and Recall values ranging from 0.95 to 0.97. The corresponding F1-scores and related metrics remained close to 1.0, with overall accuracies between 97% and 98.5%. These results indicate that while these classes were well-represented in the data, the model encountered minor challenges in recalling all instances, likely due to slight feature overlap with other sports-related categories, such as FIFA_2022.

The COVID-19 and FIFA_2022 classes demonstrated slightly lower performance compared to the top-performing classes. For COVID-19, the model achieved a Precision of 0.94 and Recall of 0.92, resulting in an F1-score of 0.93 and an accuracy of 93%. Similarly, FIFA_2022 reached a Precision of 0.99, Recall of 0.93, and an F1-score of 0.96 with an accuracy of 96%. These values suggest that while the model generally identified these categories well,

the lower Recall values indicate some misclassification, possibly due to shared contextual features with related categories like Depression (for COVID-19) and other sports classes (for FIFA_2022).

As depicted in Figure 14, the performance for Flood and Hurricane highlights areas where the model struggled. The Flood class had a Precision of 0.49 and a Recall of 0.64, leading to an F1-score of 0.56 and an accuracy of only 56.5%. Similarly, Hurricane achieved a Precision of 0.50 and a Recall of 0.85, resulting in an F1-score of 0.63 and accuracy of 67.5%. These discrepancies indicate that the model faced difficulties in learning distinct features for these classes, potentially due to overlapping characteristics with other natural disaster categories (e.g., Earthquake or Landslide). The higher Recall for Hurricane, compared to its Precision, suggests that the model predicted this class more often than it should, leading to a higher rate of false positives.

In comparing the results within classes, the model demonstrated outstanding consistency and reliability for distinct and isolated categories like Acne, Earthquake, and Vitiligo. On the other hand, for classes with subtle feature overlaps (e.g., Flood, Hurricane, and FIFA_2022), performance metrics dropped due to feature ambiguity. Sequential dependencies, a strength of the LSTM component, helped in improving Recall for categories like Hurricane, though they were insufficient to fully distinguish these complex classes. Overall, the CNN+LSTM model showcased exceptional performance for simpler, well-represented classes but faced challenges with ambiguous and imbalanced classes like Flood and Hurricane. Enhancing data representation for underperforming classes or employing techniques like data augmentation or transfer learning could further refine the model's performance.

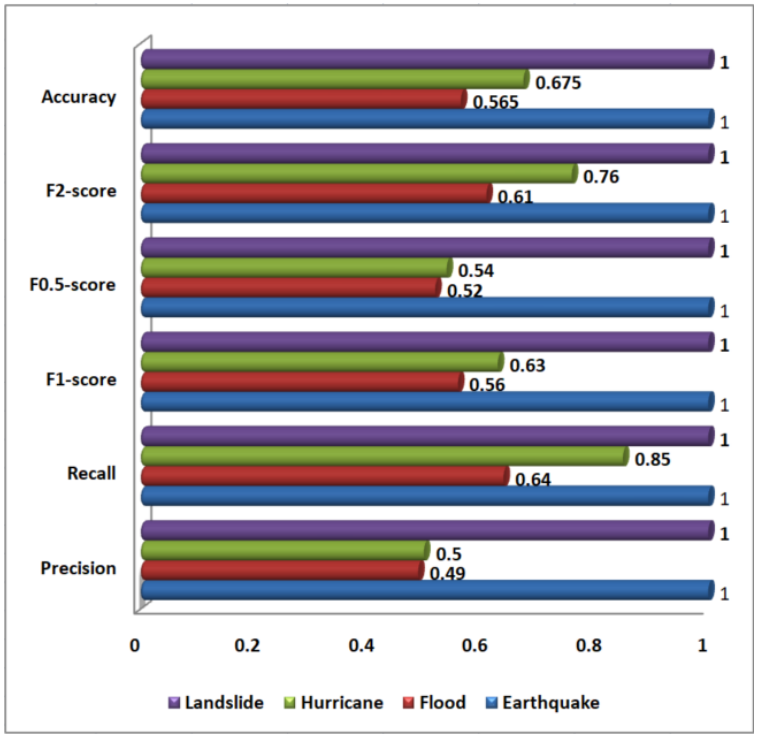


Figure 14. CNN+LSTM for Natural Disaster.

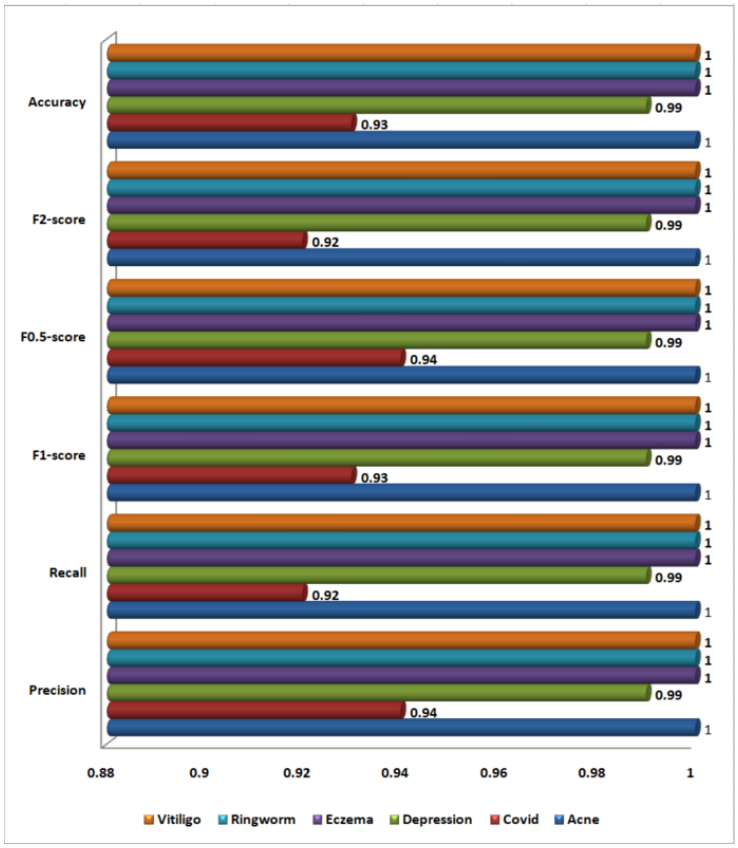


Figure 15. CNN+LSTM for Disease Event classification.

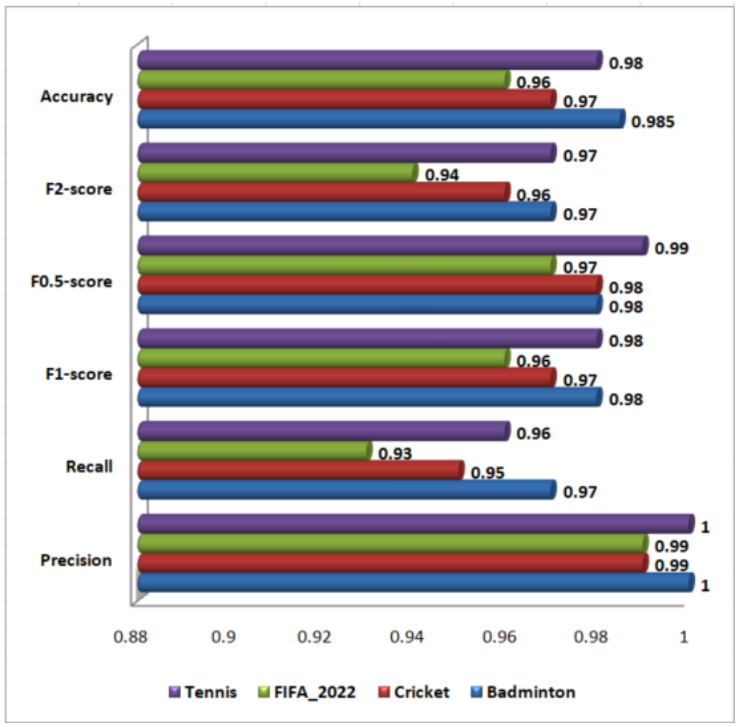


Figure 16. CNN+LSTM for Sports Event classification.

7. Comparative Study

The proposed integrated CNN+LSTM architecture demonstrates strong classification performance across a diverse range of event categories evaluated in this study. To place these results in the context of existing research, we compare them with previously reported

findings from studies such as SatCoBiLSTM, CrisisSpot, Dual-CNN, COfEE, and FireExpert. However, these studies employed different datasets, data distributions, preprocessing procedures, experimental configurations, and evaluation protocols. Therefore, the reported results are presented here as contextual references rather than as direct, like-for-like performance comparisons. Among the previously reported approaches, SatCoBiLSTM achieved F1-scores of 94.00% on CrisisLexT6, 85.02% on CrisisLexT26, and 89.01% on 2CTweets. These results demonstrate strong performance on several crisis-related textual datasets. However, the variation across datasets also indicates that performance can depend substantially on the characteristics and domain of the evaluation data. Similarly, CrisisSpot reported an F1-score of 95.50% on the CrissMDD dataset and 50.10% on TSEqD. This substantial variation illustrates the difficulty of comparing event detection performance across datasets with different characteristics and distributions.

Other approaches have also reported varying levels of performance. Dual-CNN achieved an F1-score of 79.80%, while its associated CNN, SVM, CART, and Naive Bayes configurations achieved 79.70%, 78.50%, 72.30%, and 73.30%, respectively. COfEE reported F1-scores of 76.60% for the SMEE+LSTM+CNN configuration, 81.40% for SMEE+LSTM, and 70.30% for LSTM+CNN. FireExpert reported an F1-score of 61.01% on the FFYOLO dataset. These findings provide useful evidence of the performance achieved by different architectures under their respective experimental settings. Nevertheless, the differences in datasets and evaluation protocols prevent these values from being interpreted as direct evidence that one approach is universally superior to another.

Under the experimental setting of the present study, the proposed CNN+LSTM architecture achieves high F1-scores across a heterogeneous set of event categories (Table 13). Specifically, it achieves 100% F1-score for Acne, Earthquake, Landslide, Ringworm, and Vitiligo, 99% for Depression, 98% for Badminton and Tennis, 97% for Cricket, and 96% for FIFA_2022. The model also achieves F1-scores of 56% and 63% for Flood and Hurricane, respectively. These results demonstrate that the proposed architecture performs strongly on several of the event-classification tasks evaluated in this study, while also indicating that performance varies across event categories.

The differences among event categories are important because social media content can exhibit substantial variation in vocabulary, discourse patterns, and contextual characteristics. The CNN component is designed to capture local textual patterns, while the LSTM component models sequential dependencies in the input representation. Their combination provides a mechanism for learning both local and sequential characteristics of event-related text. The relatively high performance observed for several categories suggests that this combination is effective for the datasets and experimental conditions considered in this study.

It is important to emphasize that the results reported for SatCoBiLSTM, CrisisSpot, Dual-CNN, COfEE, and FireExpert were obtained under their respective experimental settings and should therefore not be interpreted as direct baselines for the proposed method. In particular, differences in dataset composition, class distribution, preprocessing, train-test splitting, and evaluation methodology can substantially influence the resulting performance. The present study relies primarily on comparisons among models evaluated using the same dataset and experimental protocol, together with the ablation analysis reported in this work.

Overall, the results indicate that the proposed CNN+LSTM framework provides strong performance across diverse event categories within the experimental setting of this study. Its evaluation across health-related events (e.g., Acne and Ringworm), natural disasters (e.g., Earthquake, Flood, and Hurricane), sports (e.g., FIFA_2022 and Cricket), and mental-health-related content (e.g., Depression) demonstrates its applicability to heterogeneous social media event-classification tasks. Nevertheless, further controlled experiments using

common benchmark datasets and identical data-splitting and evaluation protocols would be necessary to establish a definitive comparison with existing approaches. Future work can therefore investigate cross-dataset evaluation, multilingual support, and real-time prediction to further assess the robustness and practical applicability of the framework.

Table 13. Comparison of reported F1-scores across different datasets and configurations.

Study	F1-Score	Dataset/Configuration/Class
FireExpert [54]	61.01%	FFYOLO
SatCoBiLSTM [55]	94.00% 85.02% 89.01%	CrisisLexT6 (Dataset 1) CrisisLexT26 (Dataset 2) 2CTweets (Dataset 3)
CrisisSpot [56]	95.50% 50.10%	CrissMDD TSEqD
Dual-CNN [37]	79.80% 79.70% 78.50% 73.30% 72.30%	Dual-CNN CNN SVM Naive Bayes CART
COfEE [57]	76.60% 70.30% 81.40%	SMEE+LSTM+CNN LSTM+CNN SMEE+LSTM
Proposed Approach	100% 98% 93% 97% 99% 100% 100% 96% 56% 63% 100% 100% 98% 100%	Acne Badminton COVID-19 Cricket Depression Earthquake Eczema FIFA_2022 Flood Hurricane Landslide Ringworm Tennis Vitiligo

8. Conclusions and Observations

This paper introduced a social event detection framework built upon a customized layered CNN+LSTM architecture, designed to capture both local n-gram features and long-range semantic dependencies within social media text. The model integrates convolutional filters for spatial pattern extraction and sequential LSTM layers for temporal encoding, enabling precise classification across diverse event categories. Our approach was evaluated against several state-of-the-art models, demonstrating superior generalizability and classification performance, particularly under conditions of data imbalance and noisy input typical of social media streams.

The proposed integrated CNN+LSTM model was benchmarked across 14 distinct event classes drawn from heterogeneous sources, including health-related, environmental, and sports-related events. The architecture achieved outstanding F1-scores of 100% on Acne, Earthquake, Landslide, Ringworm, and Vitiligo, reflecting its robustness in handling both frequent and rare event types. Additionally, it obtained 99% on Depression, 98% on Badminton and Tennis, 97% on Cricket, 96% on FIFA_2022, 93% on COVID-19, 63% on Hurricane, and 56% on Flood.

The observed performance of the proposed model highlights the potential of layered fusion strategies for extracting discriminative features for event identification. Unlike approaches based on single configurations or shallow classifiers such as SVM and CART, the proposed system employs deep hierarchical encoding to capture both syntactic and semantic patterns in short, unstructured social media texts.

The study highlights the potential of domain-specific text representations and customized neural architectures to improve classification reliability for real-time event detection systems. Future research will extend the proposed framework by incorporating pre-trained Transformer-based models, such as BERT and RoBERTa, and conducting systematic ablation studies to assess the individual contributions of text representations, AI-generated data augmentation, and model architecture.

Future research will further assess the robustness and statistical reliability of the proposed framework through repeated experiments with different random seeds, confidence-interval estimation, and formal statistical significance testing. These analyses will provide stronger evidence of model stability and help determine whether observed performance differences are statistically meaningful.

Further investigations will also examine the impact of retaining URLs, hashtags, user mentions, and emojis, which may provide useful semantic, contextual, or social-media-specific information.

Additionally, future work may explore generative LLMs for event explanation and summarization and multimodal data integration for more robust cross-platform event detection.

In conclusion, the fine-tuned and integrated CNN+LSTM model presented in this work sets a new benchmark for social event detection tasks. It not only outperforms other individual models under the experimental setting considered in this study in terms of precision and robustness but also provides a scalable and adaptable solution for real-world crisis informatics, public health monitoring, and digital event surveillance applications.

Author Contributions: Conceptualization, M.A.W.; methodology, M.A.W. and S.S.; software, M.A.W. and S.S.; validation, M.A.W. and K.A.S.; formal analysis, M.A.W. and S.S.; investigation, S.S.; resources, K.A.S. and A.A.; data curation, S.S.; writing—original draft preparation, M.A.W. and S.S.; writing—review and editing, K.A.S. and A.A.; visualization, M.A.W.; supervision, K.A.S. and A.A.; project administration, A.A.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: The work was funded bu Princess Nourah bint Abdulrahman University Researchers Supporting Project (number PNURSP2026R757), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The link to the data repository related to this project can be made available upon request. If required, it will be made publicly available on the author's GitHub profile (https://github.com/mudasirahmadwani, accessed on 19 June 2026) under the title of the paper.

Acknowledgments: The work was supported by the Princess Nourah bint Abdulrahman University Researchers Supporting Project (number PNURSP2026R757), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Goswami, A.; Kumar, A. A survey of event detection techniques in online social networks. *Soc. Netw. Anal. Min.* **2016**, *6*, 107. [CrossRef]

2. Zhou, X.; Chen, L. Event detection over twitter social media streams. *VLDB J.* **2014**, *23*, 381–400. [CrossRef]
3. Edosomwan, S.; Prakasan, S.K.; Kouame, D.; Watson, J.; Seymour, T. The history of social media and its impact on business. *J. Appl. Manag. Entrep.* **2011**, *16*, 79.
4. Sykora, M.; Elayan, S.; Hodgkinson, I.R.; Jackson, T.W.; West, A. The power of emotions: Leveraging user generated content for customer experience management. *J. Bus. Res.* **2022**, *144*, 997–1006. [CrossRef]
5. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G.; Lv, J. Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Trans. Cybern.* **2020**, *50*, 3840–3854. [CrossRef]
6. Guo, T.; Dong, J.; Li, H.; Gao, Y. Simple convolutional neural network on image classification. In Proceedings of the 2017 IEEE 2nd International conference on big data analysis (ICBDA), Beijing, China, 10–12 March 2017; pp. 721–724.
7. Elngar, A.A.; Arafa, M.; Fathy, A.; Moustafa, B.; Mahmoud, O.; Shaban, M.; Fawzy, N. Image classification based on CNN: A survey. *J. Cybersecur. Inf. Manag.* **2021**, *6*, 18–50.
8. Huang, R.; Wei, C.; Wang, B.; Yang, J.; Xu, X.; Wu, S.; Huang, S. Well performance prediction based on Long Short-Term Memory (LSTM) neural network. *J. Pet. Sci. Eng.* **2022**, *208*, 109686. [CrossRef]
9. Kumar, J.; Goomer, R.; Singh, A.K. Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters. *Procedia Comput. Sci.* **2018**, *125*, 676–682. [CrossRef]
10. Muhuri, P.S.; Chatterjee, P.; Yuan, X.; Roy, K.; Esterline, A. Using a long short-term memory recurrent neural network (LSTM-RNN) to classify network attacks. *Information* **2020**, *11*, 243. [CrossRef]
11. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
12. Imran, M.; Castillo, C.; Diaz, F.; Vieweg, S. Processing social media messages in mass emergency: A survey. *ACM Comput. Surv. (CSUR)* **2015**, *47*, 1–38. [CrossRef]
13. 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, Raleigh, NC, USA, 26–30 April 2010; pp. 851–860.
14. De Albuquerque, J.P.; Herfort, B.; Brenning, A.; Zipf, A. A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 667–689. [CrossRef]
15. Lee, C.H.; Kim, Y.C. Real-time event detection in Twitter stream for viral marketing. *Ind. Manag. Data Syst.* **2016**, *116*, 1804–1820.
16. De Choudhury, M.; Gamon, M.; Counts, S.; Horvitz, E. Predicting depression via social media. In Proceedings of the International AAAI Conference on Web and Social Media, Cambridge, MA, USA, 8–11 July 2013; Volume 7, pp. 128–137.
17. Paul, M.J.; Dredze, M. *Social Monitoring for Public Health*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2017.
18. Wani, M.A. Optimized CNN–LSTM Modeling for Crisis Event Detection in Noisy Social Media Streams. *Mathematics* **2026**, *14*, 1369. [CrossRef]
19. Mohd, M. Named entity patterns across news domains. In Proceedings of the BCS IRSG Symposium: Future Directions in Information Access, Glasgow, UK, 28–29 August 2007; pp. 30–36.
20. Wani, M.A.; ELAffendi, M.; Shakil, K.A.; Abuhaimed, I.M.; Nayyar, A.; Hussain, A.; Abd El-Latif, A.A. Toxic Fake News Detection and Classification for Combating COVID-19 Misinformation. *IEEE Trans. Comput. Soc. Syst.* **2023**, *11*, 5101–5118. [CrossRef]
21. Wani, M.A.; ELAffendi, M.; Bours, P.; Imran, A.S.; Hussain, A.; Abd El-Latif, A.A. CoDeS: A Deep Learning Framework for Identifying COVID-Caused Depression Symptoms. *Cogn. Comput.* **2024**, *16*, 305–325. [CrossRef]
22. Wani, M.A.; ELAffendi, M.A.; Shakil, K.A.; Imran, A.S.; Abd El-Latif, A.A. Depression screening in humans with AI and deep learning techniques. *IEEE Trans. Comput. Soc. Syst.* **2022**, *10*, 2074–2089. [CrossRef]
23. Wani, M.A.; ElAffendi, M.; Shakil, K.A. AI-Generated Spam Review Detection Framework with Deep Learning Algorithms and Natural Language Processing. *Computers* **2024**, *13*, 264. [CrossRef]
24. Wani, M.; Agarwal, N.; Bours, P. Impact of Unreliable Content on Social Media Users during COVID-19 and Stance Detection System. *Electronics* **2021**, *10*, 5. [CrossRef]
25. Shakil, K.A.; Tabassum, K.; Alqahtani, F.S.; Wani, M.A. Analyzing user digital emotions from a holy versus non-pilgrimage city in Saudi Arabia on twitter platform. *Appl. Sci.* **2021**, *11*, 6846. [CrossRef]
26. Petrovic, S.; Osborne, M.; Lavrenko, V. Streaming first story detection with application to twitter. In *Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT'10)*; Association for Computational Linguistics: Los Angeles, CA, USA, 2010; pp. 181–189.
27. Mathioudakis, M.; Koudas, N. Twittermonitor: Trend detection over the twitter stream. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, Indianapolis, IN, USA, 6–10 June 2010; pp. 1155–1158.
28. Atefeh, F.; Khreich, W. A survey of techniques for event detection in twitter. *Comput. Intell.* **2015**, *31*, 132–164. [CrossRef]
29. 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, Melbourne, Australia, 24–28 August 1998; pp. 28–36.

30. Yang, Y.; Carbonell, J.G.; Brown, R.D.; Pierce, T.; Archibald, B.T.; Liu, X. Learning approaches for detecting and tracking news events. *IEEE Intell. Syst. Their Appl.* **1999**, *14*, 32–43. [CrossRef]
31. McCreadie, R.; Macdonald, C.; Ounis, I. News vertical search: When and what to display to users. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 28 July–1 August 2013; pp. 253–262.
32. Mele, I.; Bahrainian, S.A.; Crestani, F. Event mining and timeliness analysis from heterogeneous news streams. *Inf. Process. Manag.* **2019**, *56*, 969–993. [CrossRef]
33. Thiombiano, J.; Traore, Y.; Malo, S. Early alert of an outbreak based on event detection and tracking on social networks: The case of COVID-19 and meningitis. In Proceedings of the 2023 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS), Murcia, Spain, 19–22 June 2023; pp. 85–91.
34. Cheng, T.; Wicks, T. Event detection using Twitter: A spatio-temporal approach. *PLoS ONE* **2014**, *9*, e97807. [CrossRef]
35. Doulamis, N.D.; Doulamis, A.D.; Kokkinos, P.; Varvarigos, E.M. Event detection in twitter microblogging. *IEEE Trans. Cybern.* **2015**, *46*, 2810–2824. [CrossRef]
36. Alomari, E.; Katib, I.; Mehmood, R. Iktishaf: A big data road-traffic event detection tool using Twitter and spark machine learning. *Mob. Netw. Appl.* **2023**, *28*, 603–618. [CrossRef]
37. Burel, G.; Saif, H.; Fernandez, M.; Alani, H. On semantics and deep learning for event detection in crisis situations. In Proceedings of the Workshop on Semantic Deep Learning (SemDeep), at ESWC 2017, Portoroz, Slovenia, 29 May 2017.
38. Dabiri, S.; Heaslip, K. Developing a Twitter-based traffic event detection model using deep learning architectures. *Expert Syst. Appl.* **2019**, *118*, 425–439. [CrossRef]
39. Imran, M.; Mitra, P.; Castillo, C. Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, 23–28 May 2016.
40. Wang, Z.; Ye, X. Hurricane Harvey Twitter Dataset: A multimodal analysis of disaster response. In Proceedings of the 1st Workshop on Natural Language Processing for Social Media, Melbourne, Australia, 20 July 2018; pp. 1–10.
41. Murthy, D.; Longwell, S.A. Twitter and disasters: The uses of Twitter during the 2010 Pakistan floods. *Inf. Commun. Soc.* **2013**, *16*, 837–855.
42. Jawad, K. Flood Tweets Sentiment Analysis (Pakistan 2022). 2022. Available online: https://www.kaggle.com/ (accessed on 23 September 2026).
43. SanderP. Badminton BWF World Tour. 2021. Available online: https://www.kaggle.com/datasets/sanderp/badminton-bwf-world-tour (accessed on 8 December 2025).
44. Grüttner, A.; Vitisvorakarn, M.; Wambsganss, T.; Rietsche, R.; Back, A. The New Window to Athletes' Soul—What Social Media Tells Us About Athletes' Performances. In Proceedings of the 53rd Hawaii International Conference on System Sciences, Maui, HI, USA, 7–10 January 2020; pp. 2479–2488. [CrossRef]
45. Singh, V.; Kumar, A.; Gupta, S. Real-time cricket event detection from Twitter streams. In Proceedings of the 2018 IEEE International Conference on Big Data, Seattle, DC, USA, 10–13 December 2018; pp. 2374–2379.
46. Aloufi, S.; El Saddik, A. Sentiment identification in football-specific tweets. *IEEE Access* **2018**, *6*, 78609–78621. [CrossRef]
47. Miller, K.; Smith, R.; Johnson, L. Social media as a source of epidemiological data for skin conditions: A systematic review. *J. Med. Internet Res. Dermatol.* **2021**, *4*, e25678.
48. Naseem, U.; Razzak, I.; Khushi, M.; Eklund, P.W.; Kim, J. COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 1003–1015. [CrossRef]
49. Bird, S. NLTK: The natural language toolkit. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, Sydney, Australia, 17–18 July 2006; pp. 69–72.
50. Verdonck, T.; Baesens, B.; Óskarsdóttir, M.; vanden Broucke, S. Special issue on feature engineering editorial. *Mach. Learn.* **2021**, *113*, 3917–3928. [CrossRef]
51. Aizawa, A. An information-theoretic perspective of tf–idf measures. *Inf. Process. Manag.* **2003**, *39*, 45–65. [CrossRef]
52. Church, K.W. Word2Vec. *Nat. Lang. Eng.* **2017**, *23*, 155–162. [CrossRef]
53. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.
54. Luo, G.; Weng, L.; Li, Y.; Sun, Y.; Hong, Y.; Wu, Y.; Luo, R.; Wang, L.; Wang, C.; Chen, L. FireExpert: Fire Event Identification and Assessment Leveraging Cross-Domain Knowledge and Large Language Model. *IEEE Trans. Mob. Comput.* **2025**, *24*, 5076–5090. [CrossRef]
55. Upadhyay, A.; Meena, Y.K.; Chauhan, G.S. SatCoBiLSTM: Self-attention based hybrid deep learning framework for crisis event detection in social media. *Expert Syst. Appl.* **2024**, *249*, 123604. [CrossRef]

56. Dar, S.S.; Rehman, M.Z.U.; Bais, K.; Haseeb, M.A.; Kumar, N. A social context-aware graph-based multimodal attentive learning framework for disaster content classification during emergencies. *Expert Syst. Appl.* **2025**, *259*, 125337. [CrossRef]
57. Balali, A.; Asadpour, M.; Jafari, S.H. COfEE: A comprehensive ontology for event extraction from text. *Comput. Speech Lang.* **2025**, *89*, 101702. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.