Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 2993877, 7 pages
https://doi.org/10.1155/2021/2993877

Research Article

High Efficiency Spam Filtering: A Manifold Learning-Based Approach

Chao Wang, Qun Li, Tian-yu Ren, Xiao-hu Wang, and Guang-xin Guo

State Grid Beijing Electric Power Company Electric Power Research Institute, Beijing, China

Correspondence should be addressed to Chao Wang; 29681987@qq.com

Received 29 June 2021; Revised 15 July 2021; Accepted 23 July 2021; Published 9 August 2021

Academic Editor: Jie Chen

Spam filtering, which refers to detecting unsolicited, unwanted, and virus-infested emails, is a significant problem because spam emails lead to unnecessary costs of Internet resources, waste of people's time, and even loss of property. Support vector machine (SVM) is the state-of-the-art method for high accuracy spam filtering. However, SVM incurs high time complexity because of the high dimensionality of the emails. In this study, we propose a manifold learning-based approach for time-efficient spam filtering. From the experiments that most of the features are not decisive, we can obtain the viewpoint that only a minor part of the spam emails can be detected using the nondecisive features. Based on the insight, we propose to employ the Laplace feature map algorithm to obtain the geometrical information from the email text datasets and extract the decisive features. Then, the extracted features are used as the input of SVM to spam filtering. We conduct extensive experiments on three datasets, and the evaluation results indicate the high accuracy time efficiency of our proposed algorithm.

1. Introduction

Email became a popular and widely adopted method in the Internet era since the 1960s for communication, advertisement, and account registration. Spam emails are defined as unsolicited, unwanted, or virus-infested emails [1, 2]. Based on the statistics from Spamlaws, nearly 85% of all emails are spam, in which the advertising, adult-related, and unwanted emails make up 36%, 31.7%, and 26.5% of the content, respectively [3]. Spam filtering, which refers to the process of detecting spam emails, is critical because spam emails are very cheap to send but have severe consequences such as annoying the recipients, wasting the Internet resources, and even leading to loss of property [4]. In spam filtering, nonspam emails should never be classified as spam because the misclassified emails can be critical for the users, which bring significant challenges [5].

The spam filtering methods can be divided into two complementary categories, i.e., origin-based and content-based [6]. In origin-based methods, the senders of the emails are classified as trusted, unknown, and spammer based on the IP addresses, email addresses, allowlists, and blocklists [7]. Emails from trusted senders and spammers will be directly classified as nonspam and spam, respectively. As for emails from unknown senders, they will be further filtered via content-based methods. This study focuses on the content-based approach in which the classification is purely based on the email content, i.e., header and body.

In the early stage, content-based approaches are mainly based on the statistics of words and phrases in spam and nonspam emails [8]. For example, more than 99% of the emails containing some words and phrases, such as "act now," "offer expires," and "winning," are spam [9]. A spam filter incorporating such statistics is called a Bayesian filter, which classifies the emails by going through the content word by word and phrase by phrase. The advantage of the Bayesian filter is that the classification accuracy can be improved when more data are collected from the users. However, Bayesian filters fail to consider the relationship among the words and phrases, resulting in limited accuracy.

Recently, machine learning becomes popular in content-based spam filtering [10]. Support vector machine (SVM) is one of the successful and cutting-edge techniques achieving higher accuracy than Bayesian filters [11]. SVM embeds the email content into a vector space and separates the emails into two classes, i.e., spam and nonspam, using a hyperplane in the vector space. The secret of SVM lies in the comprehensive embedding of the email content and separation using a hyperplane. The embedding of the email content incorporates the complex relationship among the words and phrases, which is a comprehensive content representation. The hyperplane separation maximizes the margin between the email embeddings and the hyperplane, making the SVM method robust in spam filtering.

However, embedding of the email content is nontrivial [12–14]. If few features are embedded, the spam filter will incur low classification accuracy; if too many features are embedded, training and applying the spam filter will incur high time overhead. A natural question comes as follows: is it possible to select few features to guarantee high accuracy for SVM-based spam filtering?

In this study, we propose a manifold learning-based approach to select the distinctive features and feed the features to the SVM model for time efficiency and accurate spam filtering. In particular, we gain the insight through experiments that most of the features used in traditional SVM-based spam filtering approaches are not decisive, using which only a minor part of the spam emails can be detected. Based on the insight, we employ an adapted manifold learning algorithm to select the decisive features. Then, the features are fed into the classic SVM model for spam filtering. In this way, our method only selects a small number but decisive features for spam filtering, which provides both high accuracy and time efficiency.

The main contributions of this study are as follows:

(i) We propose an adapted manifold learning approach to extract the decisive features for spam filtering. The features can not only be used in SVM but also other machine learning-based spam filtering algorithms.

(ii) We propose a time-efficient SVM-based approach that takes the decisive features as input and filters spam emails

(iii) We extensively evaluate the proposed spam filtering algorithms, and the experimental results indicate the high accuracy and time efficiency of the proposed method.

The rest of the study is organized as follows. Section 2 presents the related work. Section 3 introduces the proposed method for spam filtering using manifold learning and SVM. Section 4 illustrates the time complexity analysis and demonstrates extensive experimental results. Finally, Section 5 concludes the study with future directions.

2. Related Work

This section presents the related work on machine learning-based spam filtering in Subsection 2.1 and manifold learning in Subsection 2.2.

2.1. Spam Filtering Algorithm. Currently, in the field of spam filtering, the traditional machine learning sorting algorithms include decision-making trees [15], SVM [16], and Bayesian classifiers. A decision-making tree is a learning algorithm for sorting out datasets based on a tree-like structure. The tree structure includes root nodes and child nodes, representing different attributes of datasets. To form a tree structure is meant to determine the position of different attributes in the decision-making tree, which serves as the learning assignment for the algorithm. Carreras et al. [17] use the decision-making tree model to sort out spam emails, a practice not widely applied to spam filtering for the fact that the attributes of spam emails are hard to be defined. SVM classifiers, however, have pretty wide applications to spam filtering. Its main goal is to learn a linear hyperplane for linear divisible sample point sets and make the sample sets under a given category placed on the one side of the hyperplane while different categories of sample points on the other sides. In training an SVM classifier, only several sample points closest to the linear hyperplane are relevant to model training, and the remaining sample point sets will not work during the training process. Therefore, those several vector-represented sample points that lie closest to the hyperplane are called support vectors. Sculley et al. [18] integrate SVM with the online learning model to filter the spam email. Renuka et al. [19] add latent semantic information in the text message to classification and sort out spam emails using the SVM model.

Undeniably, there are other machine learning classifiers applied to email sorting research, such as ensemble learning algorithm [17], naive Bayesian classifier [20, 21], and reinforcement learning [22, 23]. For small email corpora, some conventional classifier algorithms work effectively in spam filtering. As for massive text corpora, typical machine learning methods are incapable of handling a large amount of text data. Such a backdrop has allowed deep learning techniques to be extensively applied to spam filtering. The deep learning techniques treat an email as a piece of text data from which keywords are extracted for spam identification. Tzortzis et al. [24] initiated a deep learning model for spam filtering, and an autoencoder is employed to detect spams [25].

Although deep learning is emerging and can be applied in spam filtering, deep learning models are not widely accepted by academia and industries for the following two reasons. First, deep learning models are rarely explainable. In spam filtering, nonspam emails should never be classified as spam because the misclassified emails can be critical for the users. Deep learning models can hardly explain the misclassification and are generally not adopted for spam filtering [26]. Second, the training and inference of deep learning models demand a large number of resources (power). The email service providers want to reduce the cost, and deep learning models are not employed [27].

2.2. Manifold Learning. In terms of text classification, such as spam filtering, representation or embedding of the text data is essential to enhance the classification performance. Text representation converts the text data into vector representations that contain necessary information without redundancy and noise. As a result, before training a classifier, we need to preprocess the text dataset and extract the features for text representation. Such a preprocessing procedure is called feature extraction. Manifold learning is an efficient approach for feature extraction.

Manifold learning is first proposed by Tenenbaum et al. in the Science Magazine in 2000 as a concept of machine learning. Building on the manifold geometric construction, manifold learning is a nonlinear dimensionality reduction technique to reduce high-dimensional samples to nonlinear structure distribution. This kind of algorithm is assumed to place high-dimensional sample point sets on a low-dimensional manifold, which has a highly complicated nonlinear structure and cannot obtain the manifold features as a whole. Therefore, the idea of localization emerges in manifold learning. Essentially, manifold learning is a process of extracting features from high-dimensional datasets, rejecting noise features of no use to a learning assignment, and retaining those useful ones. Therefore, manifold learning is mainly applied to preprocessing datasets, simplifying data representations and reducing time for a learning assignment. At present, there are two types of manifold learning algorithms, i.e., the global structure-preserving dimensionality reduction algorithm, e.g., Isomap [28] containing the geodesic distance between all sample points and local structure-preserving dimensionality reduction algorithms, as shown in locally linear embedding (LLE) [29], Laplacian eigenmaps (LEP) [30], local tangent space alignment (LTSA) [31], and locality preserving projections (LPP) [32].

Text datasets are typically characterized by a highly complex structure in the feature space, and manifold learning can be applied to obtaining the neighboring local structure and the complicated overall structure of datasets. In this way, datasets in the form of text can be processed effectively.

3. Our Proposed Algorithm

This study explores operating the manifold learning algorithm on email text datasets to extract useful features and train the classifier with the SVM algorithm for email classification. This section introduces the manifold learning algorithm and then elaborates on the steps for the mentioned algorithms.

3.1. Laplacian Eigenmaps. The idea of localization is first proposed in manifold learning. That means the critical step to manifold learning is to divide the neighborhood of datasets and then excavate the geometrical characteristics of each neighborhood. In this study, we use the Laplacian eigenmap (LEP) algorithm [30] to extract the useful features of datasets. As a local structure-preserving algorithm, the LEP is time-efficient and features a flexible internal mechanism under which regular terms and other structural information can be added. For the sake of presentation, we suppose the input sample set is expressed as $\{x_1, x_2, \ldots, x_N\}$, and after dimensionality reduction, we have the output sample set expressed as $\{y_1, y_2, \ldots, y_N\}$. The steps in detail for the LEP algorithm go as follows:

(i) Step I: calculate the k-neighborhood of all input sample points with the k-nearest neighbor algorithm, and the corresponding neighborhood for x_i is expressed as U_i.

(ii) Step II: construct the adjacent map on the input sample set and establish the edge structure only between any two points within the sample neighborhood. Each edge is endowed with the weight $w_i j$ as

$$w_{ij} = \begin{cases} \exp^{-\left(\left\|x_i - x_j\right\|^2/2\sigma^2\right)}, & \text{if } x_j \in U_i, \\ 0, & \text{if } x_j \notin U_i, \end{cases} \tag{1}$$

where $\exp^{-\left(\|x_i - x_j\|^2/2\sigma^2\right)}$ denotes a Gaussian function, and σ is the parameter of the Gaussian function.

(iii) Step III: reconstruct a group of sample point sets in the low-dimensional space, so that between the low-dimensional field points, this weight structure can still be satisfied, so the low-dimensional space expression sets can be obtained by optimizing the following function:

$$\min \sum_{ij} \left\|y_i - y_j\right\|^2 w_{ij}. \tag{2}$$

Let $Y = [y_1, y_2, \ldots, y_N]$ and $[W]_i\ j = w_{ij}$; the above objective function can be written in the form of matrix representation:

$$\min \operatorname{trace}\left(\mathrm{YLY}^T\right) \quad \text{s.t. } \mathrm{YDY}^{\mathrm{T}} = \mathrm{I}, \tag{3}$$

where matrix D is a diagonal matrix, $D_{ii} = \sum_{i=1}^{N} w_i j$, and matrix $L = D - W$ is called the Laplacian matrix.

3.2. Feature Extraction-Based Email Classification Algorithm. Generally speaking, the email datasets we deal with are text datasets. As a result, to fulfill the learning assignment, we need to convert these text datasets into digital ones that algorithms can process. Therefore, the algorithm as a whole has three steps. First, preprocess text datasets; second, use the manifold learning algorithm to extract the features of datasets; finally, operate the SVM algorithm on low-dimensional datasets for classification training. Emails are datasets in the form of text, so we use the doc2vec technique [33] to convert them into datasets in the form of a vector, ensuring machine learning algorithms can process them. Note that doc2vec is a natural language processing tool for representing documents as a vector and is a generalization of the word2vec method [34, 35].

As for a manifold learning algorithm, there are three choices, i.e., Isomap, LLE, and LEP. The Isomap algorithm is inefficient in processing massive ultrahigh-dimensional datasets due to its high time complexity. The LLE algorithm assumes that the local neighborhood of datasets is linear space and acquires the linear correlation representation of the neighborhood. Compared with Isomap and LLE, the LEP algorithm has quite flexible design methods. It works by constructing the adjacent map of datasets and obtaining the weight on the edge of the map. We can provide additional feature information during this weight acquisition process, like the context structure information of the text and deep layer nonlinear structure information distributed in the text datasets. Therefore, to address the complexity of text datasets, we employ the LEP algorithm to extract text features.

The specific steps for this algorithm are shown in Figure 1 as follows:

(i) Step I: build an email dictionary. Using the doc2vec technique, we convert the text information of each email into the form of vector representation. After the doc2vec conversion, the initial vector dimensionality can be quite different. To this end, we use the principal component analysis (PCA) [36] to adjust the vector dimensionality and ensure all vector dimensions in good conformity.

(ii) Step II: reduce, with the LEP algorithm, the dimensionality of the datasets obtained in step I, with specific algorithm steps shown in Subsection 3.1. Moreover, we can obtain the context structure information of the email and apply it to the weight calculation process as regular items. Based on the successive order of sentences and paragraphs in the text, we provide their context weights. Two neighboring sentences boast higher weights than others, and sentences within the same paragraph bear higher weights than those in different paragraphs. Applying this weight information as regular items to the Gaussian function, we can calculate the weight on the adjacent map.

(iii) Step III: use SVM to classify the datasets of low-dimensional feature representation obtained

(iv) Step IV: predict the email classification accuracy. We use the method mentioned in Step I to process test emails and convert them into training datasets of the same dimensionality.

4. Analysis and Experiments

The section introduces several types of common datasets in email classification, based on which we compare the accuracy of our new algorithm with conventional classification algorithms to demonstrate the strengths of this new algorithm.

4.1. Time Complexity Analysis. The SVM classifier features the training and testing complexities to be $O(m^2N^2)$ and $O(m^2N)$, respectively, in which N stands for sample point number and m represents the sample feature dimension, following the traditional SVM classifiers [37]. The time complexity of the LEP algorithm involves two parts. On the one hand, k-neighborhood set division takes $O(N^2(m+k))$ time. On the other hand, graph embedding calculation takes $O(dN^2)$ time, where d stands for low-dimensional representation of dimensionality. Therefore, we can use the LEP algorithm and the SVM algorithm to obtain a new algorithm with time complexity: $O(N^2(m+k+d+d^2))$. If the input sample set dimension m is very high, the new algorithm's time complexity will be approximate $O(N^2m)$. Compared with the SVM algorithm, the time complexity is much lower.

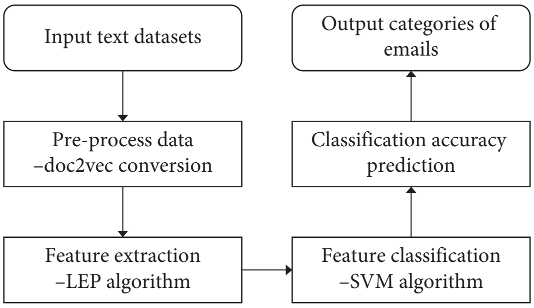
Input text datasets
Pre-process data –doc2vec conversion
Feature extraction –LEP algorithm
Feature classification –SVM algorithm
Classification accuracy prediction
Output categories of emails

Figure 1: Algorithm flow schematic diagram.

4.2. Spam Datasets. We found there are six representative spam datasets, i.e., EnronSpam, PU1, PU2, PU3, PUA, and GenSpam, as described in Table 1. In particular, EnronSpam [38] is currently a common spam dataset. EnronSpam includes 33,702 emails altogether, including 16,764 regular emails and 16,938 junk emails, accounting for around 50% of the total. With largely the same format comprising subject and text, these emails were mainly from 150 users. These datasets are preprocessed, with each email represented by one text and each text numbered chronologically. This experiment is designed to conduct a contrastive analysis of the algorithm accuracy for these datasets. In terms of PU1, PU2, PU3, and PUA, their content distributions are similar. In this study, we only consider PU1, which is a representative dataset out of the four. To summarize, the performance of our proposed method is compared with the state-of-the-art solutions on three datasets, i.e., EnronSpam, PU1, and GenSpam. We divide each dataset into 70% as the training data and 30% as the test data randomly as usual.

4.3. Experimental Results. This experiment design involves two parts. First, we train classifiers with datasets. Then, we use the trained classifiers to conduct spam email sorting prediction for the test sample sets before calculating their corresponding prediction accuracy. In this experiment, we do the contrastive analysis of the performance of our new algorithm from two aspects, i.e., accuracy prediction and the time taken by the algorithm. Note that the misclassification ratios of the proposed algorithms and the benchmarks are zero through fine calibration. As a result, the misclassification ratio is not included in the comparison results. Specifically, we operate two types of SVM classification algorithms as displayed in [18, 19] on the datasets of PU1 and GenSpam, respectively. Their corresponding accuracy results are given in Table 2. The table provides that the classification accuracy of our SVM + LEP algorithm is not considerably different from that of the other algorithms. With the LEP algorithm applied to the email text's structural

Table 1: Several types of spam email corpora.

Corpora	Email	Spam email	Regular email	Spamming rate	Time
EnronSpam [38]	33702	16938	16764	50%	2006
PU1 [39]	1099	484	615	40%	2000
PU2 [39]	721	144	577	20%	2003
PU3 [39]	4139	1821	2318	44%	2003
PUA [39]	1142	570	572	50%	2000
GenSpam [38]	41404	32295	9109	78%	2005

Table 2: Different algorithms' classification accuracy.

Datasets	SVM [19]	SVM [18]	SVM + LEP	SVM + LEP + Struc
EnronSpam	92.1%	93.5%	93.9%	94.7%
PU1	95.8%	96.1%	95.6%	96.9%
GenSpam	93.7%	94.6%	93.9%	95.1%

Table 3: Time consumption for different algorithms.

Datasets	SVM [19]	SVM [18]	SVM + LEP	SVM + LEP + Struc
EnronSpam	3523 s	2582 s	983 s	1312 s
PU1	743 s	642 s	236 s	285 s
GenSpam	2452 s	1834 s	634 s	715 s

Table 4: Different manifold learning algorithms' classification accuracy.

Datasets	Isomap + SVM	LLE + SVM	LPP + SVM	SVM + LEP	SVM + LEP + Struc
EnronSpam	92.4%	93.3%	92.8%	93.9%	94.7%
PU1	94.7%	95.8%	95.1%	95.6%	96.9%
GenSpam	92.9%	93.7%	93.5%	93.9%	95.1%

Table 5: Different manifold learning algorithms' time consumption.

Datasets	Isomap + SVM	LLE + SVM	LPP + SVM	SVM + LEP	SVM + LEP + Struc
EnronSpam	6468 s	894 s	965 s	983 s	1312 s
PU1	1683 s	247 s	229 s	236 s	285 s
GenSpam	5478 s	608 s	615 s	634 s	715 s

information, their corresponding classification accuracy improves remarkably.

Now, we analyze the time consumption of different algorithms. We still contrast the two algorithms with our algorithm, whose results are presented in Table 3. The table provides that the time consumption of our new algorithm is significantly lower than that of the other two algorithms. We can use the manifold learning algorithm to extract the features of datasets and effectively reduce the time it takes to train and test these classifiers. Manifold learning also works to remove data noise; so to some extent, it rejects invalid information from the datasets.

4.4. Discussion of Manifold Learning Algorithms. The experiment above helps compare the LEP algorithm-based email classification method with conventional classification methods. Since 2000, manifold learning has given rise to a series of classic algorithms with distinct advantages. In this section, we contrast several types of manifold learning algorithms with the LEP algorithm to show the latter's advantages in email feature extraction. The selected algorithms are Isomap, LLE, and LPP. In designing the experiment, we have contrastive analysis from two aspects: classification accuracy and time consumption.

We still use three groups of datasets for experimentation on classification accuracy: EnronSpam, PU1, and GenSpam. First, we use different manifold learning algorithms to reduce the dimensionality of the text datasets down to the same low dimension. We then implement the SVM algorithm to fulfill the classification assignment, and the ultimate classification results are presented in Table 4. This table provides that the LEP algorithm with no structural information added sees its dimensionality reduction results similar to those of the other three algorithms. With the structural information added, however, classification accuracy improves significantly. Therefore, in contrast with other manifold learning algorithms, the LEP algorithm has more flexibility in design. It invites additional structural information, while the LPP algorithm learns linear dimension reduction mapping and thus is not as flexible as the LEP algorithm.

In terms of time consumption, we compare different algorithms' time consumption for dimensionality reduction, and the corresponding experiment contrast results are listed in Table 5. These results show that Isomap has high time complexity, rendering it inefficient in processing massive datasets. Like the LLE and LPP algorithms, the LEP algorithm represents a local dimensionality reduction technique, meaning their time consumption varies barely.

5. Conclusion and Future Directions

Spam filtering has been a critical concern across sectors and industries. If we regard it as a scientific problem, addressing this issue could be a classification issue. However, this classification assignment involves how to have data representation of text and data preprocessing and improve the accuracy of sorting algorithms.

For the top priority of text processing, feature extraction plays a vital role in the follow-up learning assignment. In terms of feature extraction, the typical principal component analysis (PCA) method has some limitations and works only to process the datasets showing the linear structure in data distribution. The standard text datasets themselves have highly complex spatial structures and show high-degree nonlinearity in spatial distribution. With that, it is ineffective to adopt the PCA method for dimensionality reduction. The manifold learning algorithm should be introduced to process the complex structure of text datasets, clean out noise information, and merge redundant features. The objective is to obtain minimum decisive features, reduce data size, and improve learning efficiency.

Manifold learning can obtain the spatial geometric construction of datasets as it bases the geometrical characteristics of datasets on algorithm construction. As for email text datasets, each text comprises a group of sentences, and currently, the mainstream algorithms are learning the feature vector representation of sentences, with few capable of obtaining the spatial structure between sentences. Including manifold assumptions into the algorithm for text analysis, we can increase the structural information of text datasets. Such an advantage is vital to natural language processing, or perhaps to a certain degree, provide some breakthroughs for developing natural language processing-related algorithms. This study is a preliminary attempt to address spam filtering concerns, and in the future, text processing in other fields can also be merged with manifold learning. As long as the complex distributed architecture of datasets is involved, we can always try to bring in the manifold hypothesis to address this problem.

This study introduced some machine learning algorithms in spam sorting. On this basis, we proposed a new learning algorithm, which, together with the manifold learning algorithm, works to preprocess datasets and effectively reduce the algorithm's time complexity. In the Experimentation section, we carried out a contrastive analysis of different algorithms' classification accuracy and time consumption.

In the future, we expect that more machine learning algorithms can be applied to ensuring cybersecurity. Take spam filtering as an example. For the corpora of text, we can introduce the embedding idea in deep learning and use the embedding method to have a vector representation of the email text. Then, for this type of vector dataset, we can choose some appropriate machine learning algorithms for classification. Unquestionably, the classification results are affected by the embedding performance to add the semantic and structural information of the text to the embedding process. In this way, we may obtain much better classification results.

Data Availability

Some or all data, models, or code generated or used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Goodman, V. C. Gordon, and D. Heckerman, "Spam and the ongoing battle for the inbox," *Communications of the ACM*, vol. 50, no. 2, pp. 24–33, 2007.

[2] Y. Jin, T. Li, G. Liang et al., "Spam transaction attack detection model based on gru and wgan-div," *Computer Communications*, vol. 161, pp. 172–182, 2020.

[3] D. Fetterly, M. Manasse, and M. Najork, "Spam, damn spam, and statistics: using statistical analysis to locate spam web pages," in *Proceedings of the 7th International Workshop on the Web and Databases: Colocated with ACM SIGMOD/PODS 2004*, Paris, France, June 2004.

[4] C. Godwin and M. Li, "A survey of emerging approaches to spam filtering," *ACM Computing Surveys (CSUR)*, vol. 44, no. 2, pp. 1–27, 2008.

[5] X. Tian, "A constant time complexity spam detection algorithm for boosting throughput on rule-based filtering systems," *IEEE Access*, vol. 8, pp. 82653–82661, 2020.

[6] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10206–10222, 2009.

[7] O. Amayri and N. Bouguila, "A study of spam filtering using support vector machines," *Artificial Intelligence Review*, vol. 34, no. 1, pp. 73–108, 2010.

[8] P. P. K. Chan, C. Yang, D. S. Yeung, and W. W. Y. Ng, "Spam filtering for short messages in adversarial environment," *Neurocomputing*, vol. 155, pp. 167–176, 2015.

[9] H. Shen and Z. Li, "Leveraging social networks for effective spam filtering," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2743–2759, 2013.

[10] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: review and approaches," *Artificial Intelligence Review*, vol. 53, pp. 1–63, 2020.

[11] B. K. Dedeturk and B. Akay, "Spam filtering using a logistic regression model trained by an artificial bee colony algorithm," *Applied Soft Computing*, vol. 91, p. 106229, 2020.

[12] T. A. Almeida, T. P. Silva, I. Santos, and J. M. Gómez Hidalgo, "Text normalization and semantic indexing to enhance instant messaging and SMS spam filtering," *Knowledge-Based Systems*, vol. 108, pp. 25–32, 2016.

[13] J. R. Méndez, T. R. Cotos-Yañez, and D. Ruano-Ordás, "A new semantic-based feature selection method for spam filtering," *Applied Soft Computing*, vol. 76, pp. 89–104, 2019.

[14] X. Tian and X. Chen, "A weighted feature enhanced hidden Markov model for spam SMS filtering," *Neurocomputing*, vol. 444, pp. 48–58, 2021.

[15] S. K. Murthy, "Automatic construction of decision trees from data: a multi-disciplinary survey," *Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 345–389, 1998.

[16] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[17] X. Carreras and L. Marquez, "Boosting trees for anti-spam email filtering," 2001, https://arxiv.org/abs/cs/0109015.

[18] D. Sculley and G. M. Wachman, "Relaxed online SVMs for spam filtering," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 415–422, Amsterdam, the Netherlands, July 2007.

[19] K. D. Renuka and P. Visalakshi, "Latent semantic indexing based SVM model for email spam classification," *Journal of Scientific and Industrial Research*, vol. 73, no. 7, 2014.

[20] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with Naive Bayes-which Naive Bayes?" in *Proceedings of the CEAS Third Conference on Email and Anti-Spam*, vol. 17, pp. 28–69, Mountain View, CA, USA, July 2006.

[21] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, P. George, and C. D. Spyropoulos, "An evaluation of Naive Bayesian anti-spam filtering," 2000, https://arxiv.org/pdf/cs/0006013.

[22] J. Wang, J. Cao, M. Stojmenovic et al., "Pattern-rl: multi-robot cooperative pattern formation via deep reinforcement learning," in *Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 210–215, IEEE, Boca Raton, FL, USA, December 2019.

[23] Y. Dou, G. Ma, P. S. Yu, and S. Xie, "Robust spammer detection by nash reinforcement learning," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 924–933, San Diego, CA, USA, August 2020.

[24] G. Tzortzis and A. Likas, "Deep belief networks for spam filtering," in *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, vol. 2, pp. 306–309, Patras, Greece, October 2007.

[25] G. Mi, Y. Gao, and Y. Tan, "Apply stacked auto-encoder to spam detection," in *Proceedings of the International Conference in Swarm Intelligence*, pp. 3–15, Springer, Beijing, China, June 2015.

[26] A. Barushka and P. Hajek, "Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks," *Applied Intelligence*, vol. 48, no. 10, pp. 3538–3556, 2018.

[27] S. Srinivasan, V. Ravi, M. Alazab, S. Ketha, A. M. Al-Zoubi, and S. Kotti Padannayil, "Spam emails detection based on distributed word embedding with deep learning," in *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, pp. 161–189, Springer, Berlin, Germany, 2021.

[28] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[29] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[30] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in Neural Information Processing Systems (NIPS)*, vol. 14, pp. 585–591, 2001.

[31] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 313–338, 2004.

[32] X. He and P. Niyogi, "Locality preserving projections," *Advances in Neural Information Processing Systems (NIPS)*, vol. 16, no. 16, pp. 153–160, 2004.

[33] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning*, pp. 1188–1196, PMLR, Beijing, China, June 2014.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Y. Bengio and Y. LeCun, Efficient estimation of word representations in vector space," in *Proceedings of the 1st International Conference on Learning Representations, ICLR 2013*, Workshop Track Proceedings, Scottsdale, AZ, USA, May 2013.

[35] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," 2016, https://arxiv.org/abs/1607.05368.

[36] L. I. Smith, *A Tutorial on Principal Components Analysis*, Bibsonomy, Kassel Hessen, Germany, 2002.

[37] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[38] X. Yang, L. Kong, Z. Liu et al., "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[39] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.