symmetry

MDPI

Article

A Novel Phishing Website Detection Model Based on LightGBM and Domain Name Features

Jingxian Zhou [1,*], Haibin Cui [2], Xina Li [2], Wenjin Yang [1] and Xi Wu [3]

[1] Information Security Evaluation Center, Civil Aviation University of China, Tianjin 300300, China
[2] College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China
[3] NWU-Salento School of Cultural Heritage and Arts, Northwest University, Xi'an 710127, China
* Correspondence: jxzhou@cauc.edu.cn

Abstract: Phishing attacks have evolved in terms of sophistication and have increased in sheer number in recent years. This has led to corresponding developments in the methods used to evade the detection of phishing attacks, which pose daunting challenges to the privacy and security of the users of smart systems. This study uses LightGBM and features of the domain name to propose a machine-learning-based method to identify phishing websites and maintain the security of smart systems. Domain name features, often known as symmetry, are the property wherein multiple domain-name-generation algorithms remain constant. The proposed model of detection is first used to extract features of the domain name of the given website, including character-level features and information on the domain name. The features are filtered to improve the model's accuracy and are subsequently used for classification. The results of experimental comparisons showed that the proposed model of detection, which integrates two types of features for training, significantly outperforms the model that uses a single type of feature. The proposed method also has a higher detection accuracy than other methods and is suitable for the real-time detection of many phishing websites.

Keywords: phishing website detection; LightGBM; domain name feature; symmetry; feature engineering

Citation: Zhou, J.; Cui, H.; Li, X.; Yang, W.; Wu, X. A Novel Phishing Website Detection Model Based on LightGBM and Domain Name Features. *Symmetry* **2023**, *15*, 180. https://doi.org/10.3390/sym15010180

Academic Editors: Kuo-Hui Yeh and José Carlos R. Alcantud

Received: 9 November 2022
Revised: 25 December 2022
Accepted: 5 January 2023
Published: 7 January 2023

1. Introduction

With the continual development of the Internet and computer technology, network security issues are becoming increasingly important. Phishing attacks, which involve using fake websites to deceive users into obtaining their private information, cause significant losses to Internet users, financial institutions, and e-commerce companies [1]. The adversaries involved usually obtain domain names and build fake webpages that are replicas of legitimate websites [2]. Users access phishing websites via links sent by attackers and are often fooled into giving out such private information as their account passwords. According to the Anti-Phishing Working Group's (APWG) report for the second quarter of 2022, the number of phishing attacks more than quadrupled compared with early 2020 and reached 1,097,811 in June 2022. This was the highest monthly total in the history of the APWG'S reporting [3]. Thus, it is important to investigate methods to identify phishing websites.

Most Internet browsers currently in use have a function for blocking phishing websites. Filtering based on black-and-white lists of websites is one of the most widely used ones [4]. However, blacklist filtering methods are becoming increasingly ineffective due to the growth in the types and numbers of phishing websites as well as improvements in technology. Phishing websites usually imitate the structure of legitimate webpages to deceive users. To make the deceit even more confusing, phishing websites also use domain-name-generation algorithms, using dates, popular search terms, and other specific transformations to generate fake domain names for registration [5]. The methods of counterfeiting commonly used in the domains of phishing websites include character deletion, character duplication, sequence exchange, and character substitution [6,7].

Most methods to identify phishing websites use rich and easily recognizable features of webpages [8]. However, these features are too complex for the real-time detection of a large number of domain names. Although domain-name-generation algorithms are different, the overall idea of domain-name generation is similar, and the features extracted by the neural network model for different kinds of DGA domain names have certain symmetry and generality, which can effectively identify counterfeit domain names [9]. To cope with the increasing number of phishing activities, adapt to their current technological methods, and thus improve the efficiency of detection of phishing websites, this paper proposes a model based on LightGBM and domain names. The key contributions of this work are as follows:

(1) By considering the task of identifying phishing websites using a two-category processing model, we detail a framework to detect them.

(2) We explain the process of identifying phishing websites by using features of the domain name of the target website, and provide a method to optimize the model to ensure high detection accuracy.

The remainder of this article is structured as follows: Section 2 describes the current research on identifying phishing websites, and Section 3 details the proposed method. The results of experiments and an analysis of the proposed approach are given in Section 4, and the conclusions of this study are presented in Section 5.

2. Related Work

To confuse users, phishers generally imitate the URL of the target website to produce a phishing URL; stable features of the legitimate website, such as the URL's statistical feature, the webpage code feature, and the webpage text feature, will inevitably be disrupted [10,11]. Therefore, the current solution for the identification of most phishing websites is to first use a feature-extraction algorithm to extract the features of legitimate websites (the extracted features are usually symmetrical and universal) and then apply these features symmetrically to accurately identify phishing websites [12–14].

Based on the context and the density of the keywords, Altay et al. used three machine-learning-based methods to extract and analyze the features of words on pages to improve the accuracy of the detection of counterfeit pages [15]. Fang et al. extracted the images and characters from phishing websites and used the Monte Carlo algorithm to train a classification model to precisely identify phishing websites [16]. Chen et al. divided phishing websites into three categories based on similarity, used the wHash and SIFT mechanisms to evaluate website similarity, and used the Microsoft website dataset to test performance in terms of detection accuracy [17]. Cersosimo et al. used the Splunk Machine Learning Toolkit to detect malicious domains [18]. Phishing websites can be accurately identified by using the features of webpages, but this is not suitable for real-time detection and the identification of many domain names. To be more confusing to users, phishing websites often use strings similar to those of the corresponding legitimate websites as domain names.

Researchers have built models of detection according to the characteristics of characters used in phishing websites. Feroz et al. used the chi-square statistic and methods to assess the gain in information to extract 16 features of the vocabulary, including two-letter combinations and host-based features. They then used a machine-learning algorithm in Mahout to establish a reliable online learning framework to classify URLs. The results of K-fold cross-validation showed that the classifier is flexible [19]. Chatterjee et al. introduced a phishing detection technique based on deep reinforcement learning to identify phishing URLs. They used their model on a balanced, labeled dataset of benign and phishing URLs, extracting 14 hand-crafted features from the given URLs to train the proposed model [20]. Mvula et al. applied malicious domain name detection to COVID-19 and realized the classification of COVID-19 malicious domain names [21]. Liu et al. proposed a method based on the generalized Levenshtein distance to measure the visual similarity between domain names and applied the minimum line-of-sight method of search based

on triangle inequality and the locally sensitive hashing algorithm to improve its efficiency of searching [22]. Zouina et al. proposed a lightweight method of detection that is based entirely on URLs and that extracts six URL features. It used the SVM algorithm and recorded an accuracy of 95.80%. This method can be integrated into smartphones and tablets because of its resource efficiency [23]. Ozgur used a large amount of data from phishing and legitimate websites to propose a real-time anti-phishing system. The features used by the system included NLP features, word vectors, and hybrid features. The author compared seven NLP functions. The machine-learning classification algorithm found that the random forest algorithm based on NLP features delivered the best performance, with an accuracy of 97.98% [24]. Wang et al. analyzed the differences between the URLs of phishing websites and legitimate websites, defined primitives and sensitivity to describe the characteristics of the language used, calculated the similarity among the primitives of domain names, and then used the random forest algorithm to learn features of the language of the sub-domains to classify URLs. The algorithm had an accuracy of 95.6% and an average recognition time shorter than 1 s [25]. Yuan et al. proposed an improved BiGRU-Attention model that classified phishing websites based on the characteristics of the characters in their URLs. The model adequately learned the vector representing the domain name information and recorded an accuracy for classifying phishing websites at 99.55% [26].

In addition to features of the characters of the domain name, features such as WHOIS information play an important role in the detection of phishing websites. Sun Dandan extracted three kinds of features of phishing websites: lexical features, WHOIS features, and page-related features. The author proposed a brand-name-anomaly algorithm based on the edit distance to improve the J48 algorithm, based on Weka as a model to classify the features of URLs [27]. Aung et al. proposed a phishing-URL-detection model that used information-rich domain and path features [28]. They split URLs into two parts, domain and path, and assume that URL patterns in the domain are less random than those in the path. The domain part consists of the URL components until the end of the domain name, whereas the path part includes the rest of the URLs until the non-alphanumeric character. To reduce the false-alarm rate in the model for the identification of phishing websites based on machine learning, Alsariera et al. proposed three meta-learner models based on the forest penalty attribute algorithm. A weight-adjustment strategy was included in the model to construct an efficient decision tree. The lowest accuracy of the three models was 96.26%, the FAR value was 0.004, and the ROC value was 0.994 [29]. Mehanovic et al. used three K-nearest neighbor classifiers, a decision tree, and random forest to classify the features of websites obtained by the Weka feature selection method. This reduced the number of features used, thereby improving the model's classification efficiency. The time needed for classification was reduced from 2.88 s to 0.02 s, and the model recorded an accuracy of classification of 100% [30].

To avoid detection and blocking, phishing websites continually change their characteristics [31]. Some phishing websites generated by using the GAN network can avoid detection [32]. A model to identify such sites thus needs to be able to adapt to ensure the security of the network environment [33]. Altyeb proposed an intelligent ensemble learning approach for phishing website detection based on weighted soft voting to enhance the detection of phishing websites [34]. However, the time complexity of the detection method was not discussed.

Oram et al. proposed a LightGBM-based model for the identification of phishing websites [35]. The proposed model showed high performance accuracy and proved to be a robust approach for phishing activity. Li et al. proposed a stacking model combining the Gradient Boosting Decision Tree, XGBoost, and LightGBM algorithms to detect phishing web pages [36]. The authors extracted features from the URL and Hypertext Markup Language (HTML) of the suspicious website. The extracted features contained 8 URL- and 12 HTML-based features to generate a feature vector. The vector was fed to the stacked model for classification and achieved an accuracy of 97.30%. Chen et al. propose a graph-

based cascade-feature-extraction method based on transaction records and a lightGBM-based Dual-sampling Ensemble algorithm to detect phishing accounts based on blockchain transactions [37]. In the phishing website detection problem, we found that LightGBM was more efficient; thus, we selected it as our classification model.

3. Methodology

3.1. Framework of the Model

We extracted features of the domain name and used machine learning methods to implement our model to identify phishing websites. Figure 1 shows the overall framework.

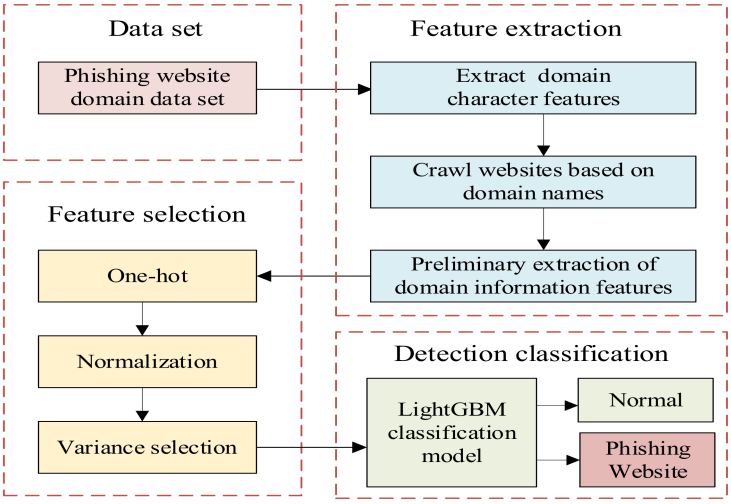

Figure 1. Framework of phishing website detection model.

The model is divided into four steps:

Step 1: Domain name dataset pre-processing. The domain name dataset that was used comes from the data published by the PhishTank website and the Alexa website. After obtaining the domain name data, clear the invalid data and check the duplicate of the two domain name datasets.

Step 2: Domain name feature extraction. Through the analysis of counterfeit domain names, two self-owned characteristics of domain names in the domain name dataset are extracted. The first feature is the domain name character feature. According to the character arrangement of domain names of phishing websites and normal websites, the model selects features that may improve the classification effect for extraction and preliminarily screens them according to the classification effect. In addition, in order to further improve the classification accuracy, this model adds domain name information features on the basis of domain-name character features. Domain name information features refer to the IP address and filing information of the domain name. Based on the domain name dataset, write a crawler program to crawl and query the domain name information published by the website, and further process the crawled data to obtain the domain name information characteristics. After that, the two types of features are integrated to obtain a feature dataset.

Step 3: Domain name feature selection. The features of different domains were relatively discrete, and there was no correlation between them. In addition, they cannot be inputted into the machine learning model LightGBM for training in the form of vectors. Thus, one-hot encoding was used to map the features into a computer-readable feature matrix and regularization was performed on them. One-hot encoding mainly used N-bit status registers to encode N states. Each state had an independent register bit, and only one bit was valid at any time. However, the existence of a large number of irrelevant, redundant or noisy features will not only bring about the problem of increasing the dimension but also directly affect the performance of the classifier. Therefore, in order to ensure the accuracy of the classification model and improve the classification efficiency, a variance

selection method was used to filter the feature matrix. By changing the parameters and conducting experiments for many times to determine the variance filtering threshold, more representative and comprehensive features of domain can be obtained.

Step 4: LightGBM recognition classification. The obtained domain name features are passed to LightGBM classifier for training, and finally the phishing website classification model is obtained, and the website classification results are output.

3.2. Feature Analysis

The features of the domain name used here can be obtained only by using known strings of domain names without obtaining information related to user privacy, such as traffic in the network. Features of the domain name can be divided into two categories according to the acquisition method: features of the characters used in the domain name and features of information on the domain name. The features of information on the domain name can be obtained through the corresponding website or other query websites to this end, whereas the features of the characters used in the domain name can be obtained through a local feature-extraction algorithm without visiting the website.

3.2.1. Features of the Characters Used in the Domain Name

Table 1 lists the domain names used by typical phishing websites. An analysis of the differences between the domain names of the phishing websites and the corresponding legitimate websites led to a total of 10 features of the characters used in the domain name in four categories. The four types of features were N-gram features, quantitative and matching features, maximum segmentation-related features, and edit distance.

Table 1. Some phishing website domain names.

Number	Domain Name
1	mazoon57168.uc.r.appspot.com (accessed on 12 June 2022)
2	alibaba.com.spatialsys.com.ru (accessed on 12 June 2022)
3	privacy.apple.com.info-sign.in (accessed on 12 June 2022)
4	www.nothingelsefilm.com (accessed on 12 June 2022)
5	paypal-limited.pdcotton.com (accessed on 12 June 2022)

There are certain differences between the character sequence of the domain name of a phishing website and that of a legitimate site. The N-gram feature can reflect this difference well. N-gram is a method of coding that is commonly used in natural language processing [38,39]. If the length of the text is l, the N-gram method can divide it into l+1−N continuous N-tuples, thereby retaining information on the word order of the text. The domain names of phishing websites may lead to extracted sequences that have a low probability of appearing in the domain names of legitimate sites. However, as N continues to increase, the feature vector space continues to increase as well, such that the feature matrix becomes increasingly sparse. Principal component analysis is used to reduce the dimensionality of the N-gram feature matrix and improve the model's classification efficiency.

The ratios of character composition, top-level domain names, and sensitive matching words of domain names of phishing websites are also different from those of the domain names of legitimate sites. The proposed model extracts vowels, numbers, single characters, and special characters from the domain name. Because domain names with shorter strings are easier to remember, they can be more easily registered early on given that the number of character combinations is small. A phishing website can usually register only a longer domain name. The hierarchical characteristics of the domain names are also extracted by the model because some domain names of phishing websites are disguised as those of legitimate websites by adding sub-domains to them. To steal users' identity and account information, the domain names of phishing websites may contain sensitive words such as "login" or "verify." The feature-extraction step involves extracting information on whether

the domain name contains such sensitive words. Different top-level domain names have different costs of registration and requirements of registrants. Some domain names of phishing websites often use cheap top-level domain names for registration and insert such strings as "com" into the domain name to give the illusion of being a well-known domain name with which users are familiar. The model thus extracts the characteristics of well-known domain names, including their type and location.

The smallest meaningful linguistic unit in English is called a morpheme. Phishing websites can construct their own domain names through word patching or by exploiting users' misspellings. The proposed model thus extracts the characteristics of morphemes used in the domain name as well. The steps of the extraction algorithm are shown in Figure 2. The final extracted features include the minimum number of divisions of the domain name, the length of each part of the division, and the ratio of misspellings.

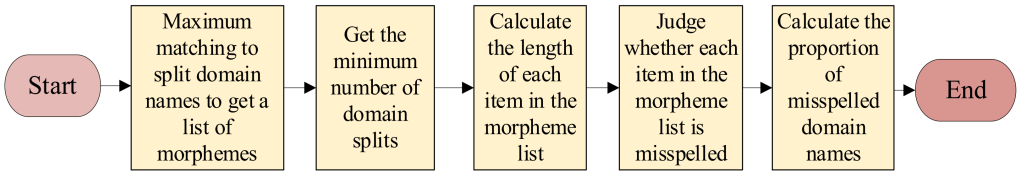

Figure 2. Morpheme-related-feature extraction algorithm flowchart.

The smallest number refers to the number of letters in a word for the match to be as long as possible if the domain name is divided into readable parts; for example, in the case of a domain name for www.southvalleypeacecenter.org\T1\textquotedblright (accessed on 18 June 2022), after break up, this domain is broken up into "WWW", "south", "valley", and "peace". The minimum partition number of "center" and "org" is six. Here, the length of each segment is denoted as {3, 5, 6, 5, 6, 3}.

To masquerade as a legitimate website, phishing websites generate domain names by adding, deleting, and replacing characters based on the domain names of legitimate sites. The proposed model thus extracts the Levenshtein distance of the domain names. This refers to the minimum number of single-character editing operations required to convert domain name "a" into "b". It reflects the likelihood of the test domain name being a counterfeit of the domain name of a legitimate website. The Levenshtein distance between domain names a and b can be described by Formula (1):

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \min(i,j) = 0 \\ \min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise \end{cases} \quad (1)$$

where $lev_{a,b}(i,j)$ refers to the distance between the first i characters of a and the first j characters of b.

Counterfeit domain names usually maintain a small editing distance from normal domain names. When the length of domain names is short, such as "qq.com", the editing distance between it and other short domain names is also small. If the similarity-related feature of domain name editing distance is only used for judgement, normal domain names with short domain names will be misjudged as counterfeit domain names. Therefore, this feature must be combined with domain name length characteristics.

By analyzing the difference between fake domain names and legitimate domain names, 10 characters of domain names in four categories are extracted. The four types of features are the N-gram feature, quantitative feature and matching feature, maximum segmentation correlation feature, and edit distance feature.

3.2.2. Features of Information on the Domain Name

The domain name of the website contains a large amount of information, such as WHOIS information and filing information. This can be used to determine the registration-related background of the domain name to infer the credibility of the website. WHOIS information is used to check whether a domain name is registered. If it is, detailed information on it, including the registrar, owner, registration date, and expiration date can be obtained. Ip138.com (accessed on 18 June 2022) and ip.tool.chinaz.com (accessed on 18 June 2022) provide a query service for information on domain names. We used Python to write a crawler to obtain information on domain names provided by these websites for the dataset of domain names. The steps of information extraction are shown in Figure 3.

We set a cookie and visited the relevant website to obtain the source code of the query page. We analyzed the structure of the source code to find fields containing information on the domain name. According to the different formats of the fields, we used the Beautiful Soup HTML parser and regular expressions to separately match each field to crawl information on the domain name. The information initially obtained by the crawler contained 12 items: IP address, physical location, registrar, person to contact, contact email, contact number, update time, creation time, expiration time, company, Domain Name Server, and filing information.

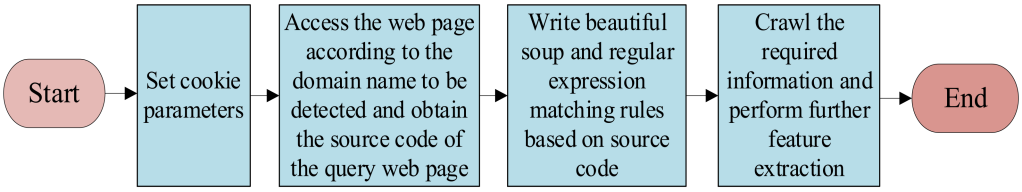

Figure 3. Domain-name information-extraction crawler program steps.

Because the attacker may have registered multiple domain names simultaneously, the location-related information of these domain names may be similar. WHOIS information is often missing in the registration information of domain names of phishing websites. The completeness of such information as the names of human contacts and their telephone numbers can thus be used to verify the security of the relevant domain name to some extent. Similarly, the presence of registration information may reflect the security of the domain name. Because they are frequently blocked by security personnel, phishing websites often need to change their domain names or IP addresses. Many phishing websites have a short duration of registration, which has thus become a factor in identifying phishing websites. We also analyzed and sorted the information obtained by the crawler program.

The final features of information on the domain name include address-related information, complete WHOIS information, time-related information, and filing information.

3.3. LightGBM

LightGBM is an additive model composed of multiple trees [40]. The model uses the negative gradient of the loss function to replace the residuals as the basis for generating a decision tree. While ensuring classification accuracy, it has a high training speed, takes up little memory, and can handle large-scale data. For a given dataset, the LightGBM model and its objective function are as follows:

$$\hat{y}_i = \sum_{t=1}^{K} f(x_i) \tag{2}$$

$$Obj^{(t)} = \sum_{t=1}^{K} (l(y_i, \hat{y}_i{}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i {f_t}^2(x_i)) + \Omega(f_t) \tag{3}$$

where f is the decision tree, $\hat{y}_i$ is the predicted value, l is the loss function, g_i is the first derivative of the loss function, h_i is its second derivative, and Ω is a regular term used to express the complexity of the model.

To optimize the objective function, the regular term is expressed as:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} {w_j}^2 \tag{4}$$

where T is the number of leaf nodes of the tree and w_i is the output of each leaf node.

The objective function is then rewritten as an expression related to T and w_i. We take the partial derivative of the objective function with respect to w_i. If its derivative is zero, the objective function yields the minimum value. We define the sample set on each leaf node j as $I_j = \{i|q(x_i) = j\}$ and substitute the obtained value of w_i into the objective function. The result is as follows:

$$w_j = -\frac{G_j}{H_j + \lambda} \tag{5}$$

$$Obj^{(t)} = -\frac{1}{2}\sum_{j=1}^{T}\left(\frac{{G_j}^2}{H_j + \lambda}\right) + \gamma T \tag{6}$$

where $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$.

Compared with the level-wise method, the leaf-wise method adopted by LightGBM makes tree generation more efficient. As shown in Figure 4, the node with the largest split gain among the leaf nodes is selected as the next split leaf node until the decision tree has grown appropriately. When splitting the tree the same number of times, the leaf-wise method can reduce errors and improve accuracy to a greater extent than the level-wise method. However, it can make the decision tree too deep, resulting in overfitting. Therefore, LightGBM adds a maximum depth limit to prevent overfitting while ensuring high efficiency.

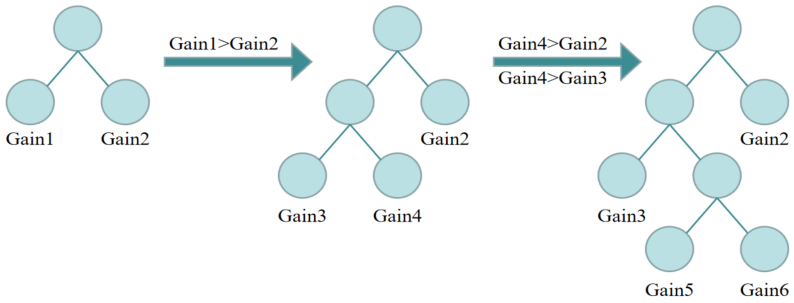

Figure 4. Leaf-wise growth strategy.

LightGBM uses histogram optimization to divide continuous values into a series of discrete domains to simplify the expression of the data and reduce the required memory. Histogram regularization also enables the model to avoid overfitting and yield better generalization results.

To reduce the size of the dataset and the feature set, LightGBM uses two algorithms: gradient-based one-sided sampling (GOSS) and exclusive feature bundling (EFB). GOSS uses samples with large and small gradients to calculate the information gain while remaining as consistent as possible with the overall data distribution and ensuring that samples with small gradient values are trained. EFB bundles mutually exclusive features to reduce the number of feature dimensions and improve computational efficiency. The conflict ratio can be used to measure the degree of non-exclusion of two features that are not completely mutually exclusive.

4. Results and Discussion

The experimental equipment included an Intel(R) Core(TM) i7-8565U CPU with 8 GB of RAM and a 64-bit Windows 10 operating system. The model was implemented in Python 3.7.4 (Python Software Foundation; Delaware, USA).

4.1. Experimental Data

The dataset of domain names used for model training was taken from publicly available data on the Alexa website (https://alexawebsitedesigns.com/) (accessed on 20 June 2022) and the PhishTank website (https://phishtank.org/) (accessed on 20 June 2022). The domain names were divided into two types: the domain names of legitimate websites and the domain names of phishing websites. The number of visits can reflect the credibility of the domain name to a certain extent. Compared with domain names of phishing websites, the normal domain name has a longer lifecycle and more visits. Therefore, we selected 12,000 domain names in the Alexa dataset with the most visits as the dataset of domain names of legitimate websites. Due to the efforts of security personnel, many phishing websites are banned in a short time at the beginning of their life. It is not easy to obtain many phishing website domain names. PhishTank is an authoritative website for publishing information about phishing websites, which users can use to submit, verify, track, and share phishing data. The website provides information on confirmed, unconfirmed, and active and inactive phishing websites. Unconfirmed records cannot determine that the website is a phishing website, and some inactive website cannot query the domain name information characteristics. We chose 12,000 pieces of information on confirmed and active phishing websites from it as the data source for the domain names of phishing websites. There were 24,000 instances, and phishing websites and legitimate websites were randomly split into 70 percent for training and 30 percent for testing, as shown in Figure 5.

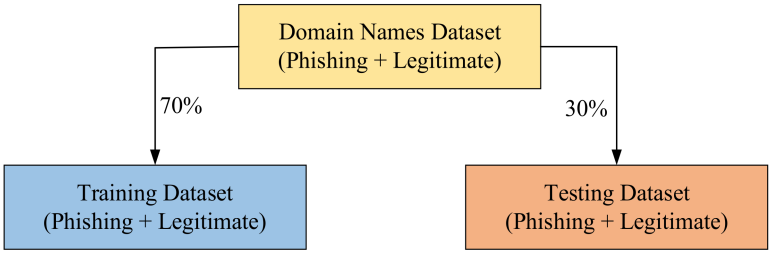

Figure 5. Dataset partition.

4.2. Evaluation Indicators

Accuracy, precision, recall, and the F value ($F1$) are commonly used indicators to assess machine-learning-based methods. Accuracy indicates the ratio of correct predictions; recall represents the correct prediction proportion of all positive data. $F1$ is the harmonic average of precision and recall. Therefore, if the precision and recall are better, the $F1$ value is also higher. We used these four indicators to evaluate our model. Their formulae are as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{10}$$

where TP is the number of positive classes predicted as positive classes, TN is the number of negative classes predicted as negative classes, FP is the number of negative classes predicted as positive classes, and FN is the number of positive classes predicted as negative classes. These four parameters together form a confusion matrix.

4.3. Influence of Model Parameters on Experimental Results

The LightGBM model has many parameters. We used a grid search to find the most suitable parameters to obtain the optimal classifier. Table 2 lists some of the experimental results. When the learning rate of the LightGBM model was lower than 0.1, the classification effect of the model would gradually increase with the increase of the learning rate. When the learning rate exceeded 0.1, the classification effect of the model would generally decline with the increase of the learning rate. When the number of estimators was less than 80, the model classification effect would increase with the increase of the number of estimators. When the number of estimators exceeded 80, the model classification effect would remain roughly unchanged. When the number of leaves was 40, the model had the best classification effect. The preset values of each model parameter were determined through several experiments. The learning rate of the proposed classification model was 0.1, the number of estimators was 80, the number of random seeds was 1000, and the maximum depth was 40.

Table 2. The influence of different parameters on detection results.

Serial Number	[Learning Rate, Estimators, Maximum Depth]	Accuracy (%)	Precision (%)	Recall (%)	*F*1 (%)
1	[0.01,60,60]	92.52	93.66	91.21	92.42
2	[0.05,60,60]	92.92	93.81	91.90	92.85
3	[0.1,60,60]	93.60	94.07	93.07	93.57
4	[0.2,60,60]	93.64	94.47	92.71	93.58
5	[0.3,60,60]	93.57	94.04	93.04	93.54
6	[0.5,60,60]	93.64	94.02	93.21	93.61
7	[1,60,60]	88.16	89.35	86.64	87.98
8	[0.2,70,60]	93.63	93.95	93.27	93.61
9	[0.2,80,60]	93.70	94.08	93.27	93.67
10	[0.2,90,60]	93.68	94.10	93.21	93.65
11	[0.2,100,60]	93.57	94.01	93.07	93.54
12	[0.2,110,60]	93.68	94.18	93.13	93.65
13	[0.2,120,60]	93.54	94.03	92.99	93.51
14	[0.2,130,60]	92.92	93.81	91.90	92.85
15	[0.2,80,50]	93.71	94.35	92.99	93.67
16	[0.2,80,40]	93.93	94.40	93.41	93.90
17	[0.2,80,30]	93.88	94.57	93.10	93.83
18	[0.2,80,20]	93.68	94.43	92.85	93.63
19	[0.2,80,10]	93.45	94.35	92.43	93.38

Through the analysis of the characteristics of counterfeit domain names, the character characteristics and information characteristics of domain names were extracted preliminarily, and the dimension of the characteristic matrix reached 78. When the dimension of the feature matrix was too large, the training speed of the model decreased. Table 3 shows the relationship between the feature dimension and the training time. Therefore, we tried to select as few features as possible to train the model while ensuring its accuracy. Variance selection was used to filter the features once they had been normalized. Figure 6 shows the relationship between the selected feature dimensions and their effect on the accuracy of the classification of the model. It can be seen that with the reduction in the dimension, the calculation of the model is reduced, and the training speed is improved. However, using fewer than 22 dimensions eliminated many features that contributed significantly to the classification, and the model's accuracy decreased. Therefore, to ensure the efficiency and accuracy of the model, we used 22 features. They were filtered to reduce the size of the feature matrix and improve the efficiency of training and detection.

Table 3. Relationship between feature dimension and the training time.

Number	Feature Dimension	Training Epoch/s
1	78	256
2	70	148
3	50	97
4	30	64
5	10	36

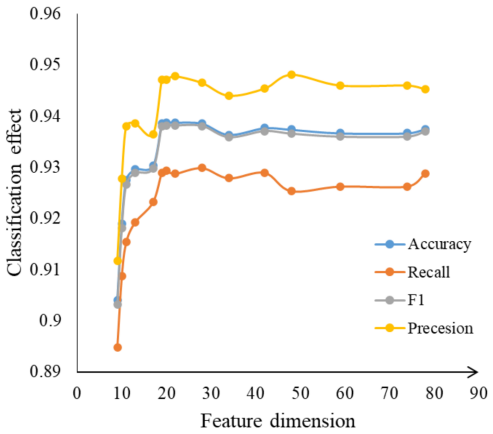

Figure 6. Relationship between feature dimension and the classification effect.

Feature normalization merges the features with similar attributes, and the variance selection method filters the features with less effect on model training. Finally, the LightGBM model selects 16 features with significant contributions from the original eight domain-name feature types. They are listed in Table 4. The first four categories contained features of the characters used in the domain names of websites, and the last four contained features of information on the domain name.

Table 4. Domain Name Features.

Number	Feature Category	Feature
1	N-gram	2-gram sequence matrix
2	Quantitative feature and matching feature	Domain name character length Percentage of numbers in domain names The number of sensitive words in the domain name Top-level domain location Types of top-level domains
3	Maximum segmentation related features	Maximum number of domain matching splits Maximum domain split length Number of misspelled divisions
4	Edit distance	Edit distance
5	Address	IP Physical location
6	Time	Update time Existence time
7	WHOIS	Completeness of WHOIS Information
8	Filling	Whether to file

4.4. Comparative Analysis of Experimental Results

The experimental results showed that the overall training accuracy of the LightGBM classifier was 93.88%, with a precision of 94.78%, a recall of 92.88%, and an F value of 93.82%, which means that our model was effective in both the training set and the test set for detecting malicious domain names and had a high recognition rate. The F value is the harmonic mean of the accuracy and recall rate. The higher F value also demonstrates the superiority of this model. To show the impact of the two features on the model's performance more intuitively, we assessed the model trained using only features of the characters used in the domain name with that trained on only features of information on the domain name in terms of the four evaluation indicators. The results are shown in Figure 7.

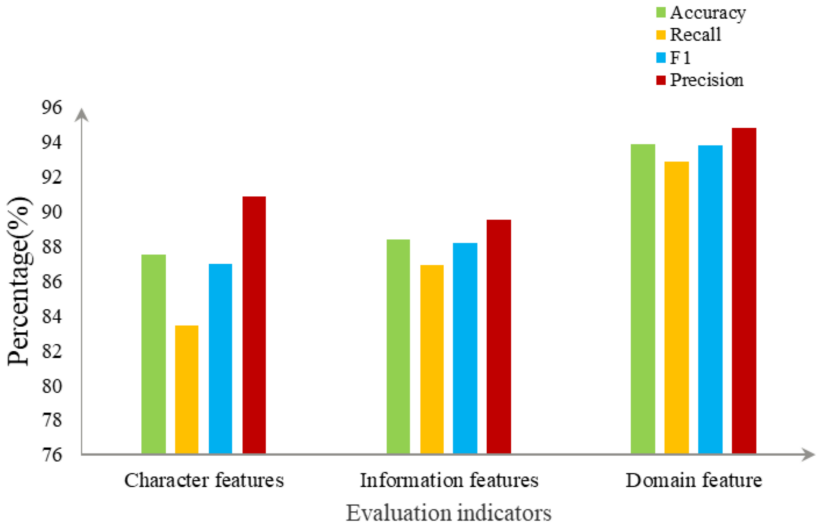

Figure 7. The relationship between feature types and detection effects.

The accuracy of the model trained using only features of the characters used in the domain name was 87.51%, and that of the model trained using only features of information on the domain name was 88.36%. Their respective accuracies of detecting phishing websites model using two domain name features were 6.37 and 5.52% higher than that of the model that used only a single feature. The domain feature model also performed well on the other three evaluation indicators, with increases in precision of 3.92 and 5.29%, those in recall of 9.46% and 5.96%, and those in the F value of 6.84 and 5.63%, respectively, over the single-feature model. The features of characters in the domain name complemented those of information on the domain name to more comprehensively reflect the differences between the domain names of phishing websites and legitimate websites. This led to better performance of the model as a whole.

GBDT, AdaBoost, XGBoost, and SVM are often used in classification problems because of their excellent performance. We compared the performance of the proposed LightGBM model in terms of identifying phishing websites with these four methods. The results are shown in Figure 8. It is clear that the LightGBM model delivered the best performance in terms of the four evaluation indicators. AdaBoost performed relatively well, with accuracy, precision, recall, and the F value all above 90%. The SVM was slightly worse, with values from 80 to 90% on the four indices.

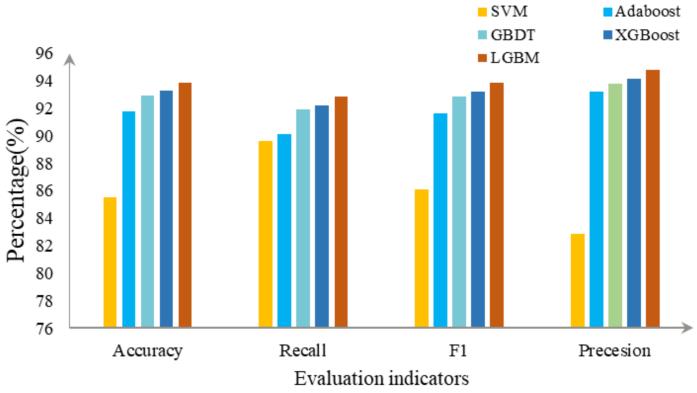

Figure 8. Detection efficiency of different models.

The GOSS and EFB algorithms used by the LightGBM model improved its training speed and are suitable for use with large amounts of data.

In this section, we compare our model to those of existing works that use feature learning for the classification of phishing websites. Like other researchers (Chatterjee et al. [20], Aung et al. [28], and DElightGBM. [37]), the results of existing works are collected from the respective papers for the comparison analysis. The listed results in Table 5 are the results obtained by respective authors with their datasets. These researchers' datasets could not be used for comparison because of the limitation of feature extraction. Aung et al. [28] has higher recall than our model, but our model achieves the highest precision and F1. This is because the detection process of our approach relies on the domain name features and those of information on the domain name, which are obtained from multiple aspects and have more information than the features from a single aspect.

Table 5. Comparison of proposed model and other approaches.

Approaches	Precision/%	Recall/%	*F*1/%
Chatterjee et al. [20]	86.71	88.00	87.30
Aung et al. [28]	92.68	94.21	93.76
DElightGBM [37]	81.96	80.50	81.22
Our model	94.78	92.88	93.82

5. Conclusions

According to the symmetry of domain name features, this paper proposes a phishing website recognition model based on LightGBM. We tested the model on data from PhishTank for testing. The features of the domain names of phishing websites were divided into features of the characters used in the domain name and features of information on the domain name and were extracted separately. Once they had been filtered, 16 features of the domain name were finally selected for model training. The grid-search method was used during training to optimize the parameters of the LightGBM model. We compared the performance of different models with the proposed one. The results showed that the model that used features of the domain name for training was significantly superior to the model that used only a single feature for training, with increases of 5% in terms of accuracy, precision, recall, and the F value. The proposed LightGBM model also outperformed the GBDT, AdaBoost, XGBoost, and SVM models.

The proposed phishing website recognition method extracts features manually according to experience. In future work, neural networks and other methods can be used as alternatives to reduce the manual workload in the model construction process and improve the method's automation.

Author Contributions: Conceptualization, J.Z. and W.Y.; methodology, H.C.; software, X.L. and X.W.; validation, J.Z., H.C., and X.W.; formal analysis, X.L.; investigation, X.L.; resources, W.Y.; data curation, H.C.; writing—original draft preparation, W.Y. and X.L.; writing—review and editing, J.Z. and X.W.; visualization, X.L.; supervision, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Project of Civil Aviation Safety Capacity under Grant PESA2019074, and Grant PESA2021009, and in part by the fundamental research funds for the central universities under No. 3122018C036.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data generated or appeared in this study are available upon request by contact with the corresponding author. Furthermore, the models and code used during the study cannot be shared at this time as the data also form part of an ongoing study.

Acknowledgments: We thank the assistant editor and the anonymous reviewers for their useful feedback that improved this paper.

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

References

1. Rao, R.S.; Vaishnavi, T.; Pais, A.R. CatchPhish: Detection of phishing websites by inspecting URLs. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 813–825. [CrossRef]
2. Hamroun, C.; Amamou, A.; Haddadou, K.; Haroun, H.; Pujolle, G. A review on lexical based malicious domain name detection methods. In Proceedings of the 6th Cyber Security in Networking Conference (CSNet), Rio de Janeiro, Brazil, 24-26 October 2022; pp. 1–7. [CrossRef]
3. APWG. Phishing Activity Trends Report, 2nd Quarter 2022. Available online: http://apwg.org/trendsreports (accessed on 20 September 2022).
4. Prakash, P.; Kumar, M.; Kompella, R.R.; Gupta, M. Phishnet: Predictive blacklisting to detect phishing attacks. In Proceedings of the IEEE Information Communications, San Diego, CA, USA, 14–19 March 2010; Volume 5, pp. 1–5.
5. Mac, H.; Tran, D.; Tong, V.; Nguyen, L.G.; Tran, H.A. DGA botnet detection using supervised learning methods. In Proceedings of the Eighth International Symposium on Information and Communication Technology, Nha Trang, Vietnam, 7–8 December 2017; pp. 211–218.
6. Mohamed, Y.E.; Ahmad, S.A. A mobile sensing method to counteract social media website impersonation. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 25–30.
7. Agten, P.; Joosen, W.; Piessens, F.; Nikiforakis, N.; Leuven, D.K. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 8–11 February 2015; Volume 2, pp. 8–11.
8. Banerjee, A.; Rahman, M.S.; Faloutsos, M. SUT: Quantifying and mitigating URL typosquatting. *Comput. Netw.* **2011**, *55*, 3001–3014. [CrossRef]
9. Hu, P.C.; Diao, L.L.; Ye, H.; Yang, Y.L. DGA domains detection based on artificial and depth features. *Comput. Sci.* **2020**, *47*, 11–317.
10. Almousa, M.; Zhang, T.; Sarrafzadeh, A.; Anwar, M. Phishing website detection: How effective are deep learning-based models and hyperparameter optimization? *Secur. Priv.* **2022**, *5*, e256. [CrossRef]
11. Zhao, H.; Chen, Z.; Yan, R. Malicious Domain Names Detection Algorithm Based on Statistical Features of URLs. In Proceedings of the 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hangzhou, China, 4–6 May 2022; pp. 11–16.
12. Almomani, A.; Alauthman, M.; Shatnawi, M.; Alweshah, M.; Alrosan, A.; Alomoush, W.; Gupta, B. Phishing website detection with semantic features based on machine learning classifiers: A comparative study. *Int. J. Semant. Web Inf. Syst.* **2022**, *18*, 1–24. [CrossRef]
13. Do, N.; Selamat, A.; Krejcar, O.; Herrera, E.; Fujita, H. Deep learning for phishing detection: Taxonomy, current challenges and future directions. *IEEE Access* **2022**, *10*, 36429–36463. [CrossRef]
14. Pan, R.; Chen, J.; Ma, H.; Bai, X. Using Extended Character Feature in Bi-LSTM for DGA Domain Name Detection. In Proceedings of the 2022 IEEE/ACIS 22nd International Conference on Computer and Information Science (ICIS), Zhuhai, China, 26–28 June 2022; pp. 115–118.
15. Altay, B.; Dokeroglu, T.; Cosar, A. Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection. *Soft Comput.-Fusion Found. Methodol. Appl.* **2018**, *23*, 4177–4191. [CrossRef]
16. Feng, F.; Zhou, Q.G.; Shen, Z.B.; Yang, X.H.; Han, L.H. The application of a novel neural network in the detection of phishing websites. *J. Ambient. Intell. Humaniz. Comput.* **2018**, 1–15. [CrossRef]
17. Chen, J.; Ma, Y.; Huang, K. Intelligent visual similarity-based phishing websites detection. *Symmetry* **2020**, *12*, 1681. [CrossRef]
18. Cersosimo, M.; Lara, A. Detecting malicious domains using the splunk machine learning toolkit. In Proceedings of the 2022 IEEE/IFIP Network Operations and Management Symposium (NOMS), Budapest, Hungary, 25–29 April 2022; pp. 1–6.

19. Feroz, M.N.; Mengel, S. Examination of data, rule generation and detection of phishing URLs using online logistic regression. In Proceedings of the IEEE International Conference on Big Data, Santa Clara, CA, USA, 29 October–1 November 2015; Volume 12, pp. 241–250.
20. Chatterjee, M.; Namin, A. Detecting phishing websites through deep reinforcement learning. In Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, 15–19 July 2019; pp. 227–232. [CrossRef]
21. Mvula, P.K.; Branco, P.; Jourdan, G.V.; Viktor, H.L. COVID-19 malicious domain names classification. *Expert Syst. Appl.* **2022**, *204*, 117553. [CrossRef]
22. Liu, T.; Zhang, Y.; Shi, J.; Jing, Y.; Li, Q.; Guo, L. Towards quantifying visual similarity of domain names for combating typosquatting abuse. In Proceedings of the Military Communications Conference, Baltimore, MD, USA, 1–3 November 2016; Volume 11, pp. 770–775.
23. Zouina, M.; Outtaj, B. A novel lightweight URL phishing detection system using SVM and similarity index. *Hum.-Cent. Comput. Inf. Sci.* **2017**, *7*, 17. [CrossRef]
24. Ozgur, K.; Buber, E.; Demir, O.; Diri, B. Machine learning based phishing detection from URLs. *Expert Syst. Appl.* **2018**, *117*, 345–357.
25. Wang, Y.; Liu, B.; Lin, G. Phishing detection algorithm based on language features of URL. *Comput. Eng. Appl.* **2019**, *26*, 11–17.
26. Yuan, L.J.; Zeng, Z.Y.; Lu, Y.K.; Ou, X.F.; Feng, T. A character-level BiGRU-attention for phishing classification. *Inf. Commun. Secur.* **2019**, *12*, 746–762.
27. Sun, D. Research on Phishing Detection Mechanism by Integrating New URL Features. Master's Thesis, Southwest Jiaotong University, Chengdu, China, 2017.
28. Aung, E.; Yamana, H. Phishing URL Detection Using Information-Rich Domain and Path Features. In Forum on Data Engineering and Information Management. 2021. Available online: https://proceedings-of-deim.github.io/DEIM2021/papers/I21-1.pdf (accessed on 22 July 2022).
29. Alsariera, Y.; Adeyemo, V.; Balogun, A. Phishing website detection: Forest by penalizing attributes algorithm and its enhanced variations. *Arab. J. Sci. Eng.* **2020**, *45*, 10459–10470. [CrossRef]
30. Mehanovic, D.; Kevric, J. Phishing website detection using machine learning classifiers optimized by feature selection. *Traitement Signal* **2020**, *37*, 563–569. [CrossRef]
31. Fernando, M.; Arachchilage, N. Why johnny can't rely on anti-phishing educational interventions to protect himself against contemporary phishing attacks? In Proceedings of the Australasian Conference on Information Systems, Perth, Australia, 9–11 December 2019; Volume 12, pp. 395–405.
32. Aleroud, A.; Karabatis, G. Bypassing detection of URL-based phishing attacks using generative adversarial deep neural networks. In Proceedings of the CODASPY '20: Tenth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, 16–18 March 2020; Volume 3, pp. 53–60.
33. Yang, P.; Zhao, G.Z.; Zeng, P. Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access* **2019**, *7*, 15196–15209. [CrossRef]
34. Taha, A. Intelligent ensemble learning approach for phishing website detection based on weighted soft voting. *Mathematics* **2021**, *9*, 2799. [CrossRef]
35. Oram, E.; Dash, P.; Naik, B.; Nayak, J.; Vimal, S.; Nataraj, S. Light gradient boosting machine-based phishing webpage detection model using phisher website features of mimic URLs. *Pattern Recognit. Lett.* **2021**, *152*, 100–106. [CrossRef]
36. Li, Y.K.; Yang, Z.G.; Chen, X.; Yuan, H.P.; Liu, W.Y. A stacking model using URL and HTML features for phishing webpage detection. *Future Gener. Comput. Syst.* **2019**, *94*, 27–39. [CrossRef]
37. Chen, W.L.; Guo, X.F.; Chen, Z.G.; Zheng, Z.B.; Lu, Y.T. Phishing scam detection on Ethereum: Towards financial security for blockchain ecosystem. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence, Yokohama, Japan, 11–17 July 2020; pp. 4506–4512.
38. Yazdani, A.; Safdari, R.; Golkar, A.; Sharareh, R.; Niakan, K. Words prediction based on N-gram model for free-text entry in electronic health records. *Health Inf. Sci. Syst.* **2019**, *7*, 6. [CrossRef] [PubMed]
39. Wang, H.T.; He, J.; Zhang, X.H.; Liu, S.F. A short text classification method based on N-gram and CNN. *Chin. J. Electron.* **2020**, *29*, 248–254. [CrossRef]
40. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Liu, T. LightGBM: A highly efficient gradient boosting decision tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 3149–3157.