

## Article

# Detecting and Classifying Typhoon Information from Chinese News Based on a Neural Network Model

Danjie Chen <sup>1,2,3,4</sup> , Fen Qin <sup>1,3</sup>, Kun Cai <sup>1,2,3,4</sup> and Yatian Shen <sup>2,4,\*</sup>

- <sup>1</sup> College of Environment and Planning, Henan University, Kaifeng 475004, China; chendanjie@henu.edu.cn (D.C.); qinfen@henu.edu.cn (F.Q.); ckheny@vip.henu.edu.cn (K.C.)  
<sup>2</sup> College of Computer and Information Engineering, Henan University, Kaifeng 475004, China  
<sup>3</sup> Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Ministry of Education, Henan University, Kaifeng 475004, China  
<sup>4</sup> Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China  
\* Correspondence: ytshen@henu.edu.cn

**Abstract:** Typhoons are major natural disasters in China. Much typhoon information is contained in a large number of network media resources, such as news reports and volunteered geographic information (VGI) data, and these are the implicit data sources for typhoon research. However, two problems arise when using typhoon information from Chinese news reports. Since the Chinese language lacks natural delimiters, word segmentation error results in trigger mismatches. Additionally, the polysemy of Chinese affects the classification of triggers. Second, there is no authoritative classification system for typhoon events. This paper defines a classification system for typhoon events, and then uses the system in a neural network model, lattice-structured bidirectional long-short-term memory with a conditional random field (BiLSTM-CRF), to detect these events in Chinese online news. A typhoon dataset is created using texts from the China Weather Typhoon Network. Three other datasets are generated from general Chinese web pages. Experiments on these four datasets show that the model can tackle the problems mentioned above and accurately detect typhoon events in Chinese news reports.

**Keywords:** typhoon; classification; event detection; polysemy; lattice; BiLSTM-CRF; Chinese news reports



**Citation:** Chen, D.; Qin, F.; Cai, K.; Shen, Y. Detecting and Classifying Typhoon Information from Chinese News Based on a Neural Network Model. *Sustainability* **2021**, *13*, 7332. <https://doi.org/10.3390/su13137332>

Academic Editor: Andreas Kanavos

Received: 21 April 2021  
Accepted: 26 June 2021  
Published: 30 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Typhoons are major natural disasters, causing serious harm to human life and property. China is close to the typhoon-prone area in the Pacific Ocean. In summer and autumn, southern China in particular is frequently attacked by typhoons. Although there are early warning mechanisms and defensive measures, typhoons still incur significant personal and economic losses. Thus, research on typhoons can provide valuable information directly related to the national economy and people's livelihoods. With more precise advanced warning, people can prepare and protect their property more effectively and efficiently. However, the sources of data necessary for typhoon research are relatively fixed. The data needed for typhoon research mainly come from image data, meteorological data, and statistical data. The statistical data regarding typhoons are usually managed by the Chinese government. As all the data comes from professional departments, data acquisition is difficult, and real-time data cannot be obtained, especially statistical data. These problems hinder typhoon research. There is an urgent need for obtaining data easily and quickly.

With the rapid development of the Internet, there is now much typhoon information publicly available online. This information includes warnings as a typhoon develops, real-time information about the weather (e.g., wind speed, rain), and information of the effects of a typhoon as it passes (e.g., flooding, damage to infrastructure such as roads and buildings). These data are diverse and easier to obtain than those of professional departments.

Wikipedia, online news, and social media have become implicit data sources of typhoon information [1]. However, the data are massive, scattered, and in various formats. Finding relevant data by manual search is almost impossible [2]. Event detection technology is the most practical solution for retrieving typhoon information from online sources.

Event detection technology can cull information of interest from massive data. Events can be defined as real-world occurrences that unfold in space and over time [3]. Event detection from conventional media sources has long been addressed in the topic detection and tracking (TDT) research program, which mainly aims at finding and following events in a stream of broadcast news stories [3]. Therefore, event detection technology can be applied to find typhoon information in massive amounts of data automatically. Event detection is the first and key step in event extraction. However, most of the research is focused on typhoon information extraction [4–7], and there is little research on typhoon event detection.

Furthermore, there is no authoritative classification system for typhoon events. Yu [6] defined a classification hierarchy for disasters recorded in microblogs. The hierarchy includes six categories: buildings, green plants, transportation, water and electricity, other, and useless. The classification scheme does not refer to typhoons in particular and is not complete. A classification system would organize information efficiently, making it more useful to people and governments, particularly in preparing for typhoons. In order to analyze all aspects of typhoon and prepare for future typhoon data extraction, typhoon information is regarded as a collection of small-scale information. According to the description of the information, the small-scale information is defined as different types. Each type corresponds to a type of event. Thus, typhoon is composed of several types of events. By analyzing a large number of typhoon reports and by considering other general disaster classification systems [5,6], a typhoon classification system is proposed.

Unlike those for common event detection, there are no ready-made experimental datasets for typhoon event detection. Much research on typhoons organize the data from volunteered geographic information (VGI) or blogs separately [4–7]. This paper creates an experimental dataset specifically for typhoon event detection. The data come from the reports published as a special column on the China Weather Typhoon Network (the website of this special typhoon column in 2020 is <http://typhoon.weather.com.cn/hist/2020.shtml> (accessed on 12 April 2021)). The reports on typhoon events cover 13 years, from 2008 to 2020. To the best of our knowledge, this is the first Chinese news dataset for typhoon detection experiments.

Event detection includes two pivotal stages. The first stage is to locate the trigger words, called trigger identification. The second stage is to classify the trigger words into corresponding event types, called trigger classification. Although neural network methods have made great achievements in event detection [8,9], there are still two issues. In the trigger identification stage, the mismatching of trigger words can severely affect the event detection performance of the resulting model. In Chinese, words are the basic semantic units, and the mainstream approaches of event detection in Chinese are mostly word-based models [10]. There are no natural delimiters between words for segmentation, so it is usually necessary to segment words first. However, word segmentation may cause problems in that a trigger word may be a part of one word or contain several words. For example, the trigger word “死伤” (die and injure) has two parts, “死” (die) and “伤” (injure), both of which are trigger words of the event types “Life: Die” and “Life: Injure” in automatic content extraction (ACE). The trigger word “恢复上课” (resume classes) contains two words, “恢复” (resume) and “上课” (begin classes). In such cases, word-based methods cannot identify trigger words effectively. Some methods have been proposed to fuse word information and character information to realize trigger identification [9,11–14]. Zeng [9] combined bidirectional long–short-term memory (Bi-LSTM) with a convolutional neural network (CNN) to capture lexical information and character information without any artificial features. However, this method still has the problem of inducing trigger segmentation errors. Lin et al. [11] proposed a word-based event block model—nugget

proposal networks—to solve the problem of the mismatches between words and trigger words. However, the method limits the scope of the trigger candidates to a fixed-sized window, which may cause overlap among the triggers.

After correctly locating the trigger words, classifying them is affected by the problem of polysemy. If a trigger word has multiple word senses, it is important to decide which applies in the context and which event type should be chosen. For example, the trigger “关闭” (close) could represent different event types. In some cases, the trigger “close” may be classified into the “Conflict: Demonstrate” event (close the subway station) in the event classification process of ACE. In other cases, the trigger “close” may be classified into the “Business: End-Org” event (close the business).

In order to prove the universality of the two problems above, Ding et al. [10] provide some statistics of these problems in common datasets—the ACE 2005 and the Knowledge Base Population (KBP) 2017 datasets. The proportion of polysemous words in the KBP 2017 dataset is greater than 50%, and the proportion of trigger word mismatches in the KBP 2017 dataset is greater than 20%. These high proportions show that these problems may affect event detection.

To solve the problem of word segmentation errors, words are generated not by any segmentation system, but by a knowledge base, HowNet [15]. HowNet is a Chinese semantic knowledge base, in which there are more than 200,000 entries. Each entry consists of a word and its word sense. Some words appear only once, but some words appear in many entries in HowNet. That is, some words have many word senses, called polysemous word. Words are automatically obtained by matching the sentences with the entries.

If only word information or character information is exploited in event detection, the other information will be ignored. This paper makes use of both word information and character information. Event detection is treated as a sequence annotation task. Bidirectional long–short-term memory with a conditional random field (BiLSTM-CRF) is a mainstream sequence annotation model. This paper uses this model to process character information. Additional processing units are needed to learn the features of words. A lattice-structured LSTM network is used to learn word senses. The words that end with the same character are the inputs of the lattice LSTM cell for this character. Moreover, if a word has multiple senses, all the senses are input into the same lattice LSTM cell. Lattice LSTM cells choose the most relevant characters and words from a given sentence. The lattice-structured BiLSTM-CRF model can leverage both word information and character information. Hence, with an external knowledge base and the utilization of character information and word information, both problems discussed above can be alleviated. Ding et al. [10] have compared some models that use the information of words and characters. The performance of the lattice-structured LSTM proved better. Experiments are carried out on the experimental typhoon dataset and three general news datasets. The results show that this method successfully detects typhoon events.

## 2. Related Work

The resources for event detection can be online news or social media data. Many studies have attempted to detect and cluster events from news reports [16–20]. For example, Liu et al. [20] clustered news reports according to daily major events such as economic and societal news, and Yu and Wu [19] aggregated news reports related to the same event into a topic-centered collection. Other than online news, many social media, such as Twitter and microblogs, are utilized in event detection [21–27]. Cordeiro [28] designed a time-decaying factor to detect events with Twitter. Petroni [23] described a large-scale automated system for extracting natural disasters and major events from news reports and social media. Ritter [25] described TwiCal, the first open-domain event extraction and categorization system for Twitter. Zhou [27] proposed a simple yet effective Bayesian model to extract information from Twitter.

At present, event detection methods are classified into two classes: feature-based [29–33] and neural network-based [8,10,34–38]. Feature-based methods utilize the features of

language for event detection. Specific features include lexical features, syntactic features, entity information, and textual features. Lan et al. noticed the effect of named entities in event detection [29]. Similar ideas are found in the work of Zhang et al. [30] and Yang et al. [31]. Kumaran applied a classification approach and the named entities to an event detection task [32]. Nguyen and Grishman took syntactic information into account for event detection [33]. Fan Hong [39] combined the improved term frequency-inverse document frequency (TF-IDF) algorithm and syntactic analysis to detect earthquake events in web news. Yang [40] proposed a fast disaster loss identification and classification method to extract the disaster information from social media data by extending the obtained context features and matching feature words. Huang et al. [4] combined events and context features to extract typhoon events. Nevertheless, such features need to be designed manually, which is time-consuming and laborious and has poor scalability.

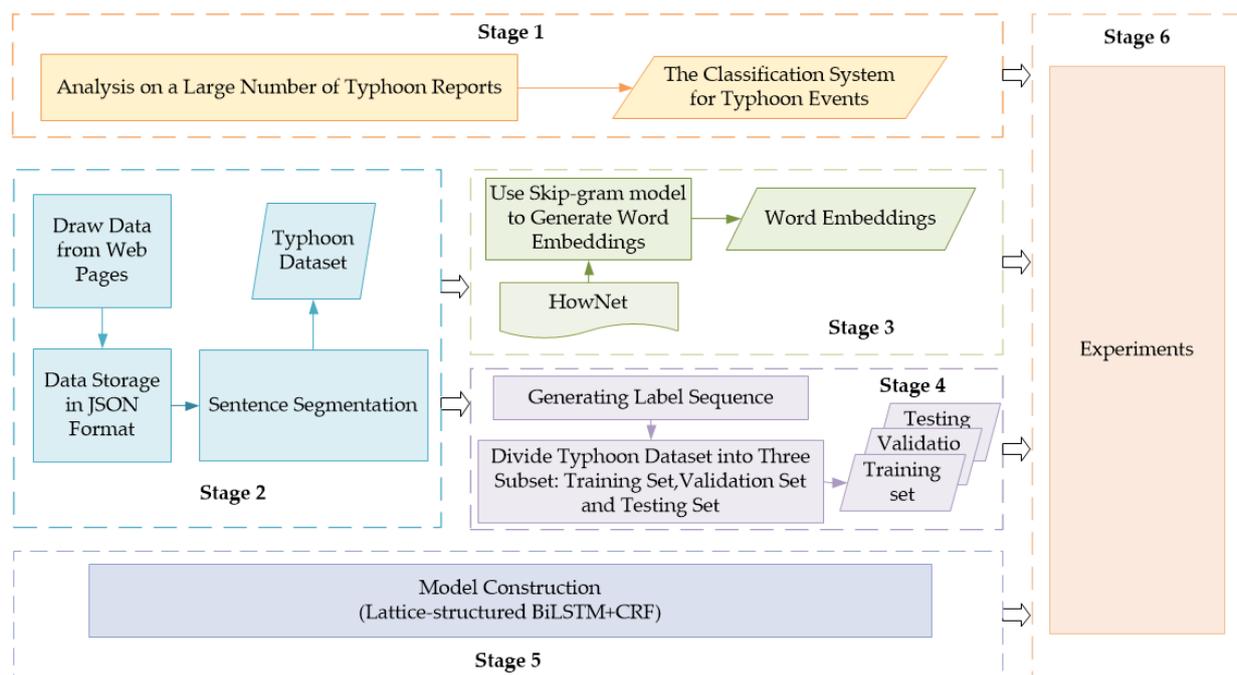
Since neural networks can learn the features of input automatically, many neural network-based methods have been applied for event detection [8,10,34–38]. Nguyen [8] studied the event detection problem using CNNs that overcome the two fundamental limitations of traditional feature-based approaches: the requirement of complicated feature engineering for rich feature sets, and the propagation of errors from the preceding stages that generate these features. He [35] proposed improving the current CNN models for event detection by introducing nonconsecutive convolution. Liu et al. [36] detected events with supervised attention mechanisms. Veyseh [37] proposed employing a self-attention mechanism for neural text modeling to achieve semantic structure induction. Lai [38] formulated event detection as a few-shot learning problem to extend event detection to new event types. Yu et al. [6] explored CNN to extract typhoon information from VGI. These methods are common and effective for English datasets, but do not solve the problems of word segmentation and word sense disambiguation in Chinese. Ding [10] proposed a trigger-aware lattice-structured neural network to detect events in Chinese. This method can solve the above problems and is suitable for Chinese datasets.

Lattice-structured recurrent neural networks (RNNs) can be viewed as natural extensions of tree-structured RNNs to directed acyclic graphs (DAGs) [41]. Lattice-based models are used to combine character information with word information [42,43]. This paper uses a lattice-based model and HowNet to prevent segmentation errors and solve the problem of polysemy in Chinese by fusing character and word information.

In view of the above review, this paper first defines a comprehensive classification system for typhoon events. Then, the paper presents a neural network-based method that solves the problems of word segmentation and the polysemy in detecting typhoon events in online Chinese news reports.

### 3. Methods

Our neural network-based method is depicted in Figure 1. In stage 1, a large number of typhoon reports were read, and the nature of reports was analyzed. Based on the analysis, the classification system for typhoons and triggers was defined. In stage 2, the typhoon data were drawn from the China Weather Typhoon Network and processed by sentence segmentation. A typhoon dataset was generated. In stage 3, the entries in HowNet were matched in sentences to generate words, which can prevent word segmentation errors. The skip-gram model [44] was used to generate the word embeddings. If a word has many senses in HowNet, the word embedding for each sense should be generated. In stage 4, the typhoon dataset was annotated with sequential labels and divided into three subsets. In stage 5, the model lattice-structured BiLSTM-CRF model was constructed. In stage 6, experiments on typhoon detection were carried out, including the training and the evaluation of the model. The model was trained on different training sets several times. Then, a set of evaluation metrics were used to evaluate whether the model can detect typhoon events accurately. By averaging these metrics obtained in different experiments, the evaluation results of the model were obtained.



**Figure 1.** Flow chart of typhoon event detections.

### 3.1. Classification System for Typhoon Events

After the reading and analysis of a large amount of typhoon reports from webpages, the typhoon information was summarized. The information falls naturally into four aspects: warning before the arrival of typhoon, location changes of the typhoon, weather as the typhoon moves, and effects, especially on infrastructure and casualty. For further research, the information summarized from webpages as transformed into events in small granularity. For example, some events are related to warning and some events are related to weather. For a class of events in small granularity, there is more granular information. Take the weather information for example. The weather events include the information of rain, wind, and the weather influence on waves and tides. A typhoon event is regarded as a collection of events in small granularity.

Based on the analysis above and by considering other general disaster classification systems [5,6], typhoon events were classified into 4 categories and 15 subcategories. The four categories are named state event, weather event, warning event, and effect event. State events refer to the changes of typhoon locations from generation to termination. The category is divided into 4 subcategories, namely generation events, development events, landing events, and termination events. Weather events include 4 subcategories. They are wind events, rain events, wave events, and tide events. Warning events refer to forecasts about wind, rain, and disaster before the arrival of a typhoon.

Effect events refer to the negative effects of typhoons, especially on casualty and infrastructure, including 7 subcategories: transportation events, education events, flood events, infrastructure events, building and crop events, commerce events, and statistics events. Among them, transportation events include events involving flights, ports, high-speed railways, and urban transportation. Educational events include events about the suspensions and resumptions of schools. Flood events refer to floods and urban waterlogging. Infrastructure events are events related to water supply, electric power, and communication. Building and crop events are those affecting houses, apartments, public facilities, trees, and crops. Commerce events refer to panic buying and closing of supermarkets, wet markets, retail businesses, and restaurants. Statistics events refer to the statistical data of the losses incurred by typhoons with respect to people, houses, and crops. This classification system comprehensively covers every aspect of typhoons, as shown in Table 1.

**Table 1.** Classification system for typhoon events with examples of triggers.

Category	Subcategory	Triggers
State Event	Generation	生成 (generate)
	Development	靠近 (near), 移动 (move), 位于 (situate)
	Landing	登陆 (land)
	Termination	停编 (stop)
Weather Event	Wind	风力 (wind power), 大风 (gale), 阵风 (gust)
	Rain	暴雨 (rainstorm), 大雨 (heavy rain), 降水 (precipitation)
	Wave	中浪 (medium wave), 大浪 (large wave), 巨浪 (mountainous wave), 风浪 (storm), 海浪 (sea wave)
	Tide	潮 (tide), 风暴增水 (storm surge)
Warning Event		预警 (early warning), 预报 (forecast)
Effect Event	Transportation	取消 (cancel), 恢复 (resume), 延误 (delay), 停航 (suspend air or shipping service), 封闭 (shut), 关闭 (close), 通行能力 (traffic capacity), 准点率 (punctuality rate), 避风 (shelter from the wind), 限流 (current limiting), 暂停发售 (suspension of sale), 停运 (railway outage), 停开 (stop), 停发 (stop sending), 增开 (run additional or new), 加开 (increase), 交通管制 (traffic control), 受阻 (obstructed), 封桥 (stop using the bridge), 关停 (close down)
	Education	停课 (suspend classes), 恢复上课 (resume classes), 复课 (resume classes)
	Flood	洪水 (flood), 内涝 (waterlogging)
	Infrastructure	中断 (shutdown), 停水 (cut off the water supply), 停电 (cut off the power supply), 恢复供电 (power restoration)
	Building and Crop	损坏 (damage), 吹倒 (blow down), 倒塌 (collapse), 受损 (suffer loss), 刮倒 (blow down), 倒杆 (pole collapse), 掀翻 (overturn), 吹掉 (blow off), 倒伏 (becomes flattened), 连根拔起 (uprooting)
	Commerce	停业 (close down), 抢购 (rush to purchase)
	Statistics	受灾 (hit by a natural adversity), 死亡 (death), 死伤 (injured and killed), 受伤 (injury), 夺走 (snatch away), 被困 (trapped), 转移人口 (transfer population), 撤离 (evacuate), 安置 (place), 损坏房屋 (damage the house), 损失 (loss), 被淹 (flooded), 成灾 (disaster), 绝收 (crop failure)

Triggers are the key elements used to detect and classify events. A trigger can be a verb, a noun, a pronoun, an adjective, etc. [45]. This paper also uses triggers to detect events. Triggers are defined for each category of typhoon events. Due to the richness of the Chinese language, the same meaning can be expressed by different triggers. Thus, for each category, there are many triggers. The triggers are also shown in Table 1.

### 3.2. Data Preparation

First, a crawler was written to collect information from webpages. The name and year of each typhoon and the time, title, and content of the related news were saved into MongoDB. One piece of data in the database corresponds to one news report. A total of 4244 typhoon news reports were obtained, including 16,513 sentences. The language technology platform (LTP) [46] method was used for sentence segmentation.

### 3.3. Data Representation

This paper used two different granularities to represent the input texts. The first granularity is character granularity. Each Chinese character is represented as a word embedding by a skip-gram model. In this way, all the Chinese characters in a text are expressed as one-dimensional vectors of the same length. Figure 2 shows the word embedding representations of Chinese characters, in which a line of squares represents the word embedding of one character.

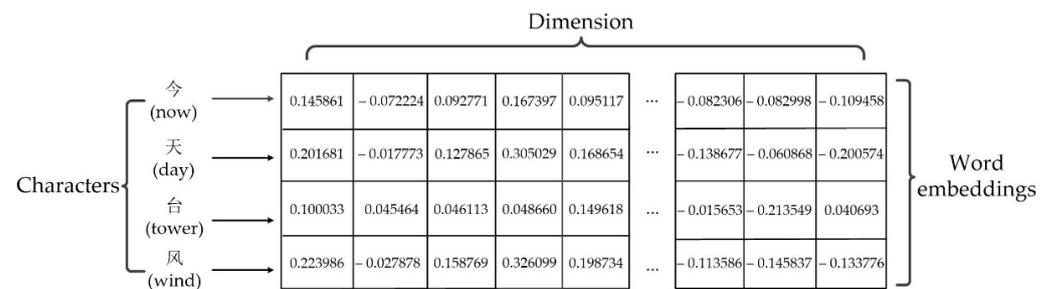


Figure 2. Word embedding representations of Chinese characters.

The second granularity is word granularity. Different from in English, the semantic meanings of sentences in Chinese cannot be expressed by characters alone. A word is an important unit of expression in Chinese. A Chinese word could be a single character or several characters. Additionally, many Chinese words are polysemous, so a word usually conveys different senses. The exact meaning of a word needs to be judged according to the context. Therefore, the polysemy of words must also be considered during event detection. For example, “past” can be used either as a noun to express a time or as a verb to express the meaning of “crossing”. To better express the semantics of a sentence, a word should also be represented as a word embedding. Different senses of a word correspond to different word embeddings. If a word has only one sense, it has only one word embedding. The skip-gram model is combined with HowNet to generate word embeddings for each word [47]. The word embedding representations of polysemous words are shown in Figure 3. The #N symbol in Figure 3, after each Chinese word, represents the nth sense of the word. Non-polysemous words have the same word embedding representation as the polysemous words.

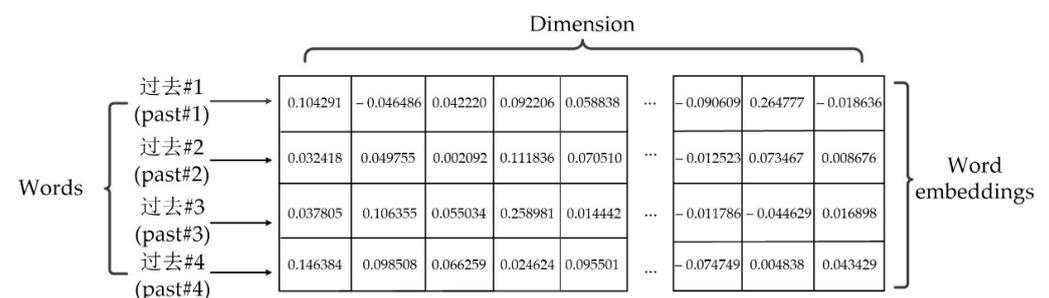


Figure 3. Word embedding representations of Chinese polysemous words.

Finally, two vector documents were generated. One document, named char.vec, saves word embeddings for characters. The other one, named sense.vec, saves word embeddings for words.

The data representation formed in this paper has two different granularities, but it expresses three levels of information in the associated text. The first level is the character information represented by the word embeddings of characters, the second level is the word information represented by the word embeddings of words, and the third level is the polysemy of a word represented by the different word embeddings of that word.

### 3.4. Generating Label Sequences for Data

The result of sentence segmentation is a TXT document, in which each line is a sentence. The characters in a sentence are marked with their positions. Combined with the triggers defined previously, BIO annotations were made for the sentences. BIO annotation is a common method for sequence annotation tasks. B stands for ‘beginning’, which means the first character of a trigger, I stands for ‘inside’, which refers to other characters in a trigger aside from the first character, and O stands for ‘outside’, which refers to the characters of nontriggers. There are suffixes in BIO annotations. For B and I annotations, the event category is used as a suffix, such as B-Flood, I-Flood, etc. After annotation, three columns were generated for each sentence. The characters of the sentence are in the first column, the position of each character is in the second column, and the BIO annotation corresponding to each character is in the third column. A dataset with sequence annotations is called a “BIO” dataset. According to the number of sentences required for model training and testing, the training set, the validation set, and the testing set are generated by randomly selecting sentences from all the news data. To verify the model, the standards of the sequence annotations, whose name contains “golden”, are generated for the testing set and the validation set separately. A golden file records the standard information regarding the triggers in four columns, namely, news ID, the position, the length, and the type of a trigger.

### 3.5. Data Preprocessing

The inputs of the model are the training “BIO” set, the validation “BIO” set, the testing “BIO” set, the validation golden file, the testing golden file, and the word embeddings of characters and words introduced above. First, three dictionaries were sorted out for three datasets, which are a sequence label dictionary, a character dictionary, and a word dictionary. The dictionaries are shared by the three datasets, excluding duplicate data items. In addition, individual arrays were generated for each of the three datasets. An array is used to store the characters, the words, and sequence labels of a dataset. The other array stores the serial numbers of characters, words, and sequence labels in their respective dictionaries. These arrays are called “value array” and “key array”. The word embeddings of characters and words are stored in a two-dimension tensor. In a two-dimension tensor, the number of rows is the number of sentences, and the number of columns is the embedding size. For the validation golden file and the testing golden files, two dictionaries were defined to store the information. The key is news ID, the position, and the length of a trigger, and the value is the event type of a trigger. Input files and the corresponding data structures are shown in Figure 4.

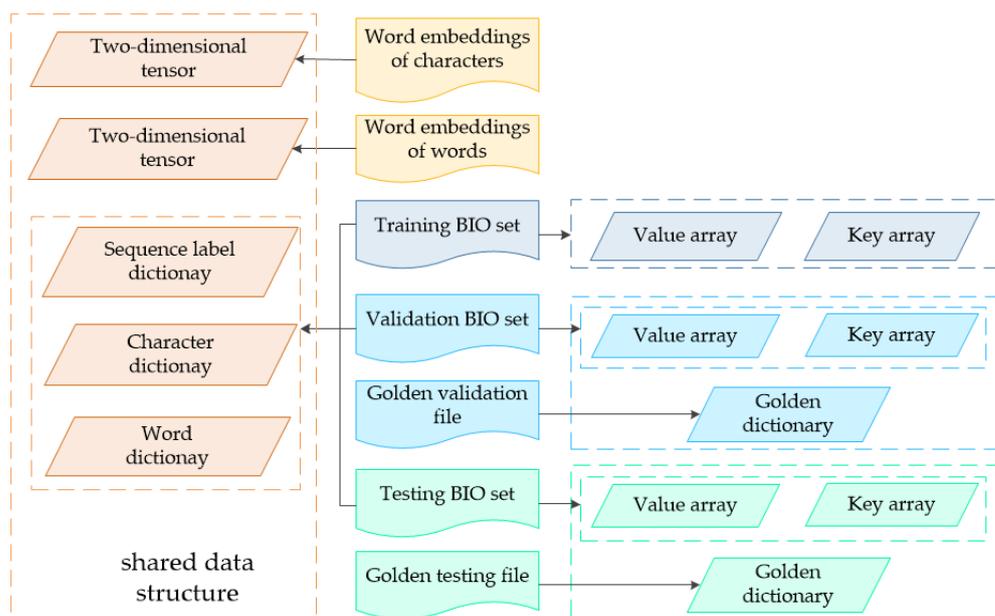
### 3.6. Event Detection Framework

The framework consists of 5 layers. From bottom to top, they are the input layer, the word embedding layer, the BiLSTM layer, the CRF layer, and the tag layer. The core LSTM of the model is lattice-structured LSTM. Lattice-structured LSTM processes not only Chinese character sequences, but also Chinese words that play a positive role in the recognition of triggers. The final CRF layer judges the outputs of the BiLSTM and provides the final serial tags. The event detection framework is shown in Figure 5.

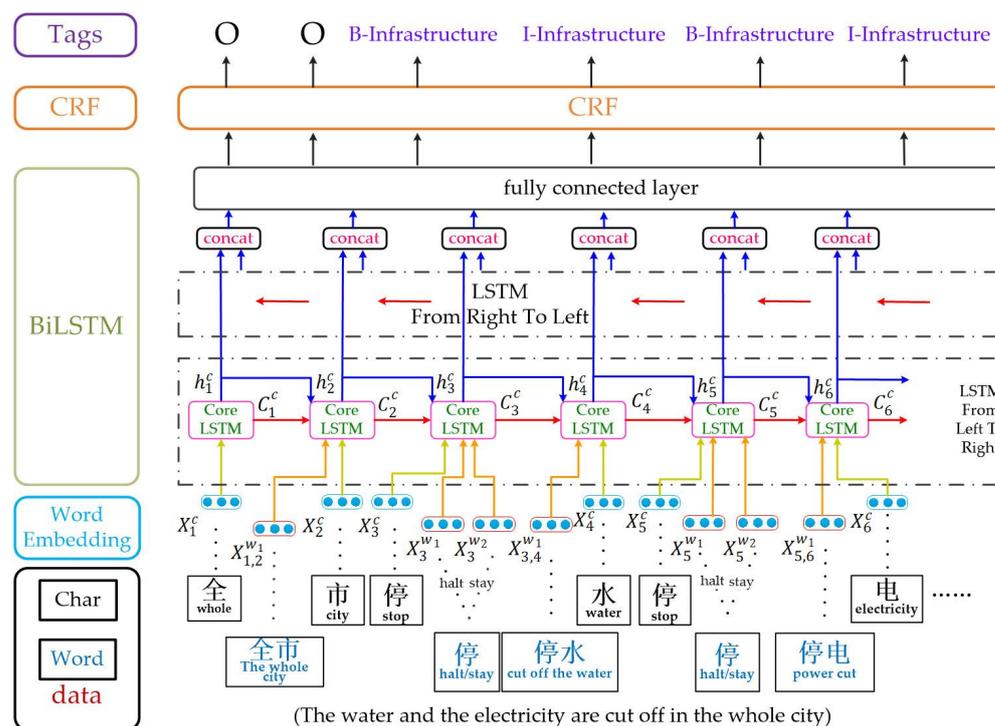
The bottom layer of the framework is the data layer. There are two types of data. One type contains Chinese characters, while the other type consists of Chinese words, which may be polysemous. The layer above the data layer is the word embedding layer, in which Chinese characters and words are converted into word embeddings, which are the inputs of the model.

For characters, the word embedding of each character,  $C_i$ , is:

$$X_i^c = e(C_i) \quad (1)$$



**Figure 4.** Input files and corresponding data structures. The files related to the training set, validation set and testing set are depicted in gray blue, blue and green separately. The files shared by three data sets are depicted in orange.



**Figure 5.** Event detection framework of the lattice-structured BiLSTM-CRF model.

For words, the word embedding of each word,  $w$ , is:

$$X_{b,e}^{w_j} = e(w_{b,e}) \tag{2}$$

The subscripts  $b$  and  $e$  indicate the positions of the beginning character and the ending character respectively, of word  $w$  in a sentence.  $j$  represents the  $j$ -th sense of a polysemous word. For a non-polysemous word, the value of  $j$  is 1.

The layer above the word embedding layer is the BiLSTM layer. The forward direction of the model starts from the beginning of a sentence, and the backward direction starts from the end of a sentence. For the same input, the results from the forward LSTM and the backward LSTM are concatenated as the final result.

The core LSTM consists of three parts. One part is a conventional LSTM cell, which receives the word embeddings of characters, including three gates: an input gate  $i$ , an output gate  $o$ , and a forget gate  $f$ . The LSTM functions are:

$$\begin{bmatrix} i_i^c \\ o_i^c \\ f_i^c \\ \tilde{c}_i^c \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} (W^{cT} \begin{bmatrix} x_i^c \\ h_{i-1}^c \end{bmatrix} + b^c) \quad (3)$$

$$c_i^c = f_i^c \odot c_{i-1}^c + i_i^c \odot \tilde{c}_i^c \quad (4)$$

$$h_i^c = o_i^c \odot \tanh(c_i^c) \quad (5)$$

where  $i_i^c$ ,  $o_i^c$ , and  $f_i^c$  denote the input, output, and forget gates, respectively.  $\tilde{c}_i^c$  denotes an intermediate state of  $C$ ,  $W^{cT}$  and  $b^c$  are model parameters,  $\sigma()$  represents the sigmoid function,  $\tanh()$  represents the activation function,  $x_i^c$  denotes the word embedding of character  $C_i$ ,  $c_i^c$  is the state of the  $i$ -th LSTM cell, and  $h_i^c$  is the output of the  $i$ -th LSTM cell.

The second part of the core LSTM is the lattice-structured LSTM cell, which receives the word embeddings of words. Each sense of a word is calculated by the lattice-structured LSTM cell independently. The cell contains 2 gates: an input gate  $i$  and a forget gate  $f$ . The lattice-structured LSTM cell functions are:

$$\begin{bmatrix} i_j^{w_{m,n}} \\ f_j^{w_{m,n}} \\ c_j^{w_{m,n}} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \tanh \end{bmatrix} (W^{wT} \begin{bmatrix} x_{m,n}^w \\ h_m^c \end{bmatrix} + b^w) \quad (6)$$

$$c_j^{w_{m,n}} = f_j^{w_{m,n}} \odot c_m^c + i_j^{w_{m,n}} \odot \tilde{c}_j^{w_{m,n}} \quad (7)$$

where  $i_j^{w_{m,n}}$  and  $f_j^{w_{m,n}}$  denote the input gate and the forget gate, respectively.  $x_{m,n}^w$  is the word embedding of a word that starts from position  $m$  and ends at position  $n$ ,  $j$  stands for the  $j$ -th sense of word  $w_{m,n}$ ,  $c_j^{w_{m,n}}$  is the cell state of the lattice-structured LSTM cell,  $h_m^c$  is the output of the  $m$ -th conventional LSTM cell, and  $c_m^c$  is the cell state of the  $m$ -th conventional LSTM cell.

The third part of the core LSTM is a gate, which merges the results from the lattice-structured LSTM cell and the conventional LSTM cell. It is a single neural network:

$$g_{m,n}^c = \sigma(W^{lT} \begin{bmatrix} x_n^c \\ c^{w_{m,n}} \end{bmatrix} + b^l) \quad (8)$$

where  $c^{w_{m,n}}$  stands for the merged result of all the senses of  $w_{m,n}$  with  $m \in \{m' | w_{m',n}^d \in \mathbb{D}\}$ .

The final cell status of the core LSTM corresponding to this character  $c_i$  is:

$$c_i^c = \sum_{m \in \{m' | w_{m',j}^d \in \mathbb{D}\}} \alpha_{m,j}^c \odot c^{w_{m,j}} + \alpha_j^c \odot \tilde{c}_j^c \quad (9)$$

The gate values  $g_{m,n}^c$  and  $i_n^c$  are normalized to  $\alpha_{m,n}^c$  and  $\alpha_n^c$  by setting their sum to 1:

$$\alpha_{m,n}^c = \frac{\exp(g_{m,n}^c)}{\exp(i_n^c) + \sum_{m' \in \{m'' | w_{m'',n}^d \in \mathbb{D}\}} \exp(g_{m',n}^c)} \quad (10)$$

$$a_n^c = \frac{\exp(i_n^c)}{\exp(i_n^c) + \sum_{m' \in \{m'' | w_{m'',n}^d \in \mathbb{D}\}} \exp(g_{m',n}^c)} \quad (11)$$

Since the lattice-structured LSTM cell has no output, the output of the core LSTM is  $h_i^c$ .

After the forward LSTM and the backward LSTM finish, their outputs are concatenated. The concatenated result is the input of the fully connected layer. The fully connected layer transforms the input into a one-dimensional vector, in which the values are probability values for the associated sequence labels. Then, the one-dimensional vector is transferred into the next layer, the CRF layer.

The CRF layer processes the input with a trained probability transformation matrix. After the calculation, the labels that have the maximal probability values are the outputs of this layer. For an input sequence  $S = \{c_1, c_2, \dots, c_n\}$ , a corresponding label sequence  $B = \{y_1, y_2, \dots, y_n\}$  is the output. The probability distribution is:

$$P(B|S) = \frac{\exp\left(\sum_{i=1}^N (W_{CRF}^{y_i} h_i + b_{CRF}^{(y_{i-1}, y_i)})\right)}{\sum_{B' \in \mathbb{C}} \exp\left(\sum_{i=1}^N (W_{CRF}^{y'_i} h_i + b_{CRF}^{(y'_{i-1}, y'_i)})\right)} \quad (12)$$

where  $\mathbb{C}$  contains all the possible label sequences for sequence  $S$ , and  $B'$  represents an arbitrary label sequence.  $W_{CRF}^{y_i}$  is a model parameter specific to  $y_i$ , and  $b_{CRF}^{(y_{i-1}, y_i)}$  is a bias specific to  $y_{i-1}$  and  $y_i$ .

The Viterbi algorithm [48] was used to obtain the highest scoring label sequence. The loss function of our model is the log likelihood at the sentence level:

$$L = \sum_{i=1}^M \log(P(B_i|S_i)) \quad (13)$$

where  $M$  is the number of sentences, and  $B_i$  is the correct label sequence for sentence  $S_i$ .

### 3.7. Model Construction

The model has only a one-layer neural network and defines a core BiLSTM unit. Four weight parameters and four bias parameters are set in the LSTM cell (Equation (3)). Its input data are the word embeddings of characters. The lattice-structured LSTM cell processes words, for which three weight parameters and three bias parameters are set (Equation (6)). One weight parameter and one bias parameter are set for the gate, which merges the states of the other two parts (Equation (8)). Finally, the negative log likelihood loss function and the Viterbi algorithm in the CRF layer should be programmed (Equations (12) and (13)).

### 3.8. Hyperparameter Settings of the Model

The dropout mechanism [49] was used in the model, and the dropout rate was set to 0.5. Stochastic gradient descent was utilized as the optimizer. The learning rate was set to 0.015, and the learning rate decay was set to 0.05. The embedding sizes of characters, words, and hidden states are 64, 200, and 160, respectively. The number of epochs was set to 20.

### 3.9. Evaluation Metrics for the Model

Accuracy (Acc), standard micro-averaged precision (P), recall (R), and F1 were used as the evaluation metrics. Accuracy was used to evaluate the correlation between the sequence annotations predicted by the model and the standard sequence annotations in the golden files. Precision is the result of the number of labels predicted correctly divided by the total number of labels. Recall is the result of the number of labels predicted correctly divided by the number of standard labels. F1 is calculated from P and R.

#### 4. Results and Discussion

According to the experimental procedure described above, three experiments were carried out, and 50%, 70%, and 100% of the typhoon dataset were randomly chosen for the three experiments. In each experiment, the data were randomly divided into a training set, validation set, and testing set at a ratio of 6:2:2. The experiment with 70% of the dataset was taken as an example to analyze the results. Finally, the data from common webpages were used to test the model and perform some analyses.

##### 4.1. Training and Testing

For training and testing, 11,559 pieces of data were randomly selected, which equates to 70% of the total data. The training set contains 6835 pieces of data. The validation set and the testing set both contain 2311 pieces of data. A total of 1116 words with no repetitions are in the sense.vec file, and these words are either polysemous or non-polysemous. In the training set, 83,736 words appear in the sense.vec file with repetition, and 20,064 polysemous words are included. In the validation set, 27,167 words appear in the sense.vec file, with 6493 polysemous words. In the testing set, 28,082 words appear in the sense.vec file, with 6618 polysemous words.

After 20 rounds of the model training procedure, two sets of evaluations were obtained. One set determines whether the locations of triggers are accurately located, and the other determines whether the types of triggers are correctly classified after the precise locations are obtained. The evaluations are shown in Table 2. For simplicity, the values in this table are displayed for every five rounds.

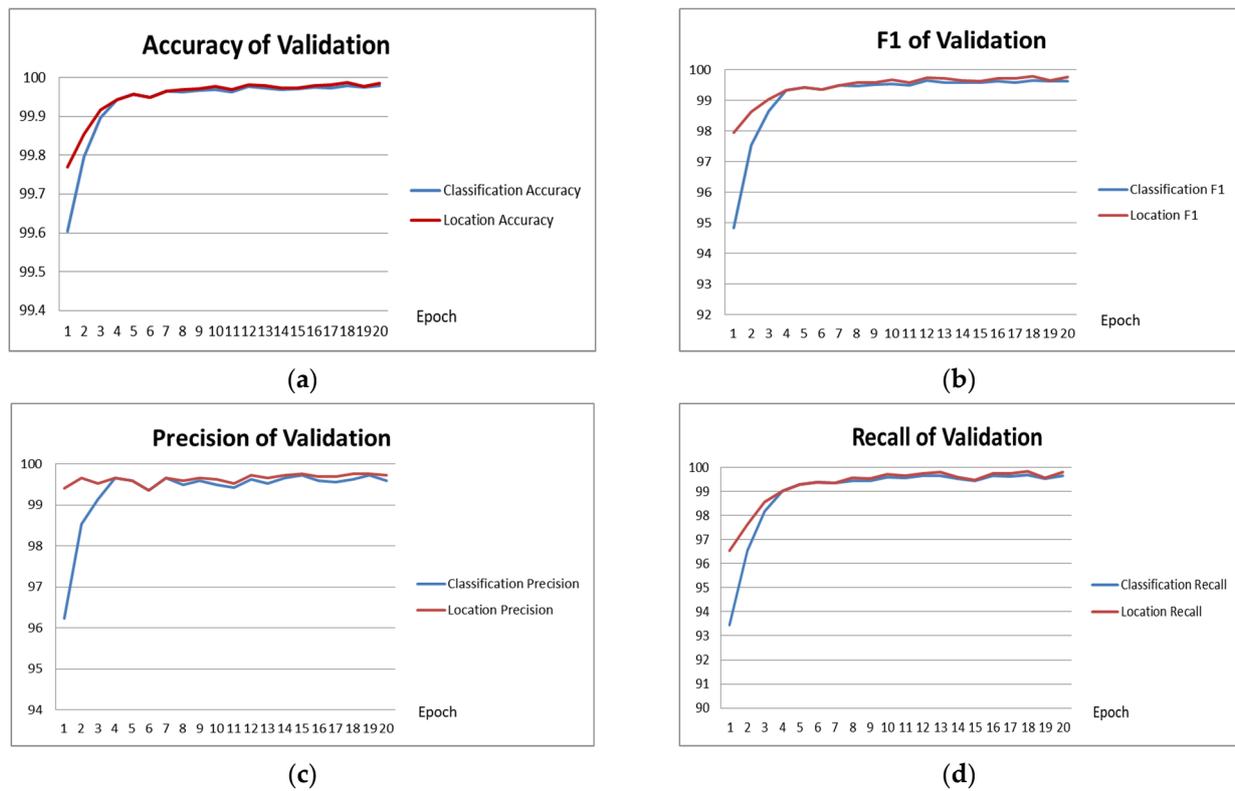
**Table 2.** Values of accuracy (Acc), precision (P), recall (R), and F1 of trigger location and trigger classification every five epochs for the validation set.

Epoch	Location of Triggers				Classification of Triggers			
	Acc	P	R	F1	Acc	P	R	F1
1	99.76	99.40	96.54	97.95	99.60	96.23	93.46	94.82
5	99.95	99.58	99.28	99.43	99.95	99.58	99.28	99.43
10	99.97	99.62	99.72	99.67	99.96	99.48	99.58	99.53
15	99.97	99.75	99.48	99.62	99.97	99.72	99.45	99.58
20	99.98	99.72	99.79	99.76	99.97	99.58	99.65	99.62

After training, all the values of the two evaluation sets were greater than 99%. Figure 6 visualizes the accuracy, precision, recall, and F1 values for every round. Every child window exhibits two curves that separately represent the same evaluation of the trigger location and the trigger classification. From the figure, it can be seen that the shape of the evaluation curve for trigger location is basically the same as that of trigger classification. The two curves are very close. The evaluation values of the trigger location were slightly higher than those of the trigger classification, suggesting that some triggers were located correctly, but their classifications are wrong.

Next, the testing set was used to evaluate the final model. Two sets of evaluations were also obtained. Detailed values are provided in Table 3. Figure 7 shows a comparison between the two sets of evaluation values.

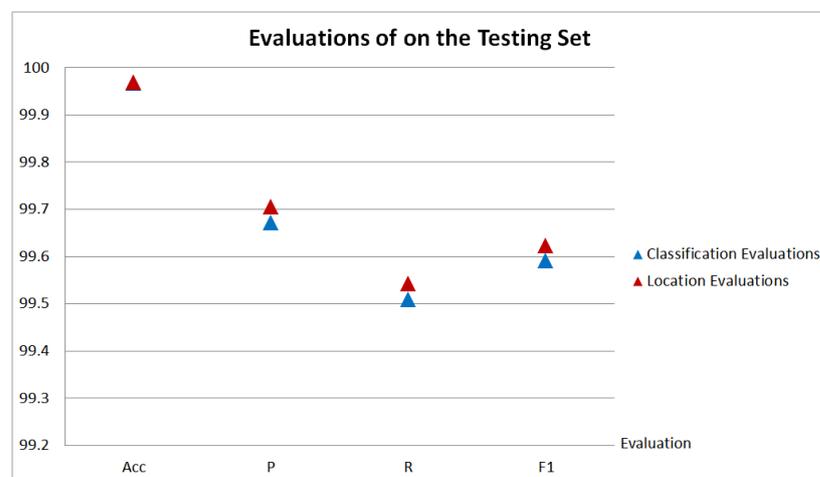
Similar to the evaluations on the training set, the evaluations of the trigger location were slightly higher than those of the trigger classification. The values of Acc, P, R, and F1 of the classification were all greater than 99%. This shows that the model can complete the task of typhoon event detection.



**Figure 6.** (a) Curves of accuracy values of the trigger location and trigger classification for the validation set; (b) Curves of F1 values of the trigger location and trigger classification for the validation set; (c) Curves of precision values of the trigger location and trigger classification for the validation set; (d) Curves of recall values of the trigger location and trigger classification for the validation set.

**Table 3.** Values of accuracy, precision, recall, and F1 of trigger location and trigger classification for the testing set.

Location of Triggers				Classification of Triggers			
Acc	P	R	F1	Acc	P	R	F1
99.96	99.70	99.54	99.62	99.96	99.67	99.50	99.59



**Figure 7.** Comparison of accuracy, precision, recall, and F1 values of trigger location and trigger classification for the testing set.

Three validation experiments were carried out with 50%, 70%, and 100% of the typhoon dataset. By averaging the results of the three validation experiments, the final evaluations of the model were obtained and are shown in Table 4. The final result shows that this model can effectively detect a typhoon.

**Table 4.** Average values of accuracy, precision, recall, and F1 of the model after three validation experiments.

Location of Triggers				Classification of Triggers			
Acc	P	R	F1	Acc	P	R	F1
99.97	99.65	99.69	99.67	99.96	99.59	99.64	99.62

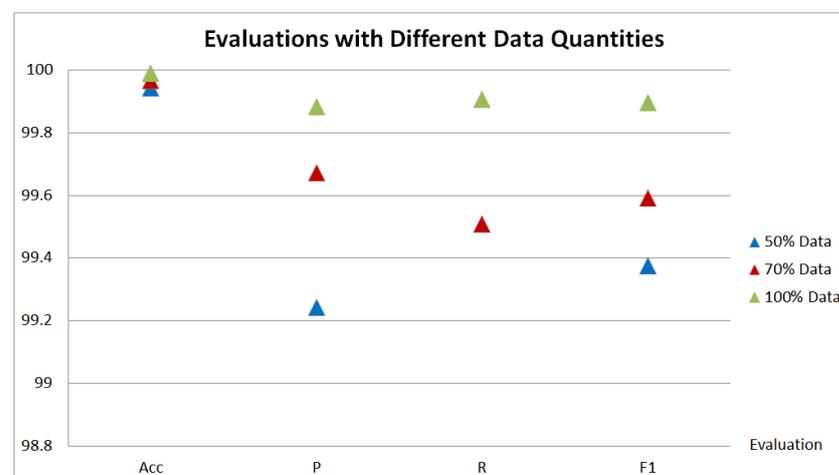
#### 4.2. Influence of Data Quantity and Data Type

First, the impact of data volume on the accuracy of the model was analyzed, where 50%, 70%, and 100% of the dataset were used to train and verify the model. The evaluations of the model in terms of trigger classification were compared under different data quantities. Table 5 shows the different quantities of data, the numbers of clauses in the training set, and the evaluations on the testing set with respect to trigger classification.

**Table 5.** Numbers of clauses in 50%, 70%, and 100% of the dataset and the values of accuracy, precision, recall, and F1 for trigger classification with 50%, 70%, and 100% of the dataset in the testing phase.

Quantity	Number of Clauses	Classification of Triggers			
		Acc	P	R	F1
50%	4924	99.94	99.24	99.50	99.37
70%	6904	99.96	99.67	99.51	99.59
100%	9864	99.99	99.88	99.90	99.89

The number of clauses of the typhoon event in each training set is significantly different. Table 5 and Figure 8 also compare the evaluations with different data quantities. The R indices with 50% of the data and 70% of the data were basically coincident. The evaluation values obtained with 100% of the data were the best. This shows that increasing the amount of data can improve the resulting model.



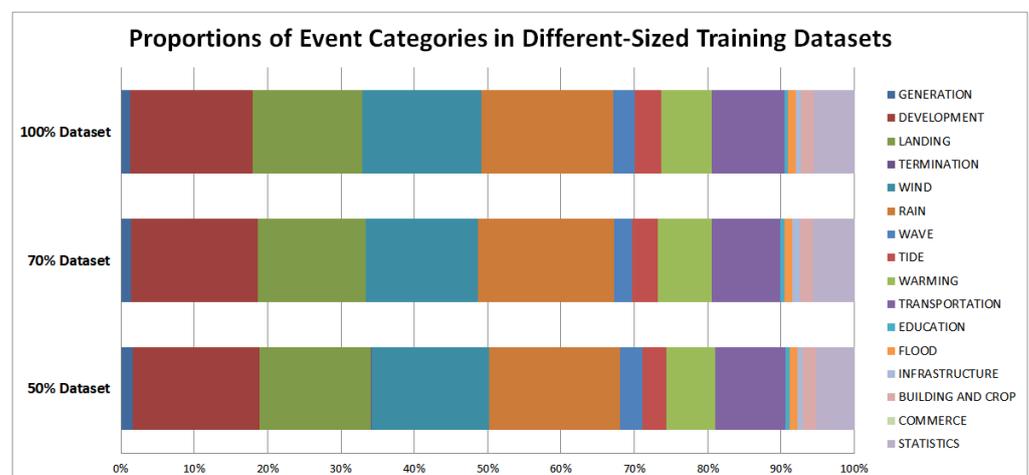
**Figure 8.** Comparison of accuracy, precision, recall, and F1 values of trigger classification with 50%, 70%, and 100% of the dataset in the testing phase.

In the training set, the number of event types also affects the model by determining whether the model can learn all the event features. There are sixteen event categories in this experiment. The numbers of the event categories in different proportions of the training sets are shown in Table 6. Taking 50% of the dataset as an example, it comprises 105 typhoon generation events, 1113 typhoon development events, 976 typhoon landing events, 2 typhoon termination events, 1034 wind events, 1146 rain events, 193 wave events, and 213 tide events. There are 426 cases of warning events, 619 cases of transportation events, 33 cases of education events, 69 cases of flood events, 48 cases of infrastructure events, 115 cases of building and crop events, 5 cases of commerce events, and 333 cases of statistics events.

**Table 6.** Number of categories of 50%, 70%, and 100% of the dataset in the training phrase.

Category	Quantity		
	50%	70%	100%
Generation	105	124	167
Development	1113	1547	2134
Landing	976	1309	1896
Termination	2	1	4
Wind	1034	1372	2076
Rain	1146	1663	2297
Wave	193	217	372
Tide	213	315	458
Warning	426	651	891
Transportation	619	839	1263
Education	33	47	57
Flood	69	100	132
Infrastructure	48	92	89
Building and Crop	115	157	226
Commerce	5	4	7
Statistics	333	503	703

Although the data quantities in the datasets and the number of each event category are different, the proportions of event categories in each training dataset were consistent. All the event categories in classification system were covered. Figure 9 shows the proportions of the various event types in the different training sets. This proves that rich features provided for the training process are helpful for optimizing the model.



**Figure 9.** Proportions of event categories in the 50%, 70%, and 100% datasets.

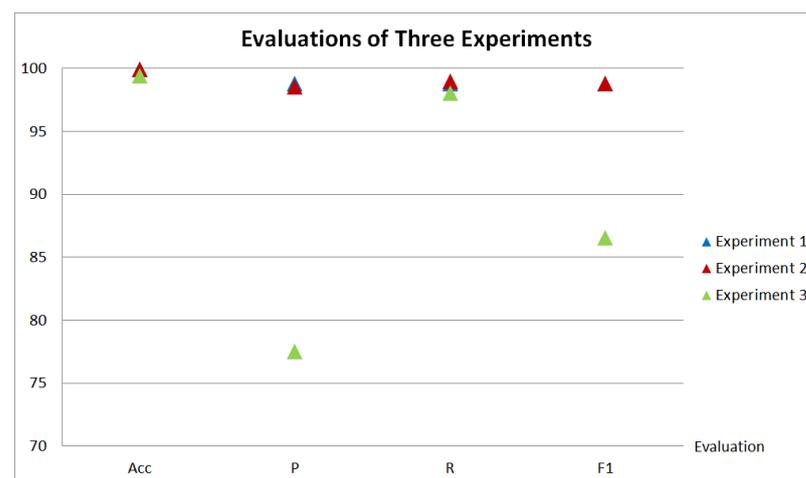
#### 4.3. Detecting Typhoon Events on General Webpages

To further test the model, three verification experiments were carried out. The data in Experiment 1 are pure typhoon information from other webpages, including 111 sentences. The data in Experiment 2 are a mixture of typhoon information and non-meteorological information from other webpages, totaling 477 sentences. The data in Experiment 3 are a mixture of typhoon information, meteorology information, and non-meteorological information from other webpages, for a total of 523 sentences. Three datasets were used to verify the same model. Here, the trigger classification evaluations are compared in the three experiments. The verification results are shown in Table 7.

**Table 7.** Values of accuracy, precision, recall, and F1 of the trigger classification results in the three experiments, where the data are from general webpages.

Experiment	Acc	P	R	F1
Experiment 1	99.91	98.78	98.78	98.78
Experiment 2	99.92	98.53	98.94	98.73
Experiment 3	99.38	77.48	98.01	86.54

Figure 10 shows the visualization results of the evaluations.



**Figure 10.** Comparison of accuracy, precision, recall, and F1 evaluation metrics for trigger classification in the three experiments, where the data are from general webpages.

It can be seen from the figure that the performances of the four evaluation metrics were different for the three datasets. The accuracy values were basically the same. The P of Experiment 1 was the highest, and the P of Experiment 2 was almost the same. However, the P of Experiment 3 was low. This shows that the model is suitable for general news and can accurately predict triggers when typhoon data are mixed with other non-meteorological news items, but meteorological information can compromise its precision. Regarding the R indices, the values of the three experiments were high, indicating that the success rate of correct result prediction was still very high, despite interference from different information types. In summary, the model can detect typhoon information on general webpages. If the information types on a given webpage are similar to typhoons, such as meteorological information and disaster information, the interference results are obvious.

For each experiment, the mispredictions of the model were analyzed. These mispredictions are summarized into three classes. The first class contains prediction errors. This means that the triggers were detected but classified into the wrong type. The second class includes missing predictions, which means that the triggers were not detected. The third class involves situations where new triggers were generated by the model and classified into an existing event type. This class shows that the model can learn similar event triggers.

In Experiment 1, the P and R values were very high. There were only three mispredictions. One belongs to the first class of mispredictions. The trigger “停业” (close down) was classified into the “Education” type, but it should belong to the “Commerce” type. The second misprediction belongs to the third class. The model learned a new trigger “倒损” (reverse and loss). The third belongs to the second class. The trigger “倒杆” (pole collapse) was not found.

The four evaluations metrics of Experiment 2 were all very high. Three mispredictions occurred, and they belong to the first class. The three triggers were “风暴增水” (storm surge), whose type is “Tide”, “风浪” (storm), whose type is “Wave”, and “连根拔起” (uprooting), whose type is “Building and Crop”. Two new triggers were learned by the model: “吹损” (blow and loss), which was classified into the “Building and Crop” type, and “受困” (trapped), which was classified into the “Statistics” type. Three pieces of non-typhoon news were mistakenly classified into state events because the triggers found in the news were the same as the triggers of state events.

In Experiment 3, the value of P decreased significantly. It was indicated that the model predicted more triggers than those in the reference standard. This is because the dataset of Experiment 3 is mixed with common meteorological news, in which the same triggers were detected, but they have nothing to do with typhoons. Due to the interference of meteorological news, there were 86 mispredictions.

From the analyses of these experiments, it is known that more data and more comprehensive event types are beneficial for better training the model. Whether the validations are carried out on the typhoon dataset or the datasets from general webpages, the model can effectively detect typhoon events in news reports.

## 5. Conclusions

In this paper, a neural network method was used to detect typhoons in Chinese news reports. First, a detailed classification system for typhoon events, which has not been defined before, was proposed. Due to the polysemy of Chinese, two data granularities, characters and words, were adopted as the inputs of the model. The skip-gram model was combined with HowNet to generate word embeddings for words and characters in order to make use of rich word senses and solve the problem of word segmentation. This paper also introduced the BiLSTM-CRF model with a lattice structure, which can leverage both word information and character information. Finally, a dataset for experimentations was generated from the China Weather Typhoon Network. After conducting the experiments, the Acc, P, and R values of the model reached 99%. Using typhoon data from other websites, the evaluation metrics also surpassed 98%. When the typhoon news is mixed with meteorology new and disaster news, the performance of the model will degrade. Experiments showed that the method proposed in this paper can accurately detect typhoon information in Chinese news reports, solving the problems of word segmentation and Chinese polysemy.

However, there are two points that can be improved. In the experiments, the total amount of data was not large, and the amount of data for each event type was small, unbalanced, and sparse. The reason for this is that typhoons themselves are relatively sparse in online news. Second, the trigger words may be out-of-vocabulary (oov), so the words cannot be obtained from an external knowledge base. In future, our plans include: (i) to collect more data from news or other resources, such as microblogs and VGI, regarding typhoons, (ii) and to solve the problem of oov.

**Author Contributions:** D.C. is the leading author of this work. She conceived the core ideas, performed the data curation process, and carried out the implementation; F.Q. and K.C. revised the paper; Y.S. offered the experimental platform and revised the paper. They provided substantial contributions to the design and analysis of this work and to the critical review of the article. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (No. 41801281 and No. U1804154).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** We would like to thank the editors and the anonymous reviewers for their very helpful suggestions, all of which have improved the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Han, X.; Wang, J.; Bu, K.; Kun, W. Progress on disaster events information acquisition from web text. *J. Geo Inf. Sci.* **2018**, *20*, 1037–1046.
2. Zhang, X.; Yun, H.; He, Y.; Hu, H. Chinese news event detection and theme extraction based on convolution neural network and K-means. *Sci. Technol. Eng.* **2020**, *20*, 1139–1144.
3. Atefeh, F.; Khreich, W. A Survey of Techniques for Event Detection in Twitter. *Comput. Intell.* **2015**, *31*, 132–164. [[CrossRef](#)]
4. Huang, Z.; Qiu, P.; Wang, H.; Wu, S. Typhoon Event Information Extraction Method Based on Event and Context Characteristics. *J. Geomat. Sci. Technol.* **2019**, *36*, 103–108.
5. Zhao, Q.; Chen, Z.; Liu, C.; Luo, N. Extracting and classifying typhoon disaster information based on volunteered geographic information from Chinese Sina microblog. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4910. [[CrossRef](#)]
6. Yu, J.; Zhao, Q.; Chin, C.S. Extracting Typhoon Disaster Information from VGI Based on Machine Learning. *J. Mar. Sci. Eng.* **2019**, *7*, 318. [[CrossRef](#)]
7. Jia, M.; Zhang, Y.; Pan, T.; Wu, W.; Su, F. Ontology modeling of marine environmental disaster chain for Internet information extraction: A case study on Typhoon Disaster. *J. Geo Inf. Sci.* **2020**, *22*, 2289–2303.
8. Nguyen, T.; Grishman, R. Event Detection and Domain Adaptation with Convolutional Neural Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 2, pp. 365–371. [[CrossRef](#)]
9. Zeng, Y.; Yang, H.; Feng, Y.; Wang, Z.; Zhao, D. *A Convolution BiLSTM Neural Network Model for Chinese Event Extraction*; Springer: Cham, Switzerland, 2016; pp. 275–287.
10. Ding, N.; Li, Z.; Liu, Z.; Zheng, H.; Lin, Z. Event Detection with Trigger-Aware Lattice Neural Network. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019.
11. Lin, H.; Lu, Y.; Han, X.; Le, S. Nugget Proposal Networks for Chinese Event Detection. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 1565–1574.
12. Qin, Y.; Wang, Z.; Zheng, D.; Zhang, M. Hybrid Representation Based Chinese Event Detection. *J. Chin. Inf. Process.* **2019**, *33*, 85–92.
13. Wu, F.; Zhu, P.; Wang, Z.; Li, P.; Zhu, Q. Chinese Event Detection with Joint Representation of Characters and Words. *Comput. Sci.* **2021**, *48*, 249–253.
14. Ding, L.; Xiang, Y. Chinese Event Detection with Hierarchical and Multi-granularity Semantic Fusion. *Comput. Sci.* **2021**, *48*, 202–208.
15. Dong, Z.; Dong, Q. HowNet—A Hybrid Language and Knowledge Resource. In Proceedings of the 2003 International Conference on Natural Language Processing and Knowledge Engineering Proceedings, Beijing, China, 26–29 October 2003.
16. Tanev, H.; Piskorski, J.; Atkinson, M. Real-Time News Event Extraction for Global Crisis Monitoring. In Proceedings of the International Conference on Natural Language & Information Systems: Applications of Natural Language to Information Systems, London, UK, 24–27 June 2008; pp. 207–218.
17. Piskorski, J.; Tanev, H.; Atkinson, M.; Goot, E.V.D.; Zavarella, V. *Online News Event Extraction for Global Crisis Surveillance*; Springer: Berlin/Heidelberg, Germany, 2011.
18. Ribeiro, S.; Ferret, O.; Tannier, X. Unsupervised Event Clustering and Aggregation from Newswire and Web Articles. In Proceedings of the 2017 EMNLP Workshop: Natural Language Processing Meets Journalism, Copenhagen, Denmark, 7 September 2017; pp. 62–67.
19. Yu, S.; Wu, B. Exploiting Structured News Information to Improve Event Detection via Dual-Level Clustering. In Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China, 18–21 June 2018; pp. 873–880.
20. Liu, M.; Liu, Y.; Xiang, L.; Chen, X.; Yang, Q. Extracting Key Entities and Significant Events from Online Daily News. In Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning, Daejeon, Korea, 2–5 November 2008; pp. 201–209.

21. Weng, J.; Lee, B.S. Event Detection in Twitter. In Proceedings of the Fifth International Conference on Weblogs & Social Media, Barcelona, Catalonia, Spain, 17–21 July 2011; pp. 401–408.
22. Zhou, D.; Xuan, Z.; He, Y. Event extraction from Twitter using Non-Parametric Bayesian Mixture Model with Word Embeddings. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; Volume 1, pp. 808–817.
23. Petroni, F.; Raman, N.; Nugent, T.; Nourbakhsh, A.; Panić, Ž.; Shah, S.; Leidner, J.L. An Extensible Event Extraction System With Cross-Media Event Resolution. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 626–635.
24. Guille, A.; Favre, C. Mention-anomaly-based event detection and tracking in Twitter. In Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Beijing, China, 17–20 August 2014; pp. 375–382.
25. Mausam, A.R.; Etzioni, O.; Clark, S. Open domain event extraction from twitter. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 1104–1112.
26. Zhou, D.; Chen, L.; He, Y. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2468–2474.
27. Zhou, D.; Chen, L. A Simple Bayesian Modelling Approach to Event Extraction from Twitter. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014; pp. 700–705.
28. Cordeiro, M. Twitter event detection: Combining wavelet analysis and topic inference summarization. In Proceedings of the Symposium on Doctoral Symposium on Informatics Engineering, Porto, Portugal, 26–27 January 2012; Volume 1, pp. 11–16.
29. Lam, W.; Meng, H.M.L.; Wong, K.L.; Yen, J.C.H. Using contextual analysis for news event detection. *Int. J. Intell. Syst.* **2001**, *16*, 525–546. [[CrossRef](#)]
30. Zhang, K.; Zi, J.; Wu, L.G. New event detection based on indexing-tree and named entity. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, 23–27 July 2007; pp. 215–222.
31. Yang, Y.; Zhang, J.; Carbonell, J.; Jin, C. Topic-conditioned Novelty Detection. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AL, Canada, 23–26 July 2002; pp. 688–693. [[CrossRef](#)]
32. Kumaran, G.; Allan, J. Text Classification and Named Entities for New Event Detection. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research & Development in Information Retrieval, Sheffield, UK, 25–29 July 2004; pp. 297–304.
33. Nguyen, T.H.; Grishman, R. Graph Convolutional Networks with Argument-Aware Pooling for Event Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
34. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
35. Nguyen, T.H.; Grishman, R. Modeling Skip-Grams for Event Detection with Convolutional Neural Networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 886–891.
36. Liu, S.; Chen, Y.; Liu, K.; Zhao, J. Exploiting Argument Information to Improve Event Detection via Supervised Attention Mechanisms. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1789–1798.
37. Veyseh, A.P.B.; Thai, M.T.; Nguyen, T.H.; Dou, D. Rumor detection in social networks via deep contextual modeling. In Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis & Mining, Vancouver, BC, Canada, 27–30 August 2019; pp. 113–120.
38. Lai, V.D.; Dernoncourt, F.; Nguyen, T.H. Extensively Matching for Few-shot Learning Event Detection. *arXiv* **2020**, arXiv:2006.10093.
39. Fan, H.; Li, H.; Du, W.; Yang, J. Web based extraction of spatiotemporal information of earthquake event by semantic technology. *Eng. J. Wuhan Univ.* **2018**, *51*, 183–188.
40. Yang, T.; Xie, J.; Li, Z.; Li, G. A method of typhoon disaster loss identification and classification using micro-blog information. *J. Geo Inf. Sci.* **2018**, *20*, 906–917.
41. Zhang, Y.; Yang, J. Chinese NER Using Lattice LSTM. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 1554–1564.
42. Li, Z.; Ding, N.; Liu, Z.; Zheng, H.; Shen, Y. Chinese Relation Extraction with Multi-Grained Information and External Linguistic Knowledge. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July 2019.
43. Yang, J.; Zhang, Y.; Liang, S. Subword Encoding in Lattice LSTM for Chinese Word Segmentation. 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. 2720–2725.
44. Mikolov, T.; Corrado, G.; Kai, C.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the International Conference on Learning Representations (ICLR 2013), Scottsdale, AR, USA, 2–4 May 2013.
45. Chen, Y.; Ding, Z.; Zheng, Q.; Qin, Y.; Shah, N. A History and Theory of Textual Event Detection and Recognition. *IEEE Access* **2020**, *8*, 201371–201392. [[CrossRef](#)]

46. Che, W.; Li, Z.; Liu, T. LTP: A Chinese Language Technology Platform. In *Coling 2010: Demonstrations*; Coling 2010 Organizing Committee: Beijing, China, 2010; pp. 13–16.
47. Niu, Y.; Xie, R.; Liu, Z.; Sun, M. Improved Word Representation Learning with Sememes. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017*; Volume 1, pp. 2049–2058.
48. Viterbi, A. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Trans. Inf. Theory* **1967**, *13*, 260–269. [[CrossRef](#)]
49. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.