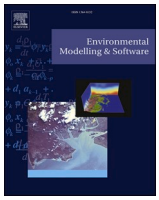
Environmental Modelling and Software 188 (2025) 106421

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



Can large language models effectively reason about adverse weather conditions?

Nima Zafarmomen [a], Vidya Samadi [a,b,*]

[a] Department of Agricultural Sciences, Clemson University, Clemson, SC, USA
[b] Artificial Intelligence Research Institute for Science and Engineering (AIRISE), School of Computing, Clemson University, Clemson, SC, USA

ARTICLE INFO

Handling Editor: Daniel P Ames

Keywords:
Large language model
Text classification
LLaMA
BART
BERT
Adverse weather conditions

ABSTRACT

This paper seeks to answer the question "can Large Language Models (LLMs) effectively reason about adverse weather conditions?". To address this question, we utilized multiple LLMs to harness the US National Weather Service (NWS) flood report data spanning from June 2005 to September 2024. Bidirectional and Auto-Regressive Transformer (BART), Bidirectional Encoder Representations from Transformers (BERT), Large Language Model Meta AI (LLaMA-2), LLaMA-3, and LLaMA-3.1 were employed to categorize data based on predefined labels. The methodology was implemented in Charleston County, South Carolina, USA. Extreme events were unevenly distributed across the training period with the "Cyclonic" category exhibiting significantly fewer instances compared to the "Flood" and "Thunderstorm" categories. Analysis suggests that the LLaMA-3 reached its peak performance at 60% of the dataset size while other LLMs achieved peak performance at approximately 80–100% of the dataset size. This study provided deep insights into the application of LLMs in reasoning adverse weather conditions.

1. Introduction

Weather hazard operations are complex and dynamic, requiring citizens to make critical decisions under time-sensitive and high-pressure conditions(Jayawardene et al., 2021; Saberian et al., 2024). One of the key elements in the decision-making process is the development of disaster response, which represents alternative steps to achieve objectives while taking into account various operational constraints (Goecks and Waytowich, 2023; Zafarmomen et al., 2024). A traditional response plan can be time-consuming during an emergency because the process of navigating through complex procedures, gathering necessary information, and making critical decisions can take significant time, especially when dealing with rapidly evolving and chaotic situations. This can potentially delay critical response actions and rescue plans during the early stages of a disaster. Therefore, existing disaster response plans must be flexible and may require adaptation to meet the requirements of the situation.

Large language models (LLMs) have emerged as valuable tools for analyzing and classifying massive text data and information, offering potential applications in various domains, including disaster response (Ghosh et al., 2022). LLMs are statistical language models, capable of understanding and generating human language by processing massive amounts of data that can be used to generate and translate text, answer questions, and perform other natural language processing (NLP) tasks. LLMs are typically based on Transformer architecture that can be trained on billions of texts and other content.

The success of LLMs has been significant, particularly in the fields of traffic, healthcare, and finance (Jiang et al., 2024; Liu et al., 2023; Yu et al., 2023). Among the various LLM capabilities, text classification has garnered the most attention due to its ability to categorize and extract meaningful information from textual data (Kadhim, 2019). As a result, the natural hazards communities have shown interest in using text classification (Fan et al., 2018; Donratanapat et al., 2020; Jayawardene et al., 2021). Most research in this field has focused on applying text classification to social media data, posts on X (formerly Twitter), to classify whether they contain specific information categories. For instance, Donratanapat et al. (2020) applied data from X to understand the citizen response to flooding in South Carolina (SC), USA. They used NLP to classify the responses into positive, negative, and neutral and identified at-risk locations for flooding in real-time during major hurricane events by integrating geotagged tweets with real-time flood forecast data. Concurrently, de Bruijn et al. (2020) incorporated

* Corresponding author. Department of Agricultural Sciences, Clemson University, Clemson, SC, USA.
E-mail address: samadi@clemson.edu (V. Samadi).

https://doi.org/10.1016/j.envsoft.2025.106421
Received 9 January 2025; Received in revised form 16 February 2025; Accepted 4 March 2025
Available online 5 March 2025
1364-8152/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

contextual hydrological information into a multimodal neural network, significantly improving the accuracy of flood detection by combining textual and hydrological data. Zhou et al. (2022) developed a novel LLM so-called bidirectional encoder representation from transformers (BERT) model for classifying disaster-related X data, identifying rescue requests, and extracting victim information. Karimiziarani and Moradkhani (2023) leveraged text classification and sentiment analysis on disaster-related X data. They categorized tweets into key humanitarian topics, such as warnings, damages, rescue plans, and emergency responses.

Karanjit et al. (2024) used the BERT model as an unsupervised open-domain question-answering system to distinguish flood-related tweets from non-flood-related ones to validate inundation areas. Otal et al. (2024) assessed the Large Language Model Meta AI (LLaMA)-2 variants and the Mistral models with emergency-disaster messages dataset linked to various disasters from social media. They showed these models are well-suited to enhance emergency and crisis management. Wilkho et al. (2024) used a BERT-based ensemble model that automates the multi-label classification of flash flood-related web data into key humanitarian topics, enhancing information extraction for effective disaster management.

Building upon these LLMs advancement, this study examines three types of transformer-based LLMs, including encoder-only, decoder-only, and encoder-decoder models. The use of LLMs in hydrology and environmental sciences is rapidly expanding, with extensive research examining their potential, and diverse applications (Foroumandi et al., 2023; Sajja et al., 2025). Despite significant advancements, existing literature has not been thoroughly tested and compared to the capabilities of various LLMs for weather hazard data detection and classification. Most existing studies primarily focus on social media data (e.g., X/Twitter posts) for event detection, leaving official NWS flood reports underexamined. Furthermore, these prior works often assess a single LLM or a limited set of models, hindering a comprehensive understanding of how different model architectures handle imbalanced and domain-specific disaster texts. To address these gaps, our research systematically evaluated reasoning capabilities of multiple cutting-edge LLMs models (BART, BERT, LLaMA-2, LLaMA-3) on a large, real-world dataset of NWS flood reports. Moreover, data imbalance and high computational costs remain inadequately addressed. This study aims to bridge these gaps by employing seven distinct LLMs along with implementing the Multi-Label Synthetic Oversampling (MLSOL) method to address category imbalance in multi-label datasets.

Furthermore, we utilized Low-Rank Adaptation (LoRA) for efficient fine-tuning and compared these approaches with the few-shot learning capabilities of the models. We also investigated the stability and scalability of different disaster categories within LLMs and assessed the impact of dataset size on model performance. This paper seeks to improve the accuracy and practicality of automated disaster detection systems through multiple LLMs. These advanced LLMs are relatively new and have demonstrated impressive performance across various domains. In this paper, we conducted an extensive study and analysis of their application to disaster data. Our key research contributions are as follows:

(i) **Capability of representation learning**: LLMs excel in discovering nuanced patterns and structures within textual datasets. These models effectively transform raw disaster-related information into meaningful, high-level representations by analyzing broad language contexts. This transformation enabled LLMs to capture subtle linguistic cues and contextual relationships, supporting more accurate text classification.

(ii) **Effectiveness of LLMs in representing complex disaster data patterns:** Disasters often come with intricate combinations of factors and diverse sources. LLMs were adaptive at identifying and differentiating these intricacies. In this study, LLMs were exposed to vast amounts of text during training to build robust internal representations that can adapt to different disaster scenarios.

(iii) **Enhancement of representation quality through fine-tuning and data augmentation:** While LLMs provide a strong baseline, customizing them for a disaster domain typically enhances accuracy. Fine-tuning of these models with domain-specific data tuned the model's parameters and aligned them more closely with the linguistics of hazard-related texts. Meanwhile, data augmentation strategies such as generating additional training samples in underrepresented categories improved the model's ability to manage category imbalance.

2. Methodology

2.1. LLMs

2.1.1. BERT model

BERT has become widely used in NLP research developed by Google (Devlin et al., 2018). Unlike earlier models, this model considers context from both sides of a word (i.e., bidirectional) for more accurate predictions (Lee and Toutanova, 2018). LLMs have demonstrated exceptional few-shot and zero-shot learning abilities, allowing them to tackle complex tasks with minimal or no specific training data (Brown, 2020). Moreover, one of the key strengths of LLMs is their ability to process and understand unstructured data, such as text from reports, social media posts, and other natural language sources (Anderson et al., 2024; Patel et al., 2024). Additionally, this model utilizes unsupervised machine learning (ML) techniques and can be trained effectively for specific tasks such as text classification (Alammary, 2022). When the model receives a paragraph, it tokenizes the text into individual words, converts them to lowercase, and appends the special tokens classification [CLS] and separator [SEP] to signify the beginning and end of the sentences, respectively. The [CLS] token is used for classification tasks, while [SEP] separates different segments or sentences. Then, the tokens are subjected to types of embeddings, including token embeddings, segment embeddings, and position embeddings. In token embeddings, the model assigns a unique embedding to each token based on the WordPiece tokenization technique (Wu, 2016). The segment embeddings take information on the segment and differentiate between different parts of the paragraph. The position embeddings involve information about the position of each token in the sequence. Finally, the combined embeddings vector is the input for the first transformer encoder layer of BERT (Fig. 1).

The multi-layer bidirectional transformer encoder is the cornerstone of BERT's architecture and consists of 12 Transformer encoder blocks with 768 hidden units each. Each block has two main sub-components: (1) a multi-head self-attention mechanism and (2) a position-wise fully connected feed-forward network. The first encoder block receives combined embeddings as the input.

The self-attention mechanism allows the model to assign different attention scores to various tokens within the input sequence. Given a query vector (Q) and multiple key-value pairs (K-V) in the dimension of K (d_k), the attention mechanism calculates the weighted sum of the value vectors based on their similarity to the query. The similarity is determined by the dot product of the query and key vectors, which is then scaled to prevent the values from becoming excessively large and potentially hindering the attention process. The query, key, and value vectors in self-attention originate from the input embeddings. The self-attention (Equation (1)) is stated by

$$Attention\ (Q, K, V) = softmax\left(\frac{QK}{\sqrt{d_k}}\right) V \qquad \text{(}\}\text{Equation}\} 1)$$

In multi-head self-attention, each token is restricted to using the past and future tokens during processing, while the multi-head self-attention mechanism captures relationships between all tokens by allowing each

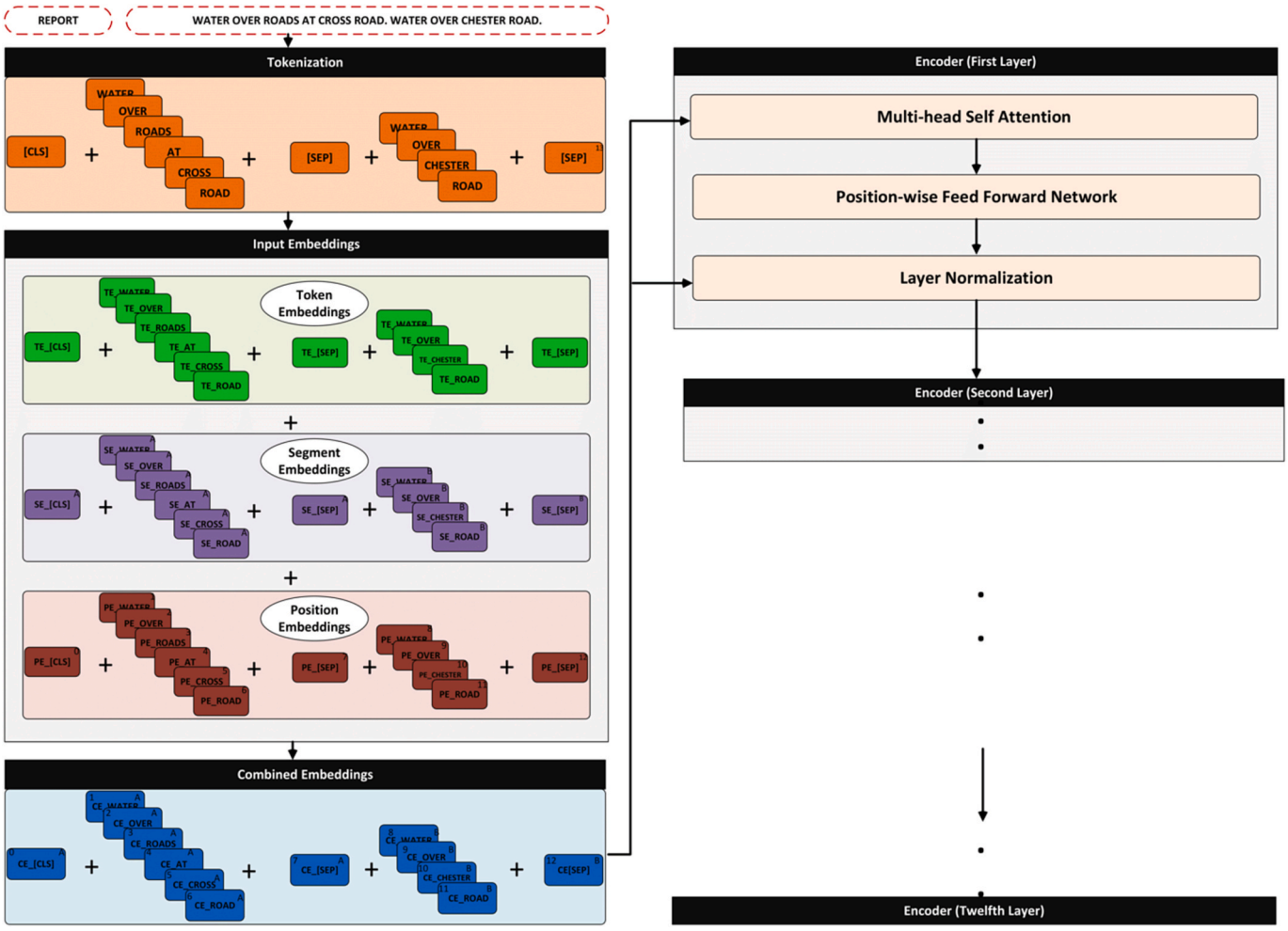


Fig. 1. Overview of the BERT model architecture. The key components of the BERT model, including Tokenization, Input Embeddings (Token Embeddings, Segment Embeddings, Position Embeddings), combined Embeddings and Transformer Encoder Layers (Multi-Head Self-Attention, Position-Wise Feed-Forward Network, Layer Normalization).

head to independently consider different perspectives on the entire sequence. Self-attention is performed in parallel across multiple heads. The results from each head are concatenated and then linearly transformed using the weight matrix W_0, as shown in Equation (2):

$$Concat\ (head_1, \ldots, head_n)\ W^0 = Attention\ \left(QW_i^Q, KW_i^K, VW_i^V\right) \quad \text{\{}\text{Equation}\} \ 2)$$

Where W^Q is the weight matrix, W^K shows the key weight matrix, and W^V denotes the value weight matrix. These weights are distinct for each head.

The encoder also incorporates a position-wise feedforward network (FFN) to enhance the representation of each token. The FFN consists of two linear transformations with a Rectified Linear Unit (ReLU) activation function between them. This structure enables the model to learn complex (i.e., deep contextual) representation by transforming the attended features of each token, as shown in Equation (3):

$$y_i = ReLU((x_i W_1 + b_1)W_2 + b_2) \quad \text{\{}\text{Equation}\} \ 3)$$

Where x_i denotes the input to this sub-layer, the output is given by y_i, W_1, and b_1 (W_2 and b_2) are the weight matrix, bias term of the first (second) linear transformation, respectively.

Layer normalization is essential for achieving stable and efficient training. It helps prevent gradients from vanishing or exploding, which can hinder convergence. Moreover, residual connections are used to add the input of a layer back to its output to learn more complex functions by building upon simpler ones, leading to better performance. The google-BERT/BERT-base-uncased model used in this study comprises 12 encoder blocks, each generating an output representation. The maximum input sequence length is 512 tokens with a 768-dimensional embedding representing each token.

2.1.2. BART model

The BART is a powerful NLP-based sequence-to-sequence model that was developed by Facebook (Lewis et al., 2019). BART combines the strengths of both BERT and Generative Pre-trained Transformer (GPT), utilizing bidirectional encoding from BERT and autoregressive decoding from GPT. This integration approach enables BART to excel in both classification tasks, such as natural language inference (NLI), and text generation tasks (Du et al., 2024; Rahman et al., 2024).

During the pre-training phase, BART introduces denoising (i.e., deletions, random replacements, insertions, and word shuffling) to the original text to create corrupted inputs. The model is then trained to reconstruct the original text from these corrupted inputs, enhancing its ability to handle noisy and incomplete data. When BART takes the text as sequence-to-sequence data, it appends the special tokens [<s>] and [</s>] to indicate the start and end of the sequence. Then, the sequences are subjected to token embeddings based on Byte Pair Encoding (BPE) and position embeddings (Balde et al., 2024). BPE merges tokens based on their frequency of occurrence, whereas WordPiece prioritizes

merges based on mutual information. BART treats the input as one continuous segment, without using segment embeddings.

BART's encoder employs unmasked (i.e., bidirectional) self-attention similar to BERT, allowing each token to attend to all other tokens in the input sequence. In contrast, its decoder utilizes masked self-attention to ensure that each token only attends to previous tokens in the output sequence, maintaining the autoregressive property during text generation.

The model uses the activation function of the Gaussian Error Linear Unit (GeLU; Equation (4)), which is defined as

$$GeLU(x) = x \times \Phi(x) \quad \text{\{}\text{Equation}\text{\}} 4)$$

Where $\Phi(x)$ represents the cumulative distribution function of the standard normal distribution.

BART's encoder enables comprehensive contextual understanding by using the input sequence bidirectionally. This bidirectional encoding allows the model to capture dependencies between tokens. The transformer's architecture revolutionized a paradigm shift in NLP by leveraging scalability and parallelization (Wang et al., 2019). Their unprecedented ability to model long-range dependencies has enabled advanced contextual comprehension and context-aware language generation. Conversely, the decoder employs an autoregressive approach to generate the output sequence. The decoder predicts each subsequent token based on the encoder's generated hidden states and the preceding tokens in the output sequence, ensuring coherent and contextually relevant text generation.

The larger variant, Facebook/BART-large, extends the architecture to 12 encoder layers and 12 decoder layers with a hidden size of 1024. Another variant of BART is Facebook/BART-large- MNLI, which has been trained on the Multi-Genre Natural Language Inference (MNLI) dataset. The maximum input sequence length for both models is 1024 tokens, allowing them to handle relatively long pieces of text, roughly equivalent to 768 words.

The BART encoder processes input weather text reports to provide rich, bidirectional contextual embeddings. While it is not used for text generation in this study, the final hidden state of the last decoder token represents the input sequence. A linear classification head is added on top of the encoder's output to map these embeddings to predefined event categories.

2.1.3. LLaMA model

LLaMA is an advanced GPT model that contains several architectural innovations to enhance the performance and scalability of the data (Touvron et al., 2023). LLaMA receives a sequence of tokens (i.e., words or sub-word units) as input. Each token is converted into a numerical vector (embedding) based on the BPE tokenization technique. As the model processes the input sequence sequentially, it requires knowledge of the positions of individual tokens. It employs Rotary Positional Embeddings (RoPE), which improve the encoding of positional information in the embedding space. This method improves long-range dependency handling and allows for the effective processing of sequential data.

The core of LLaMA is a stack of Transformer blocks. The self-attention block allows the model to weigh the importance of different parts of the input sequence when processing a specific token. Then, it uses Grouped Query Attention (GQA) to optimize the multi-head attention mechanism (Ainslie et al., 2023). It reduces computational complexity while effectively managing larger context windows (e.g., 4096 tokens in LLaMA-2). This enhancement enables improved performance on tasks requiring extensive textual input, such as document analysis, conversation summarization, and long-form content generation. A simple neural network that applies a non-linear transformation to each token's representation. For normalization, the LLaMA model employs RMSNorm, which enhances training stability through rescaling invariance and implicit learning rate adaptation, thereby contributing to more robust model convergence.

After the input passes through the Transformer blocks, LLaMA generates a sequence of output representations. These representations are then processed by a linear layer followed by a Softmax function to produce a probability distribution over the vocabulary. The model also uses a non-linear activation function called the Switchable Gated Linear Unit (SwiGLU), a variant of the GLU; see Fig. 2). LLaMA is a decoder-only Transformer optimized for generative tasks and is similar in architecture to GPT models. This study compares the performance of LLaMA-2 and LLaMA-3 models. For LLaMA-2, we utilized the 7B and 13B parameter versions, comprising 32 and 40 Transformer layers, respectively (Touvron et al., 2023). For LLaMA-3 and LLaMA-3.1, the 8B models were used, each with 32 layers (Dubey et al., 2024).

A classification head is added on top of the decoder of the LLaMA model architecture, adapting it for classification tasks. This allows the model to learn from input sequences and classify them into predefined categories, leveraging a rich contextual embedding generation.

2.2. Model variants and fine-tuning

We used seven pre-trained models, including (i) Facebook/BART-large (model-1), (ii) Facebook/BART-large- MNLI (model-2), (iii) Google-BERT/BERT-base-uncased (model-3), (iv) Meta-LLaMA/LLaMA-2-7b-hf (model-4), (v) Meta-LLaMA/LLaMA-2-13b-hf (model-5), (vi) Meta-LLaMA/LLaMA-3-8b-hf (model-6), and Meta-LLaMA/LLaMA-3.1-8b-hf (model-7) through the Hugging Face platform for multi-label text classification. These models encompass a range of sizes, with parameter counts of 406 million, 406 million, 110 million, 7 billion, 13 billion, 8 billion, and 8 billion parameters, respectively. The selection of models was based on the need to ensure comparability with previous studies and to evaluate performance under resource-constrained conditions. BART and BERT were included due to their widespread applications in prior studies, enabling direct comparison of results. For the LLaMA family, we selected LLaMA-2 models with mid-range parameter sizes (e.g., 7B, 13B), known for their efficiency and competitive performance despite their smaller size, making them suitable for computational tasks with limited GPU/CPU resources. Additionally, LLaMA-3 models were incorporated as they represent the latest advancements in the LLaMA series, offering optimized performance and efficiency, ideal for evaluating the trade-offs between model size and real-world applicability. One of the most significant advantages of using these pre-trained models is their extensive pre-training on vast datasets, which enhances their performance across various NLP tasks. These models have been pre-trained on trillions of tokens. For instance, the LLaMA models pre-trained on 1.8 trillion tokens required substantial training time. The LLaMA-2-7B-hf model took 184,320 h, while the LLaMA-2-13B-hf model required 368,640 h (Touvron et al., 2023). The LLaMA-3.1-8B-hf model was also trained on over 15 trillion tokens using a custom-built GPU cluster, consuming 1.46 million compute hours (Dubey et al., 2024). In this study, we leveraged these existing pre-trained models to analyze flood report text data for weather classification. To effectively fine-tune these models while addressing memory constraints and mitigating overconfidence in LLMs, this study employed the parameter-efficient fine-tuning technique, Low-Rank Adaptation (LoRA; Hu et al., 2021). This method kept the original model's parameters frozen and reduced memory consumption. Using LoRA approach we then inserted small rank decomposition matrices into each model. These matrices were computationally efficient to train and drastically reduced the number of trainable parameters. For example, this method reduced the trainable parameters of a 70 billion parameter model to approximately 131 million. The optimized values of parameters used for training each model are provided in Table 1. Also, the pseudocode for disaster event classification provided in Fig. 3 outlines the key steps involved in developing the code for the entire text mining process.

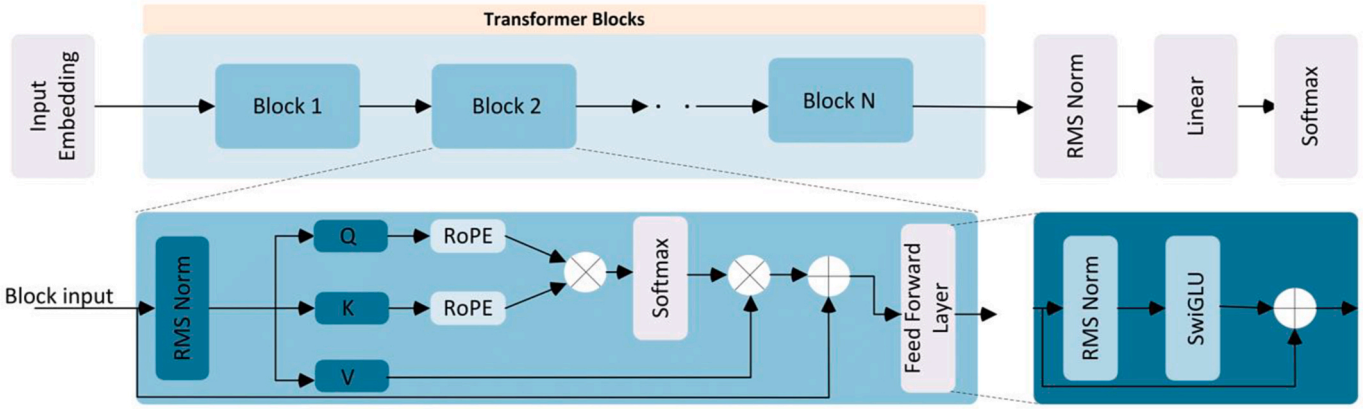


Fig. 2. Overview of the LLaMA model architecture developed in this research. The key components of the LLaMA model, including Input Embeddings, Transformer Blocks (RMS Norm, Self-Attention Mechanism with Query, Key, and Value matrices, Rotary Positional Embedding, Softmax, Feed Forward Layer, RMS Norm, SwiGLU), and Output Layers (RMS Norm, Linear Layer, Softmax).

Table 1
General parameters for fine-tuning configuration.

Parameter	Value	Min Value	Max Value
LoRA r	8	4	64
LoRA alpha	32	8	128
LoRA dropout	0.1	0.0	0.5
Learning rate	1e-5	1e-6	1e-3
Loss function	Cross-entropy	–	–
Optimizer	AdamW	–	–

2.3. Study area and dataset processing

We selected Charleston County, SC (USA), due to its susceptibility to flooding caused by hurricanes and tropical storms (see Fig. 4). In this region, weather extremes can cause precipitation and wind shear and pressure effects offshore, resulting in the co-occurrence of high astronomical tides, tropical cyclones, storm surges, and coastal flooding simultaneously. This compound effect poses significant risks to the region's infrastructure, economy, and public safety (Phillips et al., 2022; Terlinden-Ruhl et al., 2024). Analysis of the Federal Emergency Management Agency's (FEMA) National Risk Index (FEMA, 2023) revealed that Charleston County is categorized as a relatively high-risk area due to the confluence of several factors including extensive urbanization, population growth, and proximity to the coast.

Since 2005 Charleston has experienced 40 distinct major extreme events based on NWS flood reports. These categories are classified into five major categories: floods, cyclonic events, thunderstorm-related phenomena, non-thunderstorm winds, and lightning/hail after the removal of irrelevant categories. These categories were used to train the models to identify key themes within a text and then extract relevant keywords that best represent these categories.

2.4. Evaluation criteria

In this research, we tackled a multi-label classification challenge, emphasizing not only the model's overall effectiveness but also its precision in identifying each specific label. We employed the accuracy (Equation (5)), precision (Equation (6)), recall (Equation (7)), and F1 (Equation (8)) scores as our evaluation metrics to assess the model's ability to classify each individual label accurately.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \{\}\text{Equation} 5)$$

$$Precision = \frac{TP}{TP + FP} \quad \{\}\text{Equation} 6)$$

$$Recall = \frac{TP}{TP + FN} \quad \{\}\text{Equation} 7)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \{\}\text{Equation} 8)$$

For each model and each distinct category, the True Positive (TP) is the count of paragraphs accurately identified within that category. The False Positive (FP) is the number of paragraphs incorrectly assigned to the category, and the False Negative (FN) is the number of paragraphs that should have been included in the category but were not. Additionally, the True Negative (TN) represents the number of paragraphs correctly identified as not belonging to the category. These metrics are scaled between 0 and 1, where higher values indicate better model performance.

The F1 score balances precision and recall, providing a metric that reflects the accuracy and completeness of the model's predictions. A higher F1 score signifies that the model effectively identifies positive cases while minimizing false positives and false negatives. Therefore, in this study, we set a threshold of 0.8 for the F1 score as a good performance to ensure balancing the trade-off between precision and recall (Sokolova and Lapalme, 2009). This threshold provides high classification accuracy while optimizing computational efficiency.

We applied the percentile bootstrap approach to the test set predictions to quantify the reliability of our different models' performance estimates (Efron and Tibshirani, 1993; Slaets et al., 2017). A key advantage of this approach is that it does not rely on strong distributional assumptions about the data, making it robust and widely applicable for evaluating model performance under uncertainty. In each model we generated bootstrap samples (BS) by randomly sampling the test indices with replacement. We recalculated the chosen performance metrics (accuracy, precision, recall, and F1-score) for each resampled dataset (Equation (9)).

$$\{\hat{x}_T^{(1)}, \hat{x}_T^{(2)}, \ldots, \hat{x}_T^{(BS)}\} \quad \{\}\text{Equation} 9)$$

After sorting the set of bootstrap estimates, then for a significance level of α the confidence interval (CI) for $\hat{x}_T$ is given (Equations (10)–(12)).

$$CI = \left[\hat{x}_T^{(j)}, \hat{x}_T^{(k)}\right] \quad \{\}\text{Equation} 10)$$

$$j = \left[\frac{\alpha}{2}(BS+1)\right] \quad \{\}\text{Equation} 11)$$

$$k = \left[\left(1 - \frac{\alpha}{2}\right)(BS+1)\right] \quad \{\}\text{Equation} 12)$$

```
Data Preprocessing
BEGIN
    - LOAD text data
    - REMOVE rows with missing text
    - FILTER out unwanted event types
    - MAP event types into disaster-related categories
    - CONVERT category labels into numeric IDs
    - CREATE dataset from columns "text" and "label"
Model Training Setup
    - INITIALIZE pretrained model name (e.g., LLaMA-2)
    - INITIALIZE tokenizer for chosen model
    - DEFINE tokenization process (truncate and pad to fixed length)
    - APPLY tokenization to all text entries
    - CONFIGURE LoRA parameters: r, alpha, dropout
    - DEFINE function compute metrics:
        CALCULATE F1, accuracy, precision, and recall
        RETURN computed metrics
Model Training and Evaluation
    - FOR each epoch_count in epoch_values:
        INITIALIZE model with a given configuration
        APPLY LoRA adapters to model
        SET training parameters
        TRAIN model for epoch_count epochs
        EVALUATE model on test dataset
        LOG results and SAVE evaluation outputs
    END FOR
END
```

Fig. 3. Pseudocode for disaster event classification including data preprocessing, model training, and evaluation.

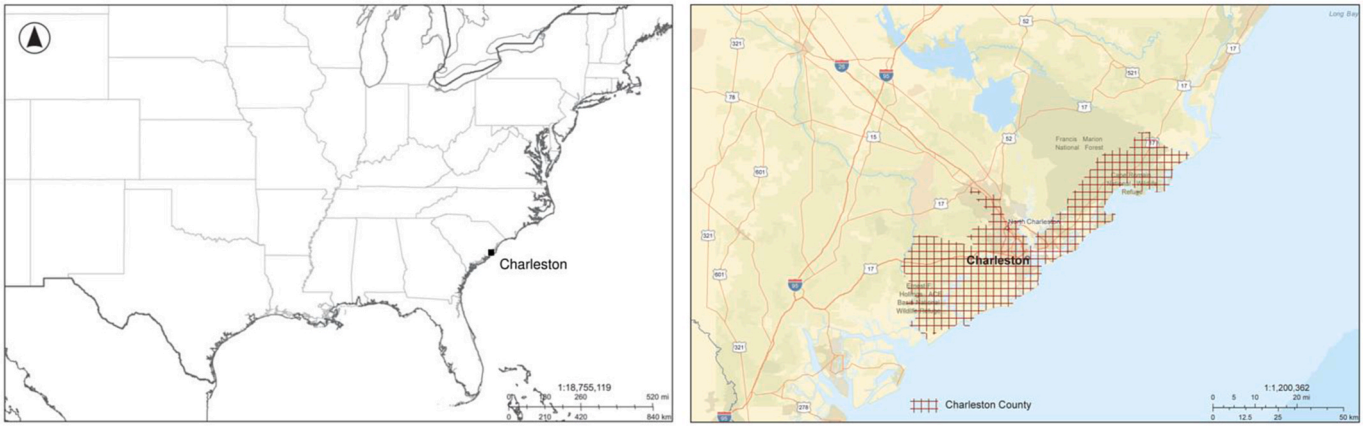


Fig. 4. The Charleston County (shaded area) is located in the coastal plain region of SC, USA.

We defined the chance level (xc) to assess whether the model's accuracy was significantly better than random chance for a multi-label classification. In the simplest balanced case with K classes, this can be taken as $x_c = 1/K$. We then calculated the one-sided p-value by determining the fraction of bootstrap estimates that fall at or below the chance-level threshold xc (Equation (13)).

$$p - value = \frac{1}{BS}\sum_{i=1}^{BS} I\left(\hat{x}_T^{(i)} \le x_c\right) \quad \{\text{Equation}\}\ 13)$$

where I(·) is the indicator function, which equals 1 when its argument is true and 0 otherwise.

3. Results

3.1. Category balancing

In this study, categories such as floods, cyclonic events, and thunderstorm-related phenomena were considered, as they posed risks

of infrastructure damage and economic loss in Charleston County. We utilized flood reports from the National Weather Service (NWS), with the repository based on the local report products. These reports include relevant labels and detailed remarks associated with various weather events.

The dataset contains 6086 paragraphs. The dataset underwent a four-step cleaning process to standardize text data for analysis. This involved handling expanding contractions and removing unnecessary punctuation, special characters, and lowercase text. During this process, approximately 5% of the dataset was excluded. These exclusions were primarily due to paragraphs that contained insufficient information, excessive noise, or errors that could not be resolved through preprocessing.

We found significant uneven data across the three major categories during data preprocessing. We used a normalized count metric for each category to measure this imbalance in our multi-label text classification. The normalized counts for each category were as follows: "Flood" (0.758), "Thunderstorm" (0.178), and "Cyclonic" (0.064) categories. Ideally, each category would have a normalized count near 0.33 in a balanced dataset among the three categories.

We addressed the imbalance issue of the dataset using the MLSOL (see Liu and Tsoumakas, 2020). This technique generates diverse, well-labeled synthetic samples by leveraging local label distributions, creating a more balanced dataset. After applying MLSOL, the normalized count metric for each label significantly improved, resulting in a more balanced distribution of "Flood" (0.355), "Thunderstorm" (0.326), and "Cyclonic" (0.319) categories.

3.2. Sensitivity analysis

3.2.1. Dataset size

One objective of this study was to assess the optimal dataset size, as running these large models required considerable time and computational resources. Fig. 5 illustrates the F1 validation scores of different LLMs across three event categories, which behave differently across varying dataset sizes. We split the dataset into ten increments, where one represents the smallest subset and ten represents the entire dataset. The optimal balanced dataset size refers to the point at which the dataset contains sufficient samples for each of the three categories such that their F1 scores are maximized. At this size, increasing the dataset further no longer leads to improved performance for these categories.

In the BART-large model, the "Cyclonic" category showed slight improvement as the dataset size increased, while "Flood" and "Thunderstorm" demonstrated more noticeable improvements. The optimal performance across all three categories was achieved at approximately 90% of the dataset, where the highest values were observed. The BART-large-MNLI model exhibited an overall increasing trend across all categories despite some variation. The "Flood" and "Thunderstorm" categories showed steady improvement as the dataset size increased, while the "Cyclonic" category exhibited a slower improvement trend. Additionally, a notable decline in "Thunderstorm" and "Cyclonic" performance was observed beyond 80% of the dataset size. The optimal performance of this model was achieved at around 80% of the dataset.

The BERT-base-uncased model showed a fluctuating trend in performance metrics for the "Cyclonic" category between 20% and 60% of dataset size, while a consistent improvement was observed for both "Thunderstorm" and "Flood" categories as the training data size increased. It is interesting to note that the "Flood" category

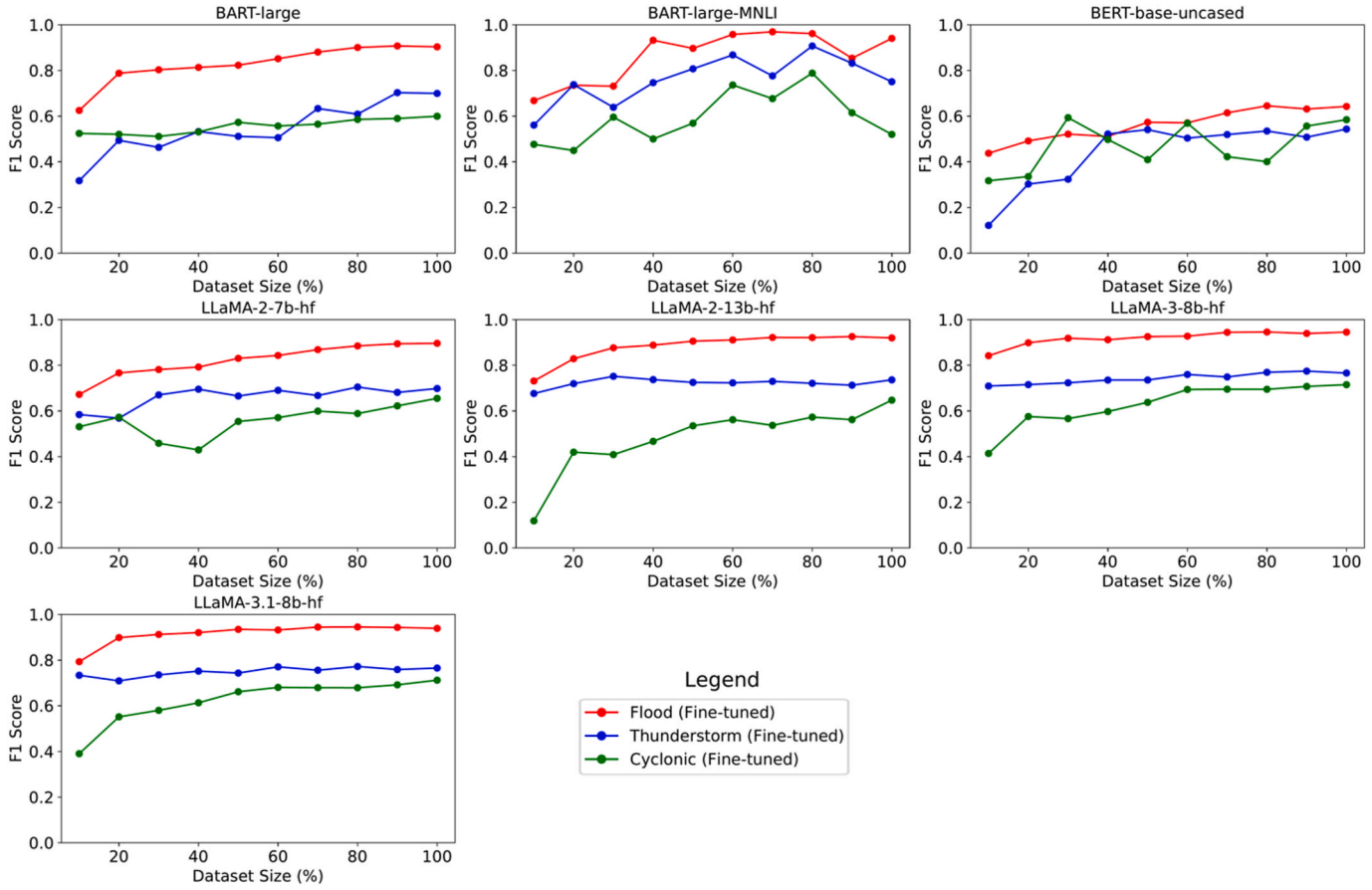


Fig. 5. Validation performance comparison of multiple LLMs across varying dataset sizes on "Flood", "Thunderstorm", and "Cyclonic" categories, based on F1 score values.

demonstrated steady improvement with a noticeable rise between 40% and 80% of the dataset size. For example, the F1 score of the "Thunderstorm" category increased from 0.335 to 0.584 as the dataset size expanded from 20% to 80%. The performance was converged for all categories at around 80%–100% of the dataset size.

For the LLaMA-2-7B-hf model, the "Flood" category showed steady improvement, stabilizing as the dataset size increased. In addition, the "Thunderstorm" category demonstrated gradual and consistent improvement while the "Cyclonic" category exhibited significant fluctuations between 20% and 60% of dataset size with slight improvement beyond 60%. The LLaMA-2-13B-hf training results indicated that the "Flood" category exhibited a steady upward trend. The "Thunderstorm" category demonstrated gradual and consistent improvement with dataset size. Unlike the "Thunderstorm" category, the "Cyclonic" category showed the steepest improvement between 10 and 60% followed by a slower but steady rise. A similar trend was also observed in the LLaMA-3-8B-hf and LLaMA-3.1-8B-hf models. The key difference was that the optimal dataset size for the LLaMA-2 model was around 80–90% whereas an optimal performance was achieved at approximately 60% of the dataset size for the LLaMA-3 model. This indicated the fact that the LLaMA-3 model can learn effectively from smaller amounts of data, showcasing higher performance even with smaller dataset sizes to reach optimal performance.

Several noticeable observations can be drawn from these results. Notably, model performance improved significantly with increasing the dataset size; however, the improvements tended to be insignificant after reaching a certain dataset size (e.g., 2768 paragraphs of text data per category for the LLaMA-3 model). In other words, each additional part of the data contributed less and less to the overall model accuracy. Additionally, the BERT and BART models exhibited more fluctuation in performance across different dataset sizes compared to the larger LLaMA-2 and LLaMA-3 models, which display more stable performance trends. The optimal dataset sizes for BART-large, BERT, and LLaMA-2 models ranged between 80 and 100%. In contrast, the LLaMA-3 model achieved a peak performance of around 60% of the dataset size. This emphasizes the importance of identifying optimal dataset sizes to balance performance and computational efficiency.

Across different categories, the "Flood" category showed a steady upward trend in the BERT and LLaMA models, while it varied significantly in the BART model. "Thunderstorm" category was less sensitive to dataset size using the LLaMA models but showed greater sensitivity in the BERT and BART models. For the "Cyclonic" category, the BART-large model remained stable while the BART-large-MNLI and BERT models showed fluctuations. The LLaMA model demonstrated a consistent increase concerning F1 score performance.

3.2.2. *Epoch size*

Fig. 6 illustrates the impact of epoch size on the performance of both the fine-tuned and the few-shot (5-shot) models. In the few-shot fine-tuning, these examples were used exclusively by the models to interpret the labels. We considered only up to ten epochs, as running these large models beyond this point was computationally costly.

In the fine-tuned BART-large model, the "Flood" and "Cyclonic" categories exhibited slight improvements in performance during the later epochs. In contrast, the "Thunderstorm" category demonstrated noticeable gains at an earlier epoch training stage. While few-shot learning exhibited instability and underperformance compared to fine-tuning, especially for more complex events such as "Cyclonic", there is no specific epoch where all three categories converged in performance. For example, after four epochs, the performance for the "Flood" and

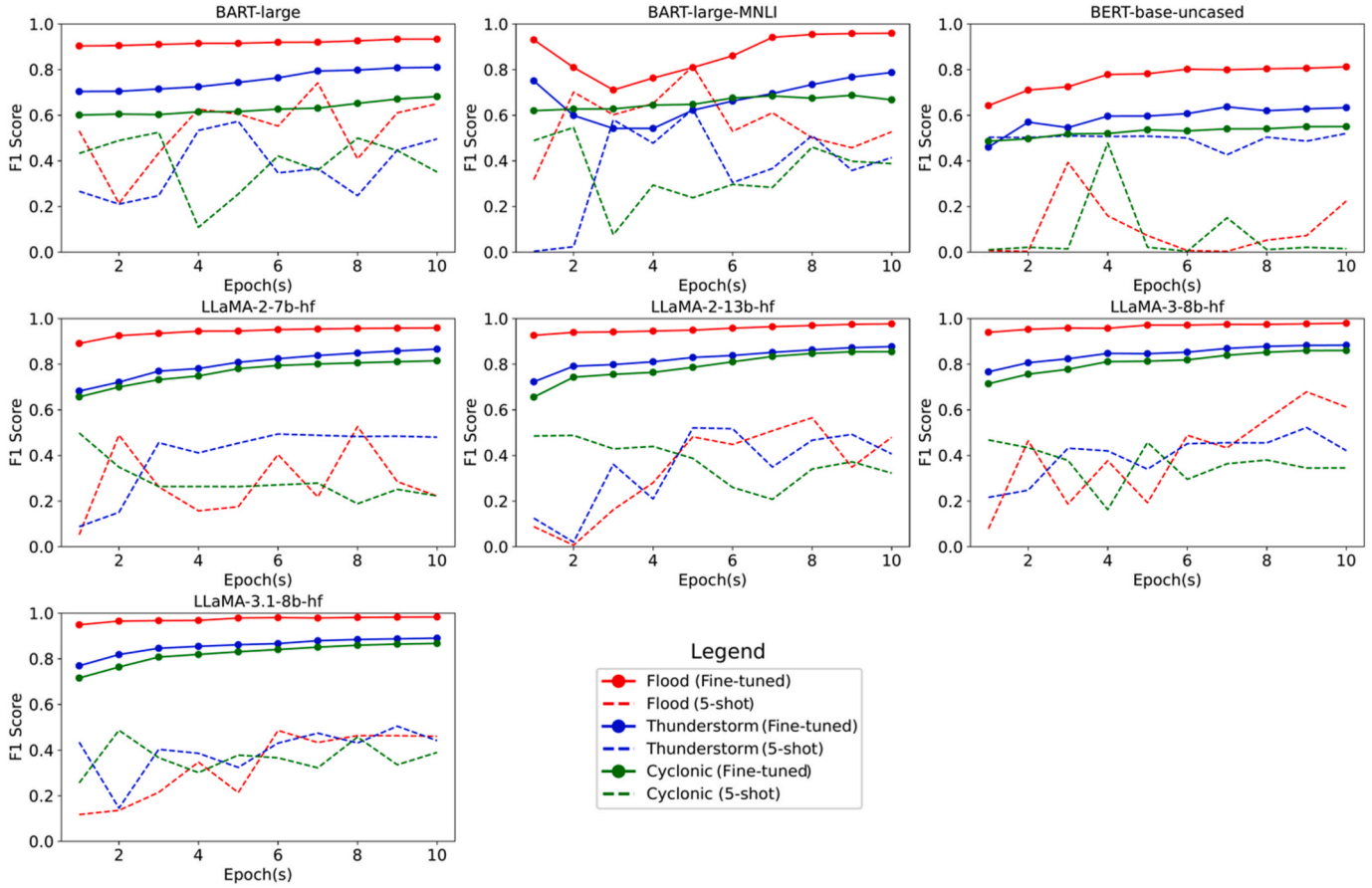


Fig. 6. Performance comparison of various LLMs across different epoch sizes. As shown fine-tuned models presented better and more consistent performance, while few-shot models exhibited more fluctuation and lower performance.

"Thunderstorm" categories was high, whereas the performance for "Cyclonic" remained low. Moreover, there was no specific epoch where all three categories converged or achieved comparable performance levels.

The fine-tuned BART-large-MNLI model exhibited a distinct performance trend, characterized by initial fluctuations across the "Flood" and "Thunderstorm" categories. During the middle epochs, the "Cyclonic" category temporarily outperformed the "Thunderstorm" category. However, this trend changed in the later epochs, with "Thunderstorm" performance surpassing "Cyclonic". This behavior may be attributed to the model's prior fine-tuning on a specific task, which created adaptation challenges when applied to a different domain. Few-shot learning showed higher variability, particularly for the "Thunderstorm" and "Cyclonic" categories. This highlights the BART-large-MNLI limitations when dealing with more complex categories; however, the performances seem to converge in the later epoch.

In the fine-tuned BERT-base-uncased model, all categories significantly improved during the middle epochs, but learning progress plateaued after the fifth epoch. In the few-shot model, the "Thunderstorm" category demonstrated promising results across different epochs, while the "Flood" and "Cyclonic" categories struggled to simulate events, exhibiting limited performance effectively.

For the fine-tuned LLaMA-2 models, increasing epochs significantly affected the "Thunderstorm" and "Cyclonic" categories. At the same time, the performance of the "Flood" category remained largely unchanged and high after the middle epochs. Among the evaluated models, the LLaMA-2-13B-hf consistently outperformed the LLaMA-2-7B-hf. Notably, for LLaMA-2-13B-hf, the performance of the "Thunderstorm" and "Cyclonic" categories stabilized after seven epochs, whereas, for LLaMA-2-7B-hf, the "Thunderstorm" category continued to improve beyond this point. Moreover, the performance of the few-shot models fluctuated widely across epochs, exhibiting an unstable trend and struggling to generalize well to the classification tasks for complex event categories, such as "Cyclonic" and "Thunderstorm,". Increasing the number of training epochs for the few-shot LLaMA-2-7B-hf model led to improved performance on the "Thunderstorm" classification task. However, this increase in epochs resulted in deteriorated performance in the "Cyclonic" category and caused fluctuations in the performance regarding the "Flood" category. Conversely, with the few-shot LLaMA-2-13B-hf model, the performance on both the "Thunderstorm" and "Flood" categories improved, whereas the performance on the "Cyclonic" category decreased.

For the fine-tuned LLaMA-3 models (i.e., 3.8B-hf and 3.1-8B-hf), increasing the number of epochs had a significant impact on the "Thunderstorm" and "Cyclonic" categories during the early training stages. In contrast, the performance of the "Flood" category remained consistently high and largely unaffected after the initial epochs. This highlights the efficiency and superior capability of the LLaMA-3 models compared to LLaMA-2 models (i.e., 2.7B-hf and 13B-hf), particularly in achieving robust performance in fewer epochs. From a few-shot view, both the LLaMA-3.8B and LLaMA-3.1–8B models demonstrated improvement in the "Flood" and "Thunderstorm" categories, while the performance for the "Cyclonic" category remained unchanged.

Overall, in the fine-tuned models, the BART-large and BART-large-MNLI models showed improvement in the later epochs while the BERT and LLaMA-2 models performed the best in the middle epochs, and the LLaMA-3 model excelled in the early epochs. This reflects the fact that beyond these optimal periods, additional training resulted in a negligible performance improvement. The F1 scores for the "Flood", "Thunderstorm", and "Cyclonic" categories generally improved as the number of epochs increased, except for the BART-large-MNLI model, which exhibited a decline in performance during the middle epochs.

In the evaluation of few-shot models, BART (both large and large-MNLI variants) exhibited inconsistent performance across different epochs. In contrast, the BERT model demonstrated limited capability and simulated only the "Thunderstorm" category well. Notably, the LLaMA-2 and LLaMA-3 models showed distinct trends. While both improved in simulating "Flood" and "Thunderstorm" as training epochs increased, LLaMA-2 experienced a decline in accuracy for "Cyclonic" simulations over time, whereas LLaMA-3 maintained a stable performance for this category. This indicates that these models rely heavily on their parameters, demonstrating how fine-tuning impacts the training process and is essential for model convergence.

3.3. Performance analysis

Table 2 presents the results of different models on the imbalanced dataset. In the "Flood" category, all models exhibited high precision, recall, F1 score, and accuracy values. The LLaMA-3-8b-hf model achieved the highest metrics with a precision of 0.989, recall of 0.997, F1 score of 0.992, and accuracy of 0.97. On the other hand, the BERT model showed the lowest among the models but still maintained decent high

Table 2
Evaluation of fine-tuned models with the imbalance dataset after 10 epochs. The best performances of each category are bolded.

Model	Event	Precision	F1 Score	Accuracy	Recall
BART-large	Flood	0.872	0.861	0.85	0.849
	Thunderstorm	0.492	0.655	0.8	**0.979**
	Cyclonic	0.848	0.130	0.75	0.070
BART-MNLI	Flood	0.757	0.857	0.88	**0.989**
	Thunderstorm	0.534	0.488	0.7	0.449
	Cyclonic	0.222	0.257	0.65	**0.305**
BERT-base-uncased	Flood	0.657	0.792	0.881	0.998
	Thunderstorm	0.358	0.205	0.804	0.144
	Cyclonic	0.256	0.097	**0.902**	0.060
LLaMA-2-7B-hf	Flood	0.978	0.987	0.96	0.997
	Thunderstorm	0.75	0.841	0.92	0.956
	Cyclonic	0.889	0.235	0.78	0.135
LLaMA-2-13B-hf	Flood	0.976	0.986	0.96	0.996
	Thunderstorm	0.736	0.829	0.91	0.949
	Cyclonic	**0.995**	0.184	0.77	0.101
LLaMA-3-8B-hf	Flood	0.989	**0.992**	**0.97**	0.997
	Thunderstorm	**0.761**	**0.853**	**0.93**	0.971
	Cyclonic	0.75	**0.32**	0.8	0.203
LLaMA-3.1-8B-hf	Flood	**0.982**	0.989	0.965	0.995
	Thunderstorm	0.759	0.846	0.92	0.956
	Cyclonic	0.733	0.297	0.79	0.186

Table 3
Evaluation of fine-tuned and few-shot LLMs for classifying the "Flood", "Thunderstorm", and "Cyclonic" categories, achieving an f1 score of 0.8 for the first time. If the F1 score does not reach 0.8, the optimal epoch (corresponding to the best performance) is indicated in parentheses. The best performances are bolded.

Fine-Tuned Models						
Model	Event	Precision	F1 Score	Accuracy	Recall	Optimal Epoch
BART-large	Flood	0.833	0.904	0.893	0.984	1
	Thunderstorm	0.792	0.808	0.796	**0.825**	9
	Cyclonic	0.682	0.667	0.674	0.652	(10)
BART-MNLI	Flood	0.79	0.809	0.806	0.830	5
	Thunderstorm	0.802	0.787	0.791	0.772	(10)
	Cyclonic	0.703	0.687	0.692	0.672	(9)
BERT-base-uncased	Flood	0.782	0.804	0.799	0.826	8
	Thunderstorm	0.653	0.637	0.646	0.622	(7)
	Cyclonic	0.556	0.55	0.555	0.544	(10)
LLaMA-2-7B-hf	Flood	0.826	0.891	0.881	0.967	**1**
	Thunderstorm	**0.834**	0.809	0.813	0.784	5
	Cyclonic	0.816	0.801	0.805	0.787	7
LLaMA-2-13B-hf	Flood	0.868	0.927	0.921	0.995	**1**
	Thunderstorm	0.815	0.811	0.812	0.807	4
	Cyclonic	0.824	0.811	0.814	0.797	6
LLaMA-3-8B-hf	Flood	**0.894**	**0.94**	**0.936**	0.991	**1**
	Thunderstorm	0.817	0.807	0.808	0.796	**2**
	Cyclonic	**0.83**	**0.811**	**0.816**	0.794	4
LLaMA-3.1-8B-hf	Flood	0.89	0.939	0.935	**0.995**	**1**
	Thunderstorm	0.822	**0.819**	**0.819**	0.815	**2**
	Cyclonic	0.817	0.808	0.810	0.8	**3**
Few-Shot Models						
Model	**Event**	**Precision**	**F1 Score**	**Accuracy**	**Recall**	**Optimal Epoch**
BART-large	Flood	**0.884**	0.742	0.591	0.639	(7)
	Thunderstorm	0.442	0.573	0.403	0.817	(5)
	Cyclonic	0.357	0.525	0.357	**0.993**	(3)
BART-MNLI	Flood	0.852	**0.815**	**0.665**	**0.781**	(5)
	Thunderstorm	0.510	**0.626**	**0.458**	0.772	(5)
	Cyclonic	**0.4**	**0.546**	**0.381**	0.861	(2)
BERT-base-uncased	Flood	0.495	0.393	0.243	0.326	**(3)**
	Thunderstorm	0.355	0.520	0.352	**0.972**	(10)
	Cyclonic	0.323	0.478	0.315	0.919	(4)
LLaMA-2-7B-hf	Flood	0.608	0.527	0.357	0.465	(8)
	Thunderstorm	0.383	0.494	0.328	0.697	(6)
	Cyclonic	0.338	0.498	0.332	0.949	**(1)**
LLaMA-2-13B-hf	Flood	0.522	0.565	0.409	0.617	(8)
	Thunderstorm	**0.516**	0.521	0.353	0.526	**(5)**
	Cyclonic	0.324	0.488	0.322	0.989	(2)
LLaMA-3-8B-hf	Flood	0.705	0.679	0.506	0.655	(9)
	Thunderstorm	0.473	0.523	0.354	0.584	(9)
	Cyclonic	0.326	0.468	0.306	0.830	**(1)**
LLaMA-3.1-8B-hf	Flood	0.641	0.486	0.321	0.391	(6)
	Thunderstorm	0.444	0.505	0.337	0.585	(9)
	Cyclonic	0.330	0.487	0.322	0.926	(2)

metrics with an F1 score of 0.792 and an accuracy of 0.881. The "Flood" category was the majority category in the imbalanced dataset, which led to a higher performance and attributed to more training examples.

In the "Thunderstorm" category, despite the smaller sample size compared to "Flood", BART-large, LLaMA-2, and LLaMA-3 demonstrated better performance. For example, the LLaMA-3-8B-hf and LLaMA-3.1-8B-hf showed the strongest performance with F1 scores around 0.85 and accuracy above 0.92. However, the BART-MNLI and BERT models exhibited unsatisfactory performance for the "Thunderstorm" category.

In the "Cyclonic" category, all models demonstrated very low recall values (ranging from 0.06 to 0.305), indicating that they disregarded a significant number of flood text data. This suggests the fact that the "Cyclonic" category is reported less in the NWS reports. The high precision with low recall implies that when the model predicts a "Cyclonic" category, it is usually correct, although the model is not very precise in detecting many "Cyclonic" categories in the flood report data.

Fig. 6 shows the convergence trends of the models' F1 scores as they stabilize with increasing epochs. Furthermore, to facilitate quantitative analysis, Table 3 presents the optimal number of epochs required for each fine-tuned model to achieve F1 scores equal to or greater than 0.8. We demonstrated that an optimal performance can be achieved with lower computational costs by identifying the minimal epochs needed to reach this performance threshold. For models that showed unsatisfactory, the values in parentheses indicate the optimal performance values.

The "Flood" category in fine-tuned models demonstrated rapid learning in both the LLaMA and BART-large models, achieving F1 scores >0.8 even during early training epochs. The LLaMA-3-8B-hf and LLaMA-3.1-8B-hf models attained 0.936 and 0.935 accuracy with corresponding F1 scores of 0.940 and 0.939, respectively. The recall scores for these top-performing LLaMA models are high (above 0.99), indicating that most "Flood" related text data are correctly identified. In comparison,

the LLaMA-2 models, including the LLaMA-2-7B-hf and LLaMA-2-13B-hf, yielded F1 scores of 0.891 and 0.921, respectively with accuracy levels of 0.881 and 0.921 and recall values of 0.967 and 0.993. While the BART-large model outperformed the LLaMA-2-7B-hf in the first epoch, attaining an F1 score of 0.904 and an accuracy of 0.893, the LLaMA-2-7B-hf exhibited superior performance for the "Flood" category as the model progressed (illustrated in Fig. 6). Meanwhile, the BERT and BART-large-MNLI models exhibited lower performance, reaching the threshold only after 5 and 8 epochs with lower performance, respectively.

In the classification of "Thunderstorm", the fine-tuned models exhibited their optimal performance at different training epochs. The LLaMA-3 models demonstrated strong performance early, achieving accuracy values of 0.808 and 0.819 at the second epoch. In contrast, the LLaMA-2 models, the LLaMA-2-7B-hf and the LLaMA-2-13B-hf, required five and four epochs, respectively, to reach comparable accuracy levels of 0.813 and 0.812. The BART-large model showed slower improvement, achieving a similar accuracy range only after nine epochs, with an overall lower performance. Additionally, the BERT and BART-large-MNLI models failed to surpass an F1 score of 0.8, showing maximum values of 0.637 and 0.787, respectively.

Among the three categories evaluated in fine-tuned models, the "Cyclonic" category demonstrated the lowest performance that required higher computational cost compared to the others. In the LLaMA-3 models, the 3-8b-hf variant achieved an accuracy of 0.81, a recall of 0.8, and a precision of 0.817 after three epochs, which are considered early-stage epochs. Similarly, the 3.1-8b-hf variant reached an accuracy of 0.816, recall of 0.794, and precision of 0.830 after four epochs. Conversely, the LLaMA-2 models attained optimal performance after a moderate number of epochs. The 7B-hf model achieved an accuracy of 0.805, a recall of 0.787, and a precision of 0.816 after seven epochs. Likewise, the 13B-hf model reached an accuracy of 0.814, a recall of 0.797, and a precision of 0.824 after seven epochs. In contrast, the BART and BERT models achieved the lowest performance for the "Cyclonic" category compared to other models with an F1 score of 0.8, and precision, recall, and accuracy values of 0.7. The low performance of the BART and BERT models is related to the fact that these models need more text data to improve their ability to understand the nuances of language and perform better on a wider range of tasks. Therefore, their performance is directly tied to the diversity and quantity of text they are trained on during the pre-training period.

Across all models, performance was superior in the "Flood" category compared to other categories. This revealed the fact that the majority of NWS flood reports were concerned about flooding impacts and consequences. "Cyclonic" category, on the other hand, yielded the lowest performance metrics across models, suggesting a potential need for more specialized training data. It should be noted that while few-shot learning enables rapid modeling deployment with limited data, few-shot learning often struggles with complex, domain-specific tasks like "Cyclonic" classification. The optimal number of epochs varies among the models, indicating different convergence rates. Among the few-shot models, the BART-MNLI model achieved the highest F1 scores and exhibited superior accuracy, precision, and recall for the "Flood", "Thunderstorm", and "Cyclonic" categories. However, the reliability of these metrics is questionable due to the fluctuating performance of the models, as illustrated in Fig. 5.

Moreover, we employed the percentile bootstrap method with 1000 iterations to calculate 95% confidence intervals for the key performance metrics across all fine-tuned models shown in Table 4. This approach resampled the dataset with replacement multiple times that provided a robust estimate of the variability for each performance metric. For instance, the LLaMA-3.1-8B-hf model consistently achieved high Precision, *F1* score, Accuracy, and Recall, with 95% confidence intervals ranging from 0.905 to 0.932, 0.902 to 0.928, 0.909 to 0.935, and 0.906 to 0.931, respectively. Furthermore, we calculated the *p*-values for Precision, *F1* Score, Accuracy, and Recall, all of which were below 0.001, confirming the statistical significance of our results.

The primary distinction in model performance arises from differences in their architecture. For example, in the BERT model, each token is assigned a unique positional vector based on its absolute position within the sequence, making the model sensitive to the exact position of each token. These positional vectors are typically added to the token embeddings before processing by the Transformer layers. In contrast, LLaMA models employed Rotary Position Embeddings (RoPE), emphasizing relative positional information. RoPE integrated positional data directly into the attention mechanism through the rotation of embeddings, allowing the model to effectively capture the distance and order between tokens and influence the computation of attention scores. RoPE encoded absolute positions using a rotation matrix while simultaneously incorporating explicit relative position dependencies into the self-attention formulation. This approach provided models with valuable properties, such as flexibility in sequence length, a natural decay of inter-token dependency with increasing relative distances, and the capability to equip linear self-attention architectures with relative position encoding. Despite the effectiveness of prior methods such as those used in models like BERT, these approaches often integrate positional information into the context representation, making them unsuitable for linear self-attention architectures. RoPE addressed this limitation by embedding positional information directly within the self-attention mechanism, thus preserving the efficiency and scalability of linear attention models.

The other main distinction lies in the model architecture. While BERT and BART typically employed feed-forward layers with dimensions ranging from 12 to 24, LLaMA models with 32–36 layers demonstrated significantly enhanced performance. This deeper architecture allows LLaMA models to more effectively capture intricate patterns within textual data. For instance, LLaMA-2-7B-hf boasts 32 layers, 4096 attention heads, a model dimension of 4096, and approximately 7 billion parameters.

3.4. Computational cost

This study employed LoRA to ensure a reliable balance between memory efficiency and fine-tuning runtime. All the models were trained and tested using a configuration consisting of three nodes with two tasks per node and two A100 GPUs allocated per node. If we consider the same computing time, the LLaMA-3.1-8B-hf and LLaMA-3-8B-hf models demonstrated a clear advantage over the LLaMA-2-13B-hf and LLaMA-2-7B-hf models, achieving performance values of 0.8 across all categories after three and four epochs, respectively. In contrast, the LLaMA-2-13B-

Table 4
Bootstrap confidence intervals (2.5%–97.5%) for Precision, *F1* Score, Accuracy, and Recall across fine-tuned models after 10 epochs.

Model	Precision	F1 Score	Accuracy	Recall
BART-large	[0.733, 0.802]	[0.764, 0.827]	[0.755, 0.821]	[0.792, 0.856]
BART-MNLI	[0.722, 0.794]	[0.721, 0.795]	[0.736, 0.794]	[0.732, 0.780]
BERT-base-uncased	[0.624, 0.671]	[0.598, 0.642]	[0.625, 0.670]	[0.616, 0.655]
LLaMA-2-7B-hf	[0.852, 0.884]	[0.872, 0.895]	[0.868, 0.898]	[0.862, 0.893]
LLaMA-2-13B-hf	[0.874, 0.903]	[0.863, 0.915]	[0.876, 0.905]	[0.863, 0.912]
LLaMA-3-8B-hf	[0.897, 0.923]	[0.895, 0.922]	[0.898, 0.925]	[0.895, 0.923]
LLaMA-3.1-8B-hf	[0.905, 0.932]	[0.902, 0.928]	[0.909, 0.935]	[0.906, 0.931]

Table 5
Comparison of fine-tuning and few-shot computing times across LLMs over ten epochs. Bolded values represented the least computational costs.

	Computing Time						
Model	**BART-large**	**BART-MNLI**	**BERT-base-uncased**	**LLaMA-2-7B-hf**	**LLaMA-2-13B-hf**	**LLaMA-3-8B-hf**	**LLaMA-3.1-8B-hf**
Fine-Tuned	12hr 52min	**11hr 19min**	12hr 17min	11hr 54min	15hr 26min	12hr 1min	15hr 59min
Few-Shot	1hr 32min	1hr 18min	**1hr 1min**	4hr 23min	7hr 51min	5hr 24min	5hr 23min

hf and LLaMA-2-7B-hf models required six and seven epochs, respectively, to reach comparable performance levels. This highlights the efficiency and superior capability of the LLaMA-3 models over the LLaMA-2 models, particularly in achieving robust performance in fewer epochs. Additionally, while the BART-large model exhibited a higher computational runtime, it outperformed both the BART-large-MNLI and BERT models, highlighting its capability to achieve superior performance despite the increased computational cost.

The model type and the number of associated parameters significantly influence computational time. The computational cost of each fine-tuned and few-shot model in a ten epochs run is presented in Table 5. In the fine-tuning setting, smaller models such as BERT-base-uncased demonstrated superior efficiency as they completed the training in 12 h and 17 min, while larger models such as LLaMA-2-13B-hf and LLaMA-3.1-8B-hf required substantially longer times. In addition, LLaMA-2-7B-hf achieved a notable balance with a fine-tuning time of 11 h and 54 min, slightly outperforming BART-large, which required 12 h and 52 min.

For few-shot models, computational times were significantly reduced compared to fine-tuned models. BERT-base-uncased demonstrated the shortest computational time at just 1 h and 1 min, closely followed by BART-MNLI at 1 h and 18 min. On the other hand, LLaMA-2-13B-hf exhibited the highest computational demand among few-shot models, requiring 7 h and 51 min. LLaMA-3.1-8B-hf and LLaMA-3-8B-hf showed moderate improvement, completing few-shot tasks in 5 h and 23 min and 5 h and 24 min, respectively. Notably, increased computational time does not necessarily correlate with improved performance in these models.

4. Conclusion

This research employed multiple LLMs to address the question "Can LLMs Effectively Reason about Adverse Weather Conditions?". To answer this question, we gathered flood report text data spanning from June 18, 2005, to September 22, 2024. We employed seven different pre-trained LLMs such as BART-large, BART-large-MNLI, BERT, LLaMA-2-7B-hf, LLaMA-2-13B-hf, LLaMA-3-8B-hf, and LLaMA-3.1-8B-hf to classify disaster-related text data based on their labels. We categorized text data into "Flood", "Thunderstorm", and "Cyclonic" categories. LLMs were implemented for Charleston County, SC where NWS issues flood reports frequently due to being a flood-prone area.

The modeling outcomes revealed that due to the random distribution of flooding events, the models performed poorly in categories with a low amount of data because of the imbalance issue. While the models achieved strong performance on the "Flood" and "Thunderstorm" categories despite the dataset imbalance, the "Cyclonic" category consistently underperformed with F1 scores not exceeding 0.32 after ten epochs. This observation aligns with previous research demonstrating that LLMs, despite their impressive capabilities, can struggle with imbalanced data (Zhu et al., 2024). Utilizing the LLaMA-3.1-8B-hf model, we observed a significant improvement in the performance of the "Cyclonic" category, achieving a 152.5% increase in the F1 score with only three epochs. This enhancement not only substantially increased the efficiency of the model for the "Cyclonic" category but also reduced computational costs due to the lower number of training epochs.

This study emphasizes the impact of dataset size on each specific category, showing that it is overly simplistic to consider only the overall performance. The performance of LLMs generally improved with larger datasets but gained a plateau after a certain size. Larger models such as LLaMA-2 and LLaMA-3 demonstrated more stable and consistent performance compared to BERT and BART, which exhibited greater variability. These findings corroborate existing research highlighting the positive correlation between model size and performance, up to a certain point, while also showing that performance levels off after a few epochs (i.e., around 10 epochs; see Kaplan et al., 2020; Wilkho et al., 2024). Optimally balanced dataset sizes typically ranged between 60 and 100% across models and categories, with LLaMA-3 reaching peak performance earlier and more effectively. In addition, we evaluated the performance of LLMs through fine-tuning and few-shot evaluations. This study used the LoRA fine-tuning method to optimize model performance while improving the efficiency of parameter updates during the training process. LoRA significantly reduced the memory requirements compared to full fine-tuning. For instance, the LLaMA-3.1-8B-hf model required approximately 16 GB of memory for fine-tuning using the LoRA method, compared to the 60 GB needed for full fine-tuning. In fine-tuned models, BART-large variants improved in later epochs, while BERT and LLaMA-2 performed best in middle epochs with LLaMA-3 excelling early. Few-shot evaluations revealed fluctuating BART performance, limited BERT capability, and distinct trends in LLaMA models where LLaMA-3 demonstrated stability across all categories while LLaMA-2 declined in "Cyclonic" accuracy over time. These results underscore the critical role of fine-tuning in optimizing the performance of LLMs. These findings align with existing literature emphasizing the importance of fine-tuning in enhancing model performance and the challenges inherent in few-shot learning (see Parthasarathy et al., 2024).

This study highlighted a significant limitation stemming from dataset imbalance, particularly in underrepresented event types such as "Cyclonic" categories. This imbalance can introduce biases during training, as transformer-based LLMs tended to be influenced by most of the categories, resulting in reduced performance for underrepresented categories. We employed MLSOL to mitigate these effects, and our findings with LLaMA models suggested promising results, indicating that the performance issues were primarily data-driven more than model-related. Future research could shed more light into these biases by expanding the dataset to include exploring advanced data augmentation techniques or incorporating active learning strategies that prioritize underrepresented categories.

Finally, LLMs showed remarkable multilingual capabilities to understand and generate text data in various categories. This versatility made these models highly effective for analyzing the frequency and severity of different disaster categories. For instance, during emergency responses, LLMs can assist in real-time translating critical information, ensuring that all affected populations receive timely and accurate updates, regardless of language barriers. Moreover, the multilingual and multi-task capabilities of large transformer models, such as the BLOOM model, have expanded substantially, enabling them to process and analyze vast amounts of multilingual data across diverse tasks (Le Scao et al., 2023). Real-time situational insights provided by LLMs can help decision-makers formulate timely, accurate responses, optimize resource allocation, and coordinate rescue operations effectively. The superior efficiency and stability of LLaMA-3 over LLaMA-2 and BART models position it as a highly effective tool for real-time disaster response. For example, these models can be designed as an enhanced mobile app to provide real-time instructions and information to the public during emergencies. The app can categorize various emergency scenarios, communicate essential data to response teams, and offer users

actionable instructions on emergency preparedness, evacuation routes, shelter plans, and disaster response.

This study underscores the capability of LLMs to effectively uncover complex patterns and structures in disaster-related texts, facilitating more accurate classification through advanced representation learning. Moreover, fine-tuning and data augmentation consistently enhanced model performance, demonstrating LLMs' adaptability to underrepresented or highly imbalanced disaster categories. The methodology developed herein can be used for enhancing disaster detection and classification in other geographic regions, improving emergency response systems, and facilitating real-time monitoring and analysis of disaster events. Additionally, it can be adapted to various sectors such as public health, and environmental monitoring where timely and accurate text-based information processing is critical for effective decision-making and resource allocation.

As LLMs continue to accumulate extensive grounded knowledge from massive amounts of text data, we anticipate rapid innovations in integrating these models with time series neural network algorithms. This could enable the intuitive development and validation of simulation-based time series-based neural network algorithms for flood forecasting. Such developments could profoundly enhance and transform how we build, test, and deploy flood forecasting models. Overall, the synergistic integration of LLMs with flood report data is a promising frontier that can provide opportunities and challenges and warrant extensive future interdisciplinary research.

CRediT authorship contribution statement

Nima Zafarmomen: Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Vidya Samadi:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Software availability

Name of packages: Facebook/BART-large [1]; Facebook/BART-large- MNLI [2], Google-BERT/BERT-base-uncased [3], Meta-LLaMA/LLaMA-2-7b-hf [4], Meta-LLaMA/LLaMA-2-13b-hf [5], Meta-LLaMA/LLaMA-3-8b-hf [6], and Meta-LLaMA/LLaMA-3.1-8b-hf [7].

Year first available: 2019 [1], 2019 [2], 2018 [3], 2023 [4], 2023 [5], 2024 [6], and 2024 [7].

Developers: Facebook AI [1], Facebook AI [2], Google AI [3], Meta AI [4], Meta AI [5], Meta AI [6], and Meta AI [7].

Package Availability: https://huggingface.co/facebook/bart-large [1], https://huggingface.co/facebook/bart-large-mnli [2], https://huggingface.co/google-bert/bert-base-uncased [3], https://huggingface.co/meta-llama/Llama-2-7b-hf [4], https://huggingface.co/meta-llama/Llama-2-13b-hf [5], https://huggingface.co/meta-llama/Meta-Llama-3-8B-hf [6], and https://huggingface.co/meta-llama/Llama-3.1-8B-hf [7].

License: Apache 2.0 License, and Custom Meta License.

Software requirements: transformers library.

Declaration of competing interest

The contact author has declared that none of the authors has any competing interests.

Acknowledgments

This work is supported by the U.S. National Science Foundation (NSF) Directorate for Engineering under grant CMMI2125283 and CBET2429082. All opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. Clemson University is acknowledged for its generous allotment of computing time on the Palmetto cluster.

Data availability

Data will be made available on request.

References

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., Sanghai, S., 2023. Gqa: training generalized multi-query transformer models from multi-head checkpoints. ArXiv Preprint ArXiv:2305.13245.

Alammary, A.S., 2022. BERT models for Arabic text classification: a systematic review. Appl. Sci. 12 (11), 5720.

Anderson, E., Fritz, J., Lee, A., Li, B., Lindblad, M., Lindeman, H., Meyer, A., Parmar, P., Ranade, T., Shah, M.A., 2024. The design of an LLM-powered unstructured analytics system. ArXiv Preprint ArXiv:2409.00847.

Balde, G., Roy, S., Mondal, M., Ganguly, N., 2024. Adaptive BPE tokenization for enhanced vocabulary adaptation in finetuning pretrained Language Models. ArXiv Preprint ArXiv:2410.03258.

Brown, T.B., 2020. Language models are few-shot learners. ArXiv Preprint ArXiv: 2005.14165.

de Bruijn, J.A., de Moel, H., Weerts, A.H., de Ruiter, M.C., Basar, E., Eilander, D., Aerts, J. C.J.H., 2020. Improving the classification of flood tweets with contextual hydrological information in a multimodal neural network. Comput. Geosci. 140, 104485.

Devlin, J., Ming-Wei, Chang, Lee, K., 2018. Bert: pre-training of deep bidirectional transformers for language understanding. ArXiv Preprint ArXiv:1810.04805.

Donratanapat, N., Samadi, S., Vidal, J.M., Tabas, S.S., 2020. A national scale big data analytics pipeline to assess the potential impacts of flooding on critical infrastructures and communities. Environ. Model. Software 133, 104828.

Du, H., Jia, Q., Gehringer, E., Wang, X., 2024. Harnessing large language models to auto-evaluate the student project reports. Comput. Educ.: Artif. Intell. 7, 100268.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., 2024. The llama 3 herd of models. ArXiv Preprint ArXiv:2407.21783.

Efron, B., Tibshirani, R.J., 1993. An Introduction to the Bootstrap. Chapman & Hall, New York, p. 436.

Fan, C., Mostafavi, A., Gupta, A., Zhang, C., 2018. A system analytics framework for detecting infrastructure-related topics in disasters using social sensing. Advanced Computing Strategies for Engineering: 25th EG-ICE International Workshop 2018, Lausanne, Switzerland, June 10-13, 2018, Proceedings, Part II 25, 74–91.

FEMA, 2023. Federal emergency management agency. National Risk Index. U.S. Department of Homeland Security. *n.d.*. https://hazards.fema.gov/nri/map.

Foroumandi, E., Moradkhani, H., Sanchez-Vila, X., Singha, K., Castelletti, A., Destouni, G., 2023. ChatGPT in hydrology and earth sciences: opportunities, prospects, and concerns. Water Resour. Res. 59 (10), e2023WR036288. Wiley Online Library.

Ghosh, S., Maji, S., Desarkar, M.S., 2022. GNoM: graph neural network enhanced language models for disaster related multilingual text classification. Proceedings of the 14th ACM Web Science Conference 2022, pp. 55–65.

Goecks, V.G., Waytowich, N.R., 2023. Disasterresponsegpt: large language models for accelerated plan of action development in disaster response scenarios. ArXiv Preprint ArXiv:2306.17271.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., 2021. Lora: low-rank adaptation of large language models. ArXiv Preprint ArXiv: 2106.09685.

Jayawardene, V., Huggins, T.J., Prasanna, R., Fakhruddin, B., 2021. The role of data and information quality during disaster response decision-making. Progress in Disaster Science 12, 100202.

Jiang, Y., Pan, Z., Zhang, X., Garg, S., Schneider, A., Nevmyvaka, Y., Song, D., 2024. Empowering time series analysis with large language models: a survey. ArXiv Preprint ArXiv:2402.03182.

Kadhim, A.I., 2019. Survey on supervised machine learning techniques for automatic text classification. Artif. Intell. Rev. 52 (1), 273–292.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D., 2020. Scaling laws for neural language models. ArXiv Preprint ArXiv:2001.08361.

Karanjit, R., Samadi, V., Hughes, A., Murray-Tuite, P., Stephens, K., 2024. Converging human intelligence with AI systems to advance flood evacuation decision making. Natural Hazards and Earth System Sciences Discussions 2024, 1–29.

Karimiziarani, M., Moradkhani, H., 2023. Social response and disaster management: insights from twitter data assimilation on hurricane ian. Int. J. Disaster Risk Reduct. 95, 103865.

Le Scao, T., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., 2023. B. Loom: A 176b-Parameter Open-Access Multilingual Language Model.

Lee, J., Toutanova, K., 2018. Pre-training of deep bidirectional transformers for language understanding. ArXiv Preprint ArXiv:1810.04805 3 (8).

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., 2019. Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ArXiv Preprint ArXiv: 1910.13461.

Liu, B., Tsoumakas, G., 2020. Synthetic oversampling of multi-label data based on local label distribution. Machine Learning and Knowledge Discovery in Databases:

European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part II, 180–193.
Liu, X., McDuff, D., Kovacs, G., Galatzer-Levy, I., Sunshine, J., Zhan, J., Poh, M.-Z., Liao, S., Di Achille, P., Patel, S., 2023. Large language models are few-shot health learners. ArXiv Preprint ArXiv:2305.15525.
Otal, H.T., Stern, E., Canbaz, M.A., 2024. LLM-assisted crisis management: building advanced LLM platforms for effective emergency response and public collaboration. 2024 IEEE Conference on Artificial Intelligence (CAI), pp. 851–859.
Parthasarathy, V.B., Zafar, A., Khan, A., Shahid, A., 2024. The ultimate guide to fine-tuning llms from basics to breakthroughs: an exhaustive review of technologies, research, best practices, applied research challenges and opportunities. ArXiv Preprint ArXiv:2408.13296.
Patel, L., Jha, S., Guestrin, C., Zaharia, M., 2024. LOTUS: enabling semantic queries with LLMs over tables of unstructured and structured data. ArXiv Preprint ArXiv: 2407.11418.
Rahman, A.B.S., Ta, H.-T., Najjar, L., Azadmanesh, A., Gönul, A.S., 2024. DepressionEmo: a novel dataset for multilabel classification of depression emotions. J. Affect. Disord. 366, 445–458.
Saberian, M., Samadi, V., Popescu, I., 2024. Probabilistic hierarchical interpolation and interpretable configuration for flood prediction. Hydrol. Earth Syst. Sci. Discuss. 2024, 1–41.
Sajja, R., Xiong, S., Mermer, O., Sermet, Y., Demir, I., 2025. A comprehensive bibliometric analysis of large Language Models in hydrology and environmental sciences. Down Earth.
Slaets, J.I.F., Piepho, H.-P., Schmitter, P., Hilger, T., Cadisch, G., 2017. Quantifying uncertainty on sediment loads using bootstrap confidence intervals. Hydrol. Earth Syst. Sci. 21 (1), 571–588.
Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 45 (4), 427–437.
Terlinden-Ruhl, L., Couasnon, A., Eilander, D., Hendrickx, G.G., Mares-Nasarre, P., Antolínez, J.A.Á., 2024. Accelerating compound flood risk assessments through active learning: a case study of Charleston County (USA). Natural Hazards and Earth System Sciences Discussions 2024, 1–34.
Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., 2023. Llama 2: open foundation and fine-tuned chat models. ArXiv Preprint ArXiv:2307.09288.
Wang, C., Li, M., Smola, A.J., 2019. Language models with transformers. ArXiv Preprint ArXiv:1904.09408.
Wilkho, R.S., Chang, S., Gharaibeh, N.G., 2024. FF-BERT: a BERT-based ensemble for automated classification of web-based text on flash flood events. Adv. Eng. Inform. 59, 102293.
Wu, Y., 2016. Google's neural machine translation system: bridging the gap between human and machine translation. ArXiv Preprint ArXiv:1609.08144.
Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z., Lu, Y., 2023. Temporal data meets LLM–explainable financial time series forecasting. ArXiv Preprint ArXiv:2306.11025.
Zafarmomen, N., Alizadeh, H., Bayat, M., Ehtiat, M., Moradkhani, H., 2024. Assimilation of sentinel-based leaf area index for modeling surface-ground water interactions in irrigation districts. Water Resour. Res. 60 (10), e2023WR036080.
Zhou, B., Zou, L., Mostafavi, A., Lin, B., Yang, M., Gharaibeh, N., Cai, H., Abedin, J., Mandal, D., 2022. VictimFinder: harvesting rescue requests in disaster response from social media with BERT. Comput. Environ. Urban Syst. 95, 101824.
Zhu, X., Fu, Y., Zhou, B., Lin, Z., 2024. Critical data size of language models from a grokking perspective. ArXiv Preprint ArXiv:2401.10463.