Volume 13, Issue 3, Page 176-183, 2024 https://doi.org/10.46810/tdfd.1519463 Research Article

BİNGÖL UNIVERSITY

Türk Doğa ve Fen Dergisi
Turkish Journal of Nature and Science

www.dergipark.gov.tr/tdfd

Classification of Temporary and Real E-mail Addresses with Machine Learning Techniques

Caner BALIM[1*], Nevzat OLGUN[2]

[1] Afyon Kocatepe University, Engineering Faculty, Software Engineering Department, Afyonkarahisar, Türkiye
[2] Afyon Kocatepe University, Engineering Faculty, Software Engineering Department, Afyonkarahisar, Türkiye
Caner BALIM ORCID No: 0000-0002-1010-129X
Nevzat OLGUN ORCID No: 0000-0003-2461-4923

**Corresponding author: cbalim@aku.edu.tr*

(Received: 20.07.2024, Accepted: 12.09.2024, Online Publication: 26.09.2024)

Keywords
E-mail classification, Natural language processing, Artificial neural network, Machine learning

Abstract: Temporary e-mail addresses are e-mail addresses that users can quickly create without signing up. These e-mail addresses are useful for privacy and to avoid spam. However, they also pose several serious cyber threats, including fraud, spam campaigns, and fake account creation In this study, a method utilizing natural language processing and machine learning techniques is proposed to classify real and temporary e-mail addresses. First, temporary and real e-mail addresses are analyzed, and features are developed to identify the differences between them. These features include lexical structures, broad contexts, and structural features of e-mail addresses. Various machine learning algorithms were then applied on the resulting feature set to differentiate e-mail addresses. The results were evaluated with K-fold cross-validation method and an accuracy rate of 96% was obtained. This success rate shows that the developed method can successfully distinguish between real and temporary e-mail addresses.

Geçici ve Gerçek E-posta Adreslerinin Makine Öğrenme Teknikleriyle Sınıflandırılması

Anahtar Kelimeler
E-mail sınıflandırma, Doğal dil işleme, Yapay sinir ağı, Makine öğrenmesi

Öz: Geçici e-posta adresleri, kullanıcıların üye olmadan hızlı bir şekilde oluşturabildikleri e-posta adresleridir. Bu e-posta adresleri gizlilik ve istenmeyen e-postalardan kaçınmak için yararlıdır. Fakat bu e-postalar adreslerinin dolandırıcılığa, spam kampanyalarında kullanılma ve sahte hesap oluşturmaya kadar bir dizi ciddi siber tehdidi de bulunmaktadır. Bu çalışmada, gerçek ve geçici e-posta adreslerini sınıflandırmak için doğal dil işleme ve makine öğrenme tekniklerinden yararlanan bir yöntem önerilmiştir. Öncelikle, geçici ve gerçek e-posta adresleri analiz edilmiş ve arasındaki farkları belirlemeye yönelik öznitelikler geliştirilmiştir. Bu öznitelikler, e-posta adreslerinin leksik yapılarını, geniş bağlamlarını ve yapısal özelliklerini içermektedir. Sonrasında elde edilen öznitelik seti üzerinde, çeşitli makine öğrenme algoritmaları uygulanmış ve e-posta adresleri ayırt edilmeye çalışılmıştır. Elde edilen sonuçlar, K-katlı çapraz doğrulama yöntemiyle değerlendirilmiş ve %96 doğruluk oranı elde edilmiştir. Bu başarı oranı, geliştirilen yöntemin gerçek ve geçici e-posta adreslerini başarılı bir şekilde ayırt edebileceğini göstermektedir.

1. INTRODUCTION

Temporary e-mail addresses are mainly used for short-term, anonymous e-mail communication. Such addresses are usually valid for periods ranging from a few minutes to a few days, after which they automatically disappear. Temporary e-mail services allow users to create e-mail addresses quickly and without registration. These services are often used during online registrations, forums, or various downloads over the Internet. In this way, users do not risk their personal or business e-mail addresses for such temporary activities. They are also widely used to avoid spam e-mails and to protect online privacy.

While temporary e-mails are helpful, they also have the potential for misuse. For example, some users may use such addresses to create multiple accounts, violate terms of service, or engage in illegal activities. This can lead to security and management challenges for online platforms and services. Therefore, detecting and managing temporary e-mail addresses has become an important issue, especially for businesses and service providers.

There are not many arguments that can be used to identify temporary e-mail addresses other than their addresses. Classification based solely on e-mail addresses seems to be a technically simpler process when performed on the basis of certain criteria, but it has its own difficulties. Although the structure of the addresses is generally standardized, it is very difficult to make inferences based on features such as the information contained in the address and the domain name used. In order to develop a good classification algorithm, it is very important to know the structure of e-mail addresses. E-mail addresses can be structurally divided into two main parts:

1. Local part: The part of the e-mail address that comes before the "@" sign. It may contain the user's name, alias, or other identifying information. The local part may be case-sensitive.

2. Domain part: The part of the e-mail address that comes after the "@" sign. This part is usually associated with the name of a website (example.com, etc.) and is not case-sensitive. The domain part includes the Top-Level domain (TLD). TLDs located within the domain are used to provide more specific addressing. Different sub-domains such as ".com", ".edu", and ".gov" can be defined.

The at ("@") and dot (".") Symbols are very important in e-mail addresses. The at sign is used to distinguish between the local part and the domain part. The dot can be used both in the local and domain parts. The dot sign is used in the local part to separate different words or sections, while in the domain it is used to separate TLDs. In addition to the period, signs such as underscore (_) and plus (+) can be used in the local part of e-mail addresses. However, the use of special characters in the domain part is very limited.

There are various studies on e-mail address classification in the literature [1], [2], [3], [4]. It was observed that Enron and Spam Assassin data sets are used in the majority of the studies [5], [6]. Since there is limited information about e-mail addresses, it has been observed that most of the classification studies have been used with the content of the e-mail address. Also, it has been observed that in most studies, researchers focus on ham-spam classification and security analysis (ham-phishing, etc.).

Ham-spam e-mail classification is one of the issues that has attracted the attention of researchers since long ago. The term "ham" means "clean" or "correct" in the context of e-mail classification. Many important works have been proposed in this area. Nowadays, many models show excellent performance with over 90% accuracy. Most of the current work is carried out using deep learning-based approaches. In particular, Long Short-Term Memory (LSTM), which is a type of Recurrent Neural Network (RNN), and Transformer-based BERT (Bidirectional Encoder Representations from Transformers) approaches are widely used [7]. Debnath and Kar obtained 99.14% accuracy by using BERT, LSTM, and natural language processing (NLP) techniques on the Enron e-mail dataset [8]. In their study on the Enron dataset, Krishnamoorthy et al. used an LSTM-based hybrid deep learning approach and obtained an accuracy rate of approximately 98% [9]. AbdulNabi and Yaseen used BERT to demonstrate the effectiveness of word embedding in spam e-mail classification [10]. In addition to the successful results obtained with deep learning techniques, some traditional algorithms also give very successful results [2], [11], [12]. Dedeturk and Akay proposed a method combining logistic regression with artificial bee colony algorithm for spam detection [11]. Saidani et al. proposed a two-step semantic analysis for e-mail spam analysis [2]. In the first step, the categorization of the e-mail is performed; in the second step, domain-specific semantic features are extracted.

Another important issue on e-mail classification that attracts the attention of researchers is the detection of phishing attacks. Phishing attacks are a type of cyber-attack in which hackers aim to mislead users into accessing sensitive information. Attackers mostly aim to steal important information such as usernames, passwords, credit cards. Such attacks are often conducted through e-mails that appear legitimate but are malicious. Rastenis et al. created a taxonomy of e-mail-based phishing attacks [13]. For more detailed information, this study can be reviewed.

An analysis of the studies in the literature shows that NLP and Machine Learning (ML) techniques have been applied to the task of ham-phishing email classification, as well as ham-spam classification [3]. Gholampour and Verma developed an enriched ham-phishing e-mail dataset on different phishing attacks using GPT-2 [14]. Kumar et al. developed a hybrid phishing detection system combining SVM classification and feature extraction [15]. Fang et al. first analyzed the e-mail structure and then used a special recurrent convolutional neural network model to model e-mails according to their features, such as header, body, and character level [16]. More details on phishing mail detection using natural language processing techniques can be found in the review by Salloum et al [17].

In this study, a model that performs the classification of temporary e-mail addresses and real e-mail addresses is proposed. Temporary e-mail addresses are services that have good features, such as providing anonymity to users, but also have the possibility of abuse (violating the terms of service or engaging in illegal activities, etc.). Within the scope of research, no previous study on this subject and no data set has been found in the literature. In the study, NLP and ML techniques, which are frequently studied by researchers, are utilized. The study's main contributions can be summarized as follows:

- A model is proposed to detect temporary e-mail addresses using NLP and ML techniques. A hybrid approach is followed, and different type of feature categories are combined.
- The impact of different feature types (lexical structures, broad contexts and structural features) on e-mail address classification is shown. In addition,

the effect of different feature pairs on classification is also shown.

- A new dataset that can be used in e-mail classification tasks has been produced. When building the dataset, care was taken to ensure that the e-mail addresses collected were from different domains and TLDs. The dataset contains a close ratio of temporary and real e-mail addresses.

The article is structured as follows: Section 2 describes the material and method used for problem solving. Section 3 presents experimental results and comments. Finally, Section 4 concludes and makes suggestions for future research.

2. MATERIAL AND METHODOLOGY

In this section, firstly, the material on which the presented methods are applied is explained, and then the general steps of the method are explained in detail.

2.1. Material

There are over 100 different temporary email providers on the Internet. These providers usually provide valid email addresses for 10 minutes with different domain addresses. As a result of the research carried out, a new data set was created since there was no publicly available data set on this subject before. For this purpose, e-mail addresses were collected from different temporary e-mail address providers. Then, verified publicly available email addresses were collected. While collecting real email addresses, sample addresses were also taken from different institutions and organizations, such as universities, governments, and e-commerce sites, for a fair approach. In order to create a balanced data set while collecting email addresses, care was taken to have a close number of samples in each class. As a result, statistics about the email addresses collected are shown in Table 1.

Table 1. Dataset Statistic

Real	Temporary	Total
719	765	1484

2.2. Methodology

The methodology aims to predict whether the e-mail is temporary or real based on combining different features extracted from e-mail addresses. The proposed approach consists of two main parts: Feature extraction and classification.

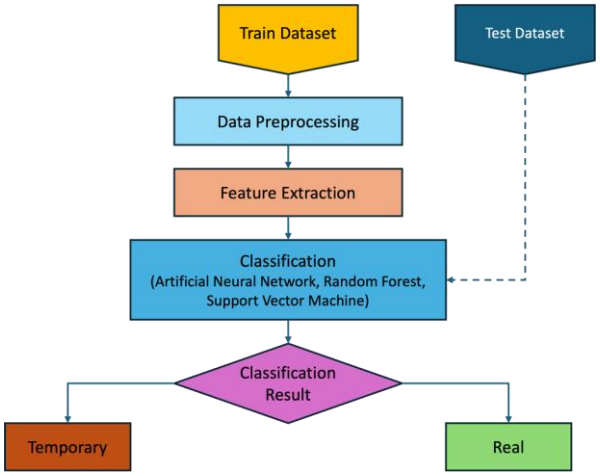

Figure 1. Schematic of the proposed system

2.2.1. Feature extraction

In the process of classifying e-mail addresses, the strategic use of NLP techniques for feature extraction is critical. While deep learning methodologies have delivered impressive results in various text analysis and processing areas, direct deep learning-based feature extraction on specialized and structured data types such as e-mail addresses presents significant challenges due to the data's limited context and its specific structural properties. Therefore, to extract features in the proposed model, weighting methods such as Term Frequency-Inverse Document Frequency (TF-IDF), which have proven successful in text classification and analysis, and spelling control techniques such as the Levenshtein distance are used to extract meaningful and actionable features from e-mail addresses.

In this study, traditional NLP techniques have been utilized to classify between temporary and real e-mail addresses. A hybrid approach that combines different types of features has been employed for better classification performance.

2.2.1.1. Term frequency-inverse document frequency (TF-IDF)

TF-IDF is a statistical method used in text mining and information retrieval. It is used to determine the importance of a term in a document relative to a corpus. TF-IDF is frequently used for NLP tasks such as e-mail classification [4], [18], [19]. It comprises two components: Term Frequency (TF) and Inverse Document Frequency (IDF).

Term Frequency (TF) quantifies the frequency of a term in a document relative to the total number of terms in that document. It is calculated using the formula:

$$\mathrm{TF}(t,d) = \frac{\text{Number of occurrences of term t in the document}}{\text{Total number of terms in the document}} \quad (1)$$

Inverse Document Frequency (IDF) measures the significance of a term across the corpus. It is calculated as:

$$\text{IDF}(t, D) = \log\left(\frac{\text{Total number of documents in the corpus}}{\text{Number of documents containing term}}\right) \quad (2)$$

The TF-IDF score is obtained by multiplying the TF and IDF scores for each term:

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (3)$$

In this study, TF-IDF is used to extract features from e-mail addresses. Before applying TF-IDF, e-mail addresses are preprocessed by removing punctuation marks (@, +, etc.) to standardize the representation and remove non-informative characters. In the next step, a character-level approach is adopted using 1-2 n-grams. This involves breaking the e-mail addresses into substrings of 1 to 2 characters and computing TF-IDF scores for these character n-grams. In this way, it is aimed to capture patterns and features in the structure of e-mail addresses that may not be noticeable at the word level.

2.2.1.2. Top-level domain features

In an e-mail address, the term "Top-Level Domain" (TLD) refers to the last part of the domain name, typically following the final dot ("."). This part of the domain name provides information about the type or purpose of the domain. Common examples of TLDs include ".com", ".org", ".net", ".gov", and ".edu". For example, ".gov" indicates a government organization, ".edu" represents an educational institution, and ".org" often denotes a non-profit organization.

In this study, the most common TLDs in the dataset were compiled and the 15 most used TLDs are added to the model as attributes. This process aimed to capture the potential influence of TLDs on the classification task. The inclusion of TLD-related features enhances the model's ability to distinguish between different types of e-mail addresses based on their domain structure, thereby improving the overall performance of the classification system. Table 2 shows the most common TLDs in the e-mail addresses in the data set.

Table 2. The 15 most mentioned TLDs in the data set.

No	TLDs
1	.com
2	.edu
3	.org
4	.gov
5	.bel
6	.net
7	.kep
8	.co
9	.app
10	.email
11	.digital
12	.info
13	.site
14	.xyz
15	.store

2.2.1.3. Spelling features

Detecting spelling errors in the local and domain parts of e-mail addresses can play an important role in determining temporary e-mail addresses. Systems that provide temporary e-mail addresses often consist of domains with irregular structures or generate usernames that do not follow spelling rules. Spell checking evaluates the meaning and integrity of local or domain domains, and the domain names of real and reliable e-mail addresses usually have a meaningful and consistent structure.

In this study, the English and Turkish dictionaries of Open Office, an open-source application, were used for spell checking [20]. The words in the dictionaries were compared with the local and domain parts of the e-mail addresses and labelled according to their presence in the dictionary. Non-matching parts are evaluated with Levenshtein distance.

The Levenshtein distance, used to measure the difference between two sequences, is a metric announced by V. I. Levenshtein in 1965 [21]. It has a wide range of applications, including plagiarism detection, spell checking, and bioinformatics [22]. The Levenshtein distance algorithm attempts to find the minimum number of edits required to compare two sequences character by character and convert one sequence into the other. It uses a dynamic programming technique to calculate the distance between sequences of different lengths.

If there are two sequences x and y and the lengths of these sequences are m and n respectively, the Levenshtein distance (L(x,y)) between the two sequences is calculated as follows:

- If both m and n are 0, then L(x,y) = 0.
- If m is 0 and n is not, L(x,y) is equal to n.
- Similarly, if n is 0 and m is not, L(x,y) is equal to m.
- If both m and n are not 0. Their last characters x_(m-1) and y_(n-1). The Levenshtein distance can be calculated by considering the following scenarios:

1. If $x_{m-1} = x_{n-1}$, no operation is needed. In this case, $L(x, y) = L(x_{m-2}, y_{n-2})$.
2. If $x_{m-1} \neq y_{n-1}$, calculate the minimum cost among three operations (insertion, deletion, substitution):

a. Insertion:

$$L(\text{x}, y_{n-1}) = L(\text{x}, y_{n-2}) + 1 \quad (4)$$

b. Deletion:

$$L(x_{m-1}, \text{y}) = L(x_{m-2}, \text{y}) + 1 \quad (5)$$

c. Substitution:

$$L(x_{m-1}, y_{n-1}) = L(x_{m-2}, y_{n-2}) + 1 \quad (6)$$

This operation is performed as the product of the dimensions of the two sequences (mxn).

In this study, for each e-mail address in the dataset, both the local and domain parts of the e-mail address are checked for spelling separately and the Levenshtein score is calculated according to the close values found in the dictionaries and a feature vector is created.

2.2.1.4. Handcrafted features

In the context of machine learning and data analytics, the term "handcrafted features" refers specifically to features designed and selected by human experts from the raw data in the dataset. In this study, five handcrafted features are used that are considered to be discriminative in classifying temporary and real e-mail addresses:

- **Check for punctuation in the local part:** This feature checks whether the local part of the e-mail address contains punctuation (".", "-", "_", etc.). When the local parts of both types are compared, it is seen that very few punctuation marks are used in the local fields of temporary e-mail addresses.

- **Number of letters in the local part:** This feature counts the number of alphabetic characters in the local part of the e-mail address. An examination of the local parts of temporary and real e-mail addresses shows that the number of letters in the local parts of temporary e-mail addresses is in most cases less than the number of letters in the local parts of real e-mail addresses.

- **Number of digits in the local part:** This feature counts the number of digits in the local part of the e-mail address. Analyses of the local parts of temporary and real e-mail addresses show that the number of digits in the local parts of temporary e-mail addresses is in most cases higher than the number of digits in the local parts of real e-mail addresses.

- **Number of punctuation marks in the local and domain parts:** This feature counts the number of punctuation marks in the local and domain parts of the email address. Analysis has shown that the number of punctuation marks used in temporary e-mail addresses is lower than the number of punctuation marks found in real e-mail addresses.

- **Country-code Top Level Domain(ccTLD) existence check:** ccTLDs are two-letter TLDs that represent a specific country or region. This feature examines whether the domain part of the e-mail address has a country or region representation. In the scans, it has been observed that the use of ccTLD in temporary e-mail addresses is quite low.

In this study, values are generated for each e-mail address in the training and test sets using the five discriminative features mentioned above and feature vectors are created with the generated values.

2.2.2. Classifiers

The classifier uses extracted features to classify e-mails as transient or real. Depending on the selected feature pairs, the classifiers detect temporary e-mails from the input data after the training phase.

2.2.2.1. Artificial neural networks (ANNs)

Artificial Neural Networks (ANNs) are an advanced machine learning method that excels in processing nonlinear data inspired by the human nervous system. ANNs operate through two main phases: Forward propagation and backward propagation. Forward propagation refers to the process by which data moves from input to output layers through transformations. The process by which the weights and biases of the network are adjusted through gradient descent to minimize errors between predicted and actual outputs is called backward propagation. This cycle repeats until the network's performance meets the desired criteria. ANNs' ability to learn from data iteratively and model complex relationships makes them highly versatile and effective for various applications, from pattern recognition to predictive modeling.

In the proposed method, a multilayer perceptron (MLP) based classifier model is used as the ANN classifier [23]. Three hidden layers are identified in the MLP model; There are 500 neurons in the first hidden layer, 100 neurons in the second hidden layer and 50 neurons in the third hidden layer. Rectified Linear Unit (ReLU) is used as the activation function. Adaptive Moment estimation (Adam) is chosen as the optimization algorithm.

2.2.2.2. Support vector machines (SVMs)

Support Vector Machines (SVM) is an ML method for classification and regression tasks [24]. SVMs are designed to classify data into distinct categories by constructing a hyperplane in a high-dimensional space. This method hinges on maximizing the margin between the hyperplane and the nearest data points from each category, known as support vectors. SVM aims to find the optimal hyperplane that separates the classes with the greatest possible margin. The effectiveness of SVM lies in its ability to transform the original feature space into a higher dimensional space where a linear separation is feasible, thanks to the kernel trick, thereby accommodating complex and nonlinear relationships between data points.

2.2.2.3. Random forest (RF)

Random Forest (RF) is a powerful ensemble learning technique for both classification and regression tasks where many decision trees are trained together [25]. Based on the concept of bootstrap aggregation (bagging), RF improves the decision tree algorithm by creating a 'forest' of trees where each tree is trained on a random subset of data and features, thus reducing variance and preventing overfitting. By combining predictions from multiple trees, RF can achieve higher accuracy and

stability than a single decision tree. RF is frequently used in classification problems such as spam detection.

2.2.3. Evaluation metrics

The evaluation metrics such as accuracy, precision, and recall values were measured by the proposed method. The formula for accuracy, precision and recall were shown in equations (7) – (9).

In Equation 7 and the following, TP defines the number of instances of correctly classified temporary e-mails, while FP defines the number of e-mails that may be incorrectly classified as temporary when an e-mail is real. TN defines the number of instances of e-mails correctly classified as real e-mails. FN refers to the case where an e-mail that is actually temporary is mistakenly classified as real.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

Precision refers to the ratio of temporary e-mail addresses predicted by the model to actual temporary e-mail addresses. The precision formula is given in Equation 8. A high precision value indicates that the model keeps the number of false positives to a minimum and produces mostly correct results.

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

Recall refers to the rate at which actual temporary e-mail addresses are correctly predicted. The Recall formula is shown in Equation 9. A high recall value means that the model does not miss actual temporary e-mail addresses and classifies most of them correctly.

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

3. EXPERIMENT RESULT

This section presents the experimental studies and discussion. The e-mail addresses in the dataset were first pre-processed with the techniques described in the previous section, and then the features were extracted and combined. The performance measurements of the proposed model are obtained by using the relations between Equations 7-9. The training and test rates were set as 70% and 30%, respectively. The training and test data of the temporary and real e-mail addresses in the dataset were separated according to the K-Fold technique. The results are shown in Table 3.

Table 3. Accuracy values of the proposed model according to K Fold options.

Classifier	1	2	3	4	5	Average
ANN	**0.9663**	0.9494	**0.9730**	0.9494	**0.9662**	**0.9609**
SVM	0.9360	0.9495	0.9428	**0.9529**	0.9392	0.9441
RF	0.9495	**0.9596**	0.9394	**0.9529**	0.9493	0.9501

As shown in Table 3, five different results were obtained with the KFold-5 technique and these results were averaged. When examined according to average values, it is seen that a test accuracy rate of 0.9609 was achieved with the ANN technique. It is observed that the most successful technique after the ANN technique is the RF technique with an accuracy value of 0.9501.

A comparison of the performance of the ANN model using various feature sets on the task of ephemeral e-mail detection is shown as a heat map in Figure 2.

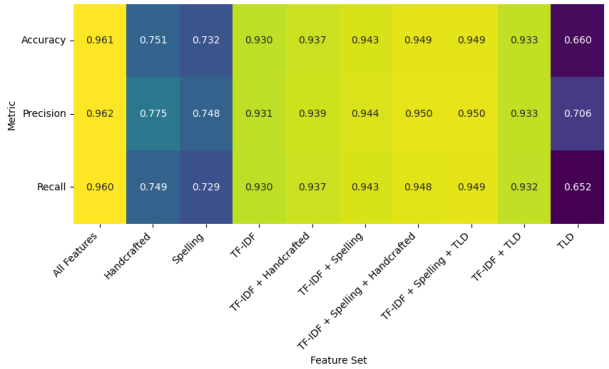

Figure 2. Performance comparison of ANN classification based on various feature sets.

When Figure 2 is examined, it shows that the feature sets used are effective in distinguishing between temporary and real e-mail addresses. When the features are evaluated individually, it is observed that the TF-IDF feature achieves more successful results compared to others. In individual evaluation, the handcrafted features are seen to be the most successful single feature after TF-IDF. When the features are evaluated in pairs with TF-IDF, it is seen that TF-IDF and spelling features (TF-IDF + Spelling) complement each other and produce more successful results than the others. However, in the three-group valuation, the results are not observed to be much different from those obtained with TF-IDF and Spelling features (TF-IDF + Spelling).

It is seen that the TLDs produces the lowest results in the experiments. The reason for this may be the presence of TLDs such as ".com" in almost all e-mail addresses. In addition, it was observed that the TLD ".edu", which was thought to be distinctive in the analyses, is also found in some temporary e-mail addresses. This explains the reason for the lower success of TLD features.

According to Figure 2, the "All Features" set shows the highest performance because it contains the most comprehensive information. However, the use of specific feature sets, such as TF-IDF and Spelling, may be sufficient to achieve nearly similar or better results in certain cases. This shows that the success of the classification model depends on the quality and relevance of the features used.

The accuracy of the proposed method with various feature sets and classifiers is shown in Figure 3. While the TF-IDF feature set has the highest accuracy value when used with the ANN model, the RF model shows higher

accuracy values in models where TF-IDF is not included. It was observed that the TLD feature was the most ineffective feature when considered alone, but when combined with other features, it improved the results. Although the Handcrafted feature set has low accuracy values when used with ANN, it is observed that its performance improves when used with SVM and RF models.

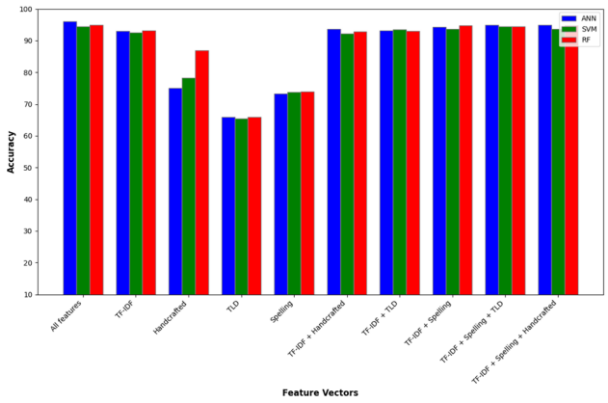

Figure 3. Accuracy of temporary e-mail detection with various feature vectors and classifiers.

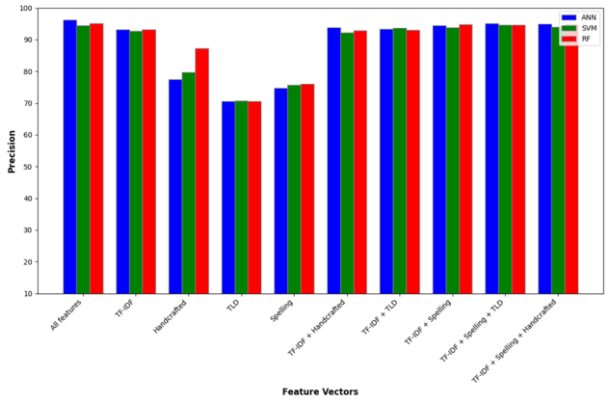

Figure 4. Precision of temporary e-mail detection with various feature vectors and classifiers

In Figure 4 and Figure 5, the precision and recall values of the proposed method are compared for various classifiers and feature sets, respectively. When the TF-IDF feature set is used, the ANN model achieved the highest precision and recall values. However, when TF-IDF and other feature sets (spelling, handcrafted, TLD) are used, different machine learning models also gave good results. In particular, the "All features" set provided the highest precision and recall values for the ANN model. When all features set is used, the precision value of the ANN model is approximately 0.962, while the precision value of the SVM model is approximately 0.945 and the precision value of the RF model is approximately 0.951. Similarly, when the all-feature set is used, the recall value of the ANN model is approximately 0.960, while the recall value of the SVM model is approximately 0.944 and the recall value of the RF model is approximately 0.950.

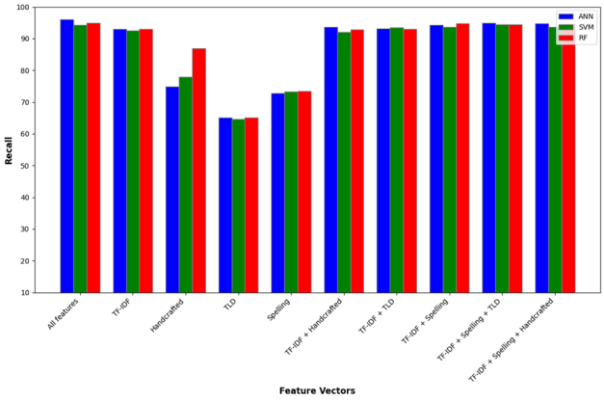

Figure 5. Recall of temporary e-mail detection with various feature vectors and classifiers

4. CONCLUSION

Temporary e-mail addresses allow users to protect their privacy and avoid potentially harmful situations such as receiving spam e-mail. However, when misused, they bring a number of serious cyber threats ranging from cyber security breaches to fraud, spam e-mails, and fake account creation. In this study, a method using various feature sets and machine learning techniques is proposed to classify temporary and real e-mail addresses. As a result of the experiments, an accuracy value of 0.9606 was obtained for the classification of temporary and real e-mail addresses, and the performance of different feature sets and machine learning models were compared in detail.

As a result of the experiments, it is observed that especially the TF-IDF feature set and its combinations with various feature sets achieve high accuracy rates. In particular, the combination of TF-IDF and spelling features is found to produce quite adequate results for this task.

When a comparison is made on a model basis, it is observed that the ANN method mostly performs the best. The ability of the ANN model to effectively handle the complexities in the dataset and the interactions between the features is considered to be an important factor in the classification success. The other methods used in the experiments, RF and SVM, also obtained competitive results, but they were not as successful as ANN. The robust structure and high generalization capacity of ANN is thought to make it prominent for this type of classification problems.

The results obtained from the experiments show that the selection of the right feature sets and learning models can significantly affect the classification performance. It is thought that the findings obtained will contribute to research in areas such as the development of e-mail classification systems and the improvement of spam filtering techniques. Future studies are planned to investigate the effectiveness of different feature selection techniques and machine learning models in this task.

REFERENCES

[1] Diale, M., Celik, T., & Van Der Walt, C. (2019). Unsupervised feature learning for spam email filtering. Computers & Electrical Engineering, 74, 89–104. https://doi.org/10.1016/j.compeleceng.2019.01.004

[2] Saidani, N., Adi, K., & Allili, M. S. (2020). A semantic-based classification approach for an enhanced spam detection. Computers & Security, 94, 101716. https://doi.org/10.1016/j.cose.2020.101716

[3] Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2021). Phishing email detection using natural language processing techniques: A literature survey. Procedia Computer Science, 189, 19–28. https://doi.org/10.1016/j.procs.2021.05.077

[4] Sanghani, G., & Kotecha, K. (2019). Incremental personalized e-mail spam filter using novel TFDCR feature selection with dynamic feature update. Expert Systems with Applications, 115, 287–299. https://doi.org/10.1016/j.eswa.2018.07.049

[5] Kaggle. (2024). The Enron email dataset. Kaggle. https://www.kaggle.com/datasets/wcukierski/enron-email-dataset

[6] Kaggle. (2024). The Spam Assassin email dataset. Kaggle. https://www.kaggle.com/datasets/ganiyuolalekan/spam-assassin-email-classification-dataset

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5999–6009). https://arxiv.org/abs/1706.03762v5

[8] Debnath, K., & Kar, N. (2022). Email spam detection using deep learning approach. In 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON) (pp. 37–41). https://doi.org/10.1109/COM-IT-CON54601.2022.9850588

[9] Krishnamoorthy, P., Sathiyanarayanan, M., & Proença, H. P. (2024). A novel and secured email classification and emotion detection using hybrid deep neural network. International Journal of Cognitive Computing in Engineering, 5, 44–57. https://doi.org/10.1016/j.ijcce.2024.01.002

[10] AbdulNabi, I., & Yaseen, Q. (2021). Spam email detection using deep learning techniques. Procedia Computer Science, 184, 853–858. https://doi.org/10.1016/j.procs.2021.03.107

[11] Dedeturk, B. K., & Akay, B. (2020). Spam filtering using a logistic regression model trained by an artificial bee colony algorithm. Applied Soft Computing, 91, 106229. https://doi.org/10.1016/j.asoc.2020.106229

[12] Gibson, S., Issac, B., Zhang, L., & Jacob, S. M. (2020). Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. IEEE Access, 8, 187914–187932. https://doi.org/10.1109/ACCESS.2020.3030751

[13] Rastenis, J., Ramanauskaitė, S., Janulevičius, J., Čenys, A., Slotkienė, A., & Pakrijauskas, K. (2020). E-mail-based phishing attack taxonomy. Applied Sciences, 10(7), 72363. https://doi.org/10.3390/app10072363

[14] Mehdi Gholampour, P., & Verma, R. M. (2023). Adversarial robustness of phishing email detection models. In Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics (IWSPA '23) (pp. 67–76). https://doi.org/10.1145/3579987.3586567

[15] Kumar, A., Chatterjee, J. M., & Díaz, V. G. (2020). A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing. International Journal of Electrical and Computer Engineering (IJECE), 10(1), 486–493. https://doi.org/10.11591/ijece.v10i1.pp486-493

[16] Fang, Y., Zhang, C., Huang, C., Liu, L., & Yang, Y. (2019). Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism. IEEE Access, 7, 56329–56340. https://doi.org/10.1109/ACCESS.2019.2913705

[17] Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2022). A systematic literature review on phishing email detection using natural language processing techniques. IEEE Access, 10, 65703–65727. https://doi.org/10.1109/ACCESS.2022.3183083

[18] Barushka, A., & Hajek, P. (2018). Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. Applied Intelligence, 48(10), 3538–3556. https://doi.org/10.1007/s10489-018-1161-y

[19] Srinivasarao, U., & Sharaff, A. (2023). Spam email classification and sentiment analysis based on semantic similarity methods. International Journal of Computer Science and Engineering, 26(1), 65–77. https://doi.org/10.1504/ijcse.2023.129147

[20] Apache OpenOffice. (2024). Apache OpenOffice extensions. OpenOffice. https://extensions.openoffice.org/

[21] Levenshtein, V. (1965). Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady.

[22] Berger, B., Waterman, M. S., & Yu, Y. W. (2021). Levenshtein distance, sequence comparison and biological database search. IEEE Transactions on Information Theory, 67(6), 3287–3294. https://doi.org/10.1109/TIT.2020.2996543

[23] Russell, S. J., & Norvig, P. (2010). Artificial intelligence: A modern approach (3rd ed.). Prentice Hall.

[24] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1007/BF00994018

[25] Liaw, A., Wiener, M., & others. (2002). Classification and regression by randomForest. R News, 2(3), 18–22.