applied sciences

MDPI

Communication

Email Campaign Evaluation Based on User and Mail Server Response

Marcin Szpyrka [1,*], Piotr Suszalski [2], Sebastian Obara [2] and Grzegorz J. Nalepa [3]

[1] Department of Applied Computer Science, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland
[2] Freshmail Sp. z o.o., 31-406 Krakow, Poland
[3] Jagiellonian Human-Centered AI Lab, Mark Kac Center for Complex Systems Research, Institute of Applied Computer Science, Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, ul. prof. Stanisława Łojasiewicza 11, 30-348 Krakow, Poland
* Correspondence: mszpyrka@agh.edu.pl

Abstract: The goal of an email service provider company is to send out a large number of emails to help its clients realise successful email marketing activities. Thousands of emails sent every minute need to be analysed in real time to reduce spam or phishing. The paper describes a method that uses real-time tracking of key campaign metrics such as the opens count, clicks count, hard bounces count, etc., to identify campaigns that should be stopped because they can be classified as spam or phishing. The key point of this solution is that we do not analyse email content. Nevertheless, the proposed neural networks are highly effective—the F1-score is above 0.95 for any used sample. Furthermore, the approach allows us to use the same model regardless of the language of an email. The method was developed and verified in collaboration with Freshmail, a leading provider of email marketing services in Poland. Validation of the method on real data provided by the company confirmed its high effectiveness.

Keywords: spam and phishing detection; artificial neural networks; email campaign metrics; transaction emails

Citation: Szpyrka, M.; Suszalski, P.; Obara, S.; Nalepa, G.J. Email Campaign Evaluation Based on User and Mail Server Response. *Appl. Sci.* **2023**, *13*, 1630. https://doi.org/10.3390/app13031630

Academic Editor: Dimitris Mourtzis

Received: 6 January 2023
Revised: 20 January 2023
Accepted: 24 January 2023
Published: 27 January 2023

1. Introduction

An email service provider (ESP) is a technology company that makes it easier for people to build a database of subscribers and send email campaigns to a list of subscribers. An ESP is fundamental to successful email marketing activities, as it both sends emails and tracks key campaign metrics. The effectiveness of the emails sent also depends on the credibility of the ESP company. The use of an ESP system must be carried out in accordance with the provisions of applicable law and the generally accepted principles of commercial Internet activities. In particular, the ESP company must ensure that emails sent are not considered spam or phishing [1–4]. Of course, part of the security is the regulations that customers must follow, such as the Anti-Spam Policy. However, such solutions do not guarantee that a client will not try to send out a campaign that is bad , i.e., it is spam or phishing. This means that it is necessary to use AI tools that will automatically analyse emails in real time and detect situations that are undesirable from the point of view of the company's credibility. In addition, such activities can significantly reduce the amount of spam or phishing that reaches recipients' inboxes. On the other hand, such methods must not block emails sent without sufficient reasons [5,6]. After all, an ESP company's goal is to deliver emails to recipients efficiently.

Most spam detection methods involve analysing the content of the message being sent [7–9]. These methods usually analyse words, the occurrence, and distributions of words and phrases in the content of emails. If the content of an email is graphic, additional

optical recognition of the text embedded in the graphics is required [10,11]. Moreover, links placed inside an email are analysed to detect those that may be dangerous to the user if clicked [12–14]. We can apply a shallow links analysis [15,16], which is largely concerned with the statistical characteristics of links, such as the number of letters, digits, words, underscores, etc., as well as a deep analysis, which allows us to see where the link leads, even if there are redirects along the way. An additional problem is the language version of the email being sent. If the client sends emails in different languages, the problem of analysing the content for spam recognition is even more complex [17,18]. All of these elements make the analysis of thousands of emails sent every minute extremely complex and require significant hardware resources on the ESP's side. The goal of the work presented in this article was to attempt to develop a method that does not directly use the content of an email. The features used in the machine learning process relate to the technical aspects of the campaign that is being sent such as the opens count, clicks count, hard bounces count, etc. It turns out that an ESP company can very successfully using real-time tracking of key campaign metrics to identify campaigns that should be stopped. Since this method does not analyse the content of the email, it works regardless of the language of the emails sent.

The presented solution was developed as part of the SendGuard project led by the FreshMail company. The project is founded by The National Centre for Research and Development, Poland. All computational experiments were performed on real data collected by FreshMail, and the developed solutions will be deployed in the company.

The main contributions of the paper can be summarised as follows:

- Based on the analysis of data collected by Freshmail, we proposed sets of features that are sufficient to analyse the technical aspects of sending campaign and transactional emails;
- We developed methods for labelling large data sets automatically;
- We defined artificial neural networks models and proved that they are highly effective (the F1-score was above 0.95 for any used sample) using real data collected by the company.

The paper is organised as follows. In Section 2, we describe the overall schema of the data processing system in the company. Section 3 describes the data sets with which we work. Artificial Neural Networks (ANN) models and the results of their application are described in Section 4. Some conclusions are provided in Section 5.

2. Data Processing and Analysis System

FreshMail is an Email Service Provider (ESP). The company implements all necessary software to send a huge amount of email every day. FreshMail sends emails in two major categories: marketing emails as well as transaction emails. Custom software used to send emails is built with PHP and Python. As a Message Transfer Agent (MTA), Freshmail uses a cluster of MailerQ instances. MailerQ gathers response information from email receiving servers (such as Gmail, Yahoo, Wp, Onet, etc.). FreshMail provides a custom application that tracks events generated by end users (subscribers) such as email open, link click, unsubscribe, abuse reports, etc. All data are transferred through message brokers built with Redis, RabbitMQ, and Kafka and then gathered in several FreshMail databases and big data storage facilities. Data are stored in Percona Server (an OLTP database) and ElasticSearch clusters (for observability and data analysis). All data are also gathered in a FreshMail Big Data Cluster built with the Hadoop technology. Using the Apache Spark and Kafka, the data are processed and analyzed in real-time and sent to the ClickHouse database (column base OLAP SQL database). Most of the experiments conducted in the SendGuard project use data from the ClickHouse database. The details of the datasets are described in the next section. The FreshMail infrastructure is depicted in Figure 1.

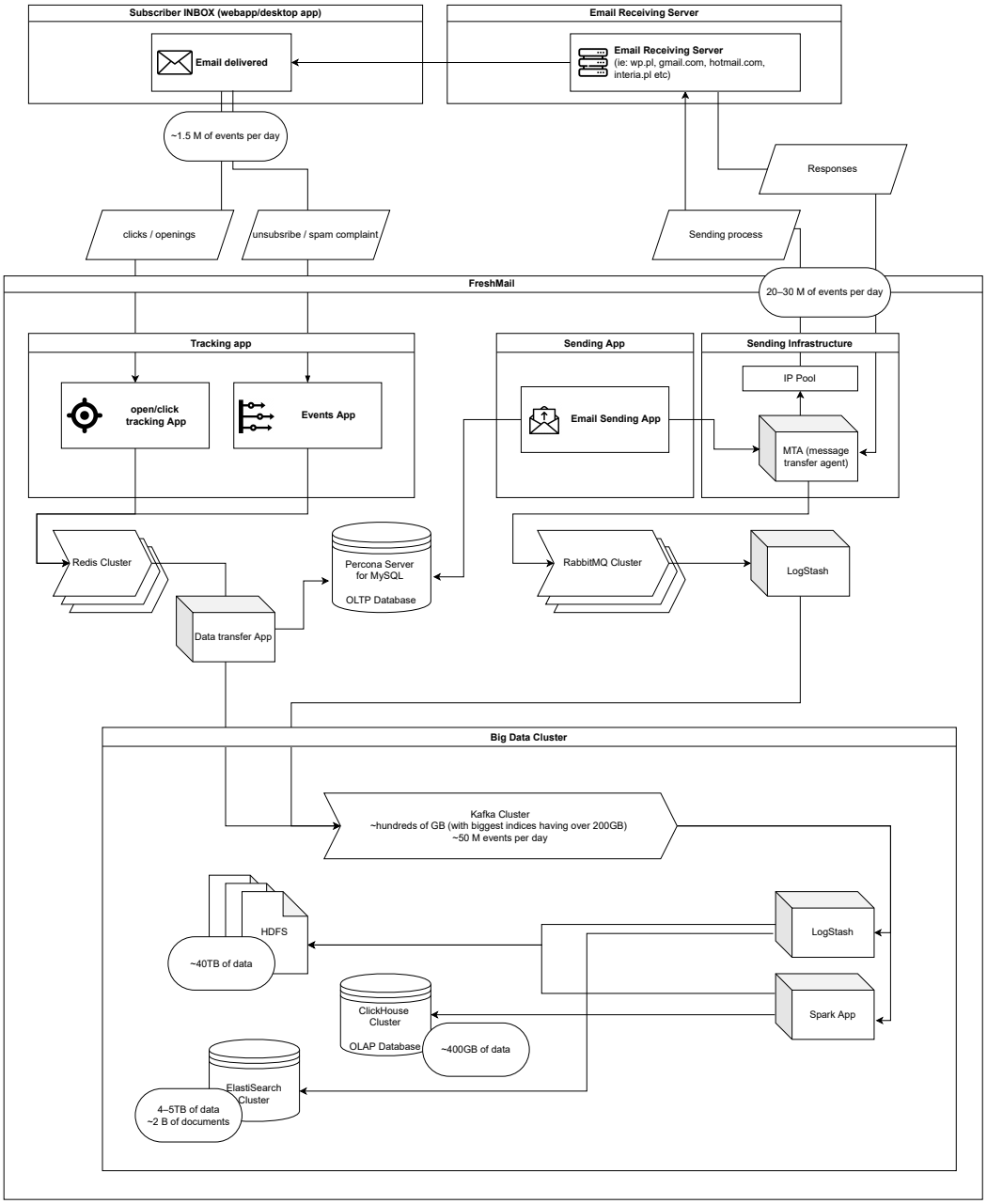

Figure 1. Freshmail data processing infrastructure.

3. Data Sets

As it was mentioned, the emails sent by FreshMail can generally be divided into two groups, campaigns and transactions. The former group concerns sending email campaigns; i.e., the same email, except for minor details such as the recipient's name, is sent to a group of recipients (list of subscribers). Depending on the size of the subscriber list, sending a campaign can take between a few seconds and several dozen minutes. If during the sending process, which lasts at least several minutes, undesirable phenomena are observed, it is possible to terminate the process. The latter group concerns sending single emails related to account activity or a commercial transaction, e.g., password resets, delivery information, and receipts. In this case, we cannot stop sending a single email, but we can block the client (sender) if undesirable phenomena are observed.

Regardless of which type of email we are dealing with, the system collects some technical information (statistics) about the emails sent, which are shown in Table 1. Each of these statistics can refer to a single transaction email, an entire campaign, or a part of it,

such as a single minute. In addition, in the case of transaction emails, we can aggregate emails sent by a single customer over a given period of time.

Table 1. Technical attributes (metrics) collected by ESP servers.

Attribute (Metric)	Description
opens count	number of opens
unique opens count	number of unique opens
clicks count	number of clicks (at least one link)
unique clicks count	number of unique clicks
soft bounced count	number of soft bounced
hard bounced count	number of hard bounced
resigned count	number of resigns
complaint count	number of complaints

3.1. Campaigns

In the case of email campaigns, the statistics shown in Table 1 are available for each minute of sending. From the point of view of machine learning, we treat each minute as a separate object. For the presented solution, each object is described by 24 conditional attributes:

- Eight statistics for the current minute (as shown in Table 1);
- Eight statistics for the previous minute (for the first minute copy of the same values was used);
- Eight statistics for historical campaigns of the same client, e.g., the average number of soft bounced per minute for previous campaigns.

For the first two groups, the values were divided by the number of subscribers for the current campaign. For the last group, the values were divided by the total number of subscribers for all previous campaigns.

Due to the huge volume of the data (millions of records), it was necessary to develop an automated method of labelling. The starting point for this procedure was the attribute that describes the total number of bad events for a given campaign defined as the sum of the soft bounced count, hard bounced count, resigned count, and complaint count (divided by the number of subscribers). Data were divided into two categories, good and bad. As a result of the experiments, five data-labelling methods were proposed:

- label1—The campaign should be stopped (label bad) if the number of bad events exceeds 5%.
- label2—The campaign should be stopped if the hard bounced count exceeds 10% or the resign count exceeds 1.7% and the unique click count does not exceed 1.5%.
- label3—The campaign should be stopped if the standardised value (z-score) of bad events is greater than 2.
- label4—The campaign should be stopped if the unique open count does not exceed 2%.
- label5—A method based on examples of bad clients identified by FreshMail.

Let us focus on the last method (label5). The company provided identifiers of clients who have sent bad campaigns in the past. A preliminary analysis of the campaigns sent by the clients showed that not all of these campaigns could be considered bad. To identify bad campaigns, a clustering of this subset was performed. Using the elbow method, it was determined that a division into eight clusters is optimal. The centres of these clusters are shown in Table 2.

Based on the analysis of the centres, clusters 0, 2, 3, and 6 are considered to be clusters containing correct campaigns (due to the high values of opens and clicks), while clusters 1, 4, 5, and 7 are considered to be clusters containing campaigns that should be stopped (due to the low values of opens and clicks or the high values of hard bounces). All campaigns were labelled according to which of the designated centres they were closest to.

Based on an analysis of the labelling results of methods 1 through 5, it was indicated that it would be optimal to use the superior method (label6), which takes advantage of those previously used. Method 6 is based on three rules:

1. If label3 = 1, then label6 = 1;
2. If label3 = 0, and label1 = label2 = label4 = label5 = 1, then label6 = 1;
3. Otherwise, label6 = 0.

Table 2. Cluster centres.

No	Opens	Unique Opens	Clicks	Unique Clicks	Soft Bounced	Hard Bounced	Resigned	Complaint
0	32.98%	23.36%	9.25%	7.12%	0.23%	2.43%	0.16%	0.00%
1	5.89%	4.81%	1.76%	1.43%	7.48%	2.42%	0.07%	0.00%
2	28.66%	17.72%	2.92%	2.38%	0.44%	2.02%	2.15%	0.00%
3	8.51%	7.25%	2.53%	2.13%	0.22%	1.53%	0.05%	0.00%
4	1.10%	0.95%	0.13%	0.11%	0.12%	0.75%	0.01%	0.00%
5	1.62%	1.37%	0.20%	0.17%	0.09%	0.72%	0.01%	0.00%
6	128.65%	48.46%	43.69%	24.45%	0.29%	2.33%	0.62%	0.02%
7	1.23%	1.02%	0.30%	0.25%	0.47%	20.71%	0.04%	0.00%

3.2. Transaction Mails

As was mentioned earlier, in the case of transaction emails, the goal is not to abort a campaign, but to block a customer who sends a lot of bad transaction emails. Due to the nature of transaction emails, they tend to be sent to recipients whose emails should be confirmed and are in response to recipient actions. For this reason, the attributes resigned count and complaint count are not considered for them. Since in this case we do not have a list of recipients of a known size, it is necessary to count how many transaction e-mails a client has sent in a given period of time (the sent count attribute). Thus, for the given interval, we use seven conditional attributes: sent count, opens count, unique opens count, clicks count, unique clicks count, soft bounced count, and hard bounced count. The values of these attributes were aggregated for the following time intervals (numbers indicate minutes): $(0, 1]$, $(1, 2]$, $(2, 3]$, $(3, 4]$, $(4, 5]$, $(6, 10]$, $(10, 15]$, $(15, 20]$, $(20, 25]$, $(25, 30]$, $(30, 60]$, $(60, 120]$, $(120, 240]$, and $(240, 360]$. The values for intervals starting from $(6, 10]$ were used as conditional attributes for the model (63 attributes).

We have assumed that we have an undesirable phenomenon when there is a significant average (counting per minute) increase in the number of hard bounces in the last five minutes before a given transaction email is sent. For each customer, the average number of hard bounces per minute in the 48-hour preshipment period was determined. This number was used as the basis for determining the chain indices for the fifth, fourth, third, second, and preceding minutes, respectively. The geometric mean of the chain indices was then calculated to determine the average increase in the number of hard bounces per minute. We assumed that we label a record as bad if the value of this attribute exceeds 2 (100% average growth).

In the case of transaction emails, it is enough to limit oneself to hard bounces, as a large number of them indicate that the client is seemingly sending transaction emails to unverified recipients. A large number of hard bounces is a negative factor for the credibility of an ESP company, so it is important to minimise this phenomenon. Additionally, the analysis was limited to customers who send many transaction emails—above the median—as only in this group relatively large amounts of hard bounces were observed.

4. Artificial Neural Networks Models

As required by the SendGuard project, an ANN model was sought that would classify cases with an efficiency F1 score of at least 0.95. In both cases (campaigns and transaction emails), we used data samples containing at least 2 million objects. To balance the classes, the cases with the bad label were drawn at random with repetition. We used 75% of the

data sample for learning the model and 25% for testing. Additional data sets from periods different from the learning set were used to validate the model.

The models were developed using the *tensorflow.keras* Python library [19,20]. The Sequential model was used to develop the artificial neural networks.

4.1. Campaigns Analysis

The structure of the artificial neural network developed for classification of emails campaigns is shown in Figure 2. The input layer contains 24 neurons for the attributes described in Section 3.1. We used three hidden layers with 72, 96, and 72 neurons, respectively. For each hidden layer, the ReLU (REctified Linear Unit) activation function was applied. Finally, a single output neuron with the sigmoid activation function was used to provide the probability that the given campaign should be aborted. We applied the Adam optimiser (clipvalue = 0.5) and the MSE (the mean squared error) objective function.

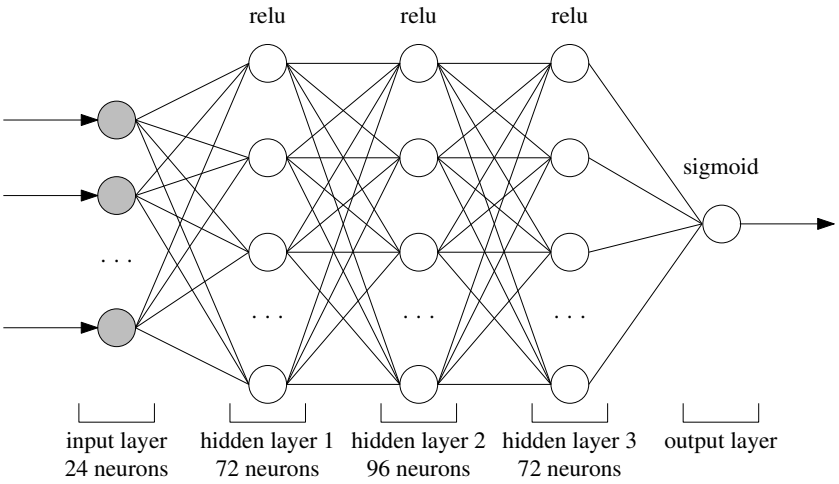

Figure 2. ANN structure for campaign classification model.

The StandardScaler function form Python sklearn library was applied to standardise the input data values. The classification report for test data is presented in Table 3.

Table 3. Classification report for model shown in Figure 2.

	Precision	Recall	F1-Score	Support
0	0.98	0.98	0.98	375,025
1	0.93	0.93	0.93	124,975
accuracy			0.97	500,000
macro avg	0.95	0.96	0.95	500,000
weighted avg	0.97	0.97	0.97	500,000

The model was additionally validated on data from periods other than the period from which the sample for learning the neural network was drawn. In each case, the F1-score value was above the required 0.95. It is worth noting that in the case of email campaigns, we treat each minute of sending as a separate record. If a decision is made to stop sending after a given minute, it can be assumed that the same decision applies to subsequent minutes. With this approach, F1-score increases by about 0.02.

4.2. Transaction Emails Analysis

The structure of the artificial neural network developed for classification of transaction emails is shown in Figure 3. The input layer contains 69 neurons for the attributes described in Section 3.2. This time, we used two hidden later with 48 neurons each. Like before, for each hidden layer, the ReLU activation function was applied and the sigmoid function was

used for the output neuron. We applied the Adam optimiser (clipvalue = 0.5) and MSE (mean squared error) objective function.

The classification report for test data are presented in Table 4. As previously stated, the model was validated on data from periods other than the period from which the sample for learning the neural network was drawn. In each case, the F1-score value was above the required 0.95.

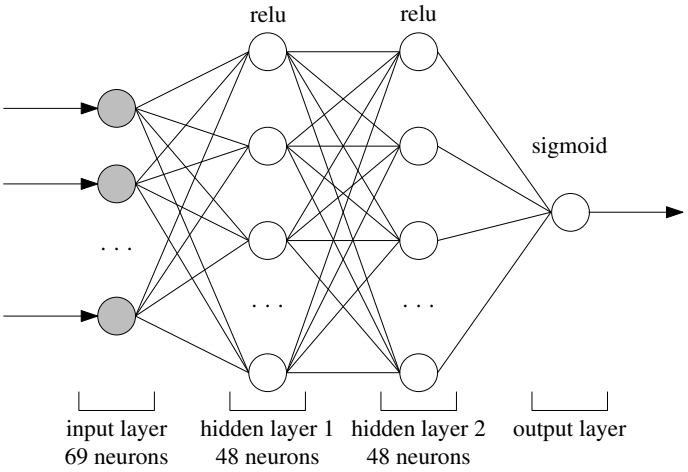

Figure 3. ANN structure for transaction emails classification model.

Table 4. Classification report for model shown in Figure 3.

	Precision	Recall	F1-Score	Support
0	0.98	0.98	0.98	250,123
1	0.98	0.98	0.98	249,877
accuracy			0.98	500,000
macro avg	0.98	0.98	0.98	500,000
weighted avg	0.98	0.98	0.98	500,000

5. Conclusions

The article presents artificial neural network models that can be used by an ESP company as part of the evaluation of the sending of email campaigns or a group of transaction emails sent by a given client. The key element of the proposed solutions are the conditional attributes used, i.e., instead of analysing the content of an email, we use statistics collected by the ESP system on the response of servers and subscribers (recipients). Such an approach allows us to use the same model regardless of the language of an email. The purpose of using the proposed models is to evaluate an email campaign (or a set of transaction emails) and identify bad cases. Since the bad label is associated with negative phenomena such as hard and soft bounces, users complaints, or resignations, it can be identified with the problem of detecting emails that we qualify as spam or phishing.

The solution presented is not intended to replace existing methods of analysing the quality of emails sent. Rather, it complements existing mechanisms. Experiments with data have shown that even as an independent method of evaluating email campaigns (transaction emails), the proposed solution is highly effective (the F1-score is above 0.95 for any used sample). In addition, the approach is rather simple, easy to implement, and reliable.

Author Contributions: Conceptualization, M.S., P.S., S.O. and G.J.N.; methodology, M.S. and P.S.; software, M.S., P.S. and S.O.; validation, M.S. and P.S.; investigation, M.S. and P.S.; resources, M.S.; data curation, P.S. and M.S.; writing—original draft preparation, M.S.; writing—review and editing, M.S. and G.J.N.; project administration, G.J.N. and S.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The National Centre for Research and Development, Poland, grant number NCBiR POIR.01.01.01-00-0202/19.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Karim, A.; Azam, S.; Shanmugam, B.; Kannoorpatti, K.; Alazab, M. A Comprehensive Survey for Intelligent Spam Email Detection. *IEEE Access* **2019**, *7*, 168261–168295. [CrossRef]
2. Muneer, A.; Ali, R.; Al-Sharai, A.; Fati, S. A Survey on Phishing Emails Detection Techniques. In Proceedings of the 2021 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 9–10 November 2021; pp. 1–6. [CrossRef]
3. Alkhalil, Z.; Hewage, C.; Nawaf, L.; Khan, I. Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Front. Comput. Sci.* **2021**, *3*, 563060. [CrossRef]
4. Gupta, B.B.; Arachchilage, N.A.; Psannis, K.E. Defending against phishing attacks: Taxonomy of methods, current issues and future directions. *Telecommun. Syst.* **2018**, *67*, 247–267. [CrossRef]
5. Abrahams, A.; Chaudhary, T.; Deane, J. A multi-industry, longitudinal analysis of the email marketing habits of the largest United States franchise chains. *J. Direct Data Digit. Mark. Pract.* **2010**, *11*, 187–197. [CrossRef]
6. Mostafa, R.; Norizan, M.Y.; Gazi, M.A. Impact of spam advertisement through e-mail: A study to assess the influence of the anti-spam on the e-mail marketing. *Afr. J. Bus. Manag.* **2010**, *4*, 2362–2367.
7. Ahmed, N.; Amin, R.; Aldabbas, H.; Koundal, D.; Alouffi, B.; Shah, T. Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges. *Secur. Commun. Netw.* **2022**, *2022*, 1862888. [CrossRef]
8. Dada, E.G.; Bassi, J.S.; Chiroma, H.; Abdulhamid, S.M.; Adetunmbi, A.O.; Ajibuwa, O.E. Machine learning for email spam filtering: Review, approaches and open research problems. *Heliyon* **2019**, *5*, e01802. [CrossRef] [PubMed]
9. Bansal, C.; Sidhu, B. Machine Learning based Hybrid Approach for Email Spam Detection. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 3–4 September 2021; pp. 1–4. [CrossRef]
10. Dhanaraj, S.; Karthikeyani, V. A study on e-mail image spam filtering techniques. In Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, Salem, India, 21–22 February 2013; pp. 49–55. [CrossRef]
11. Nam, S.G.; Jang, Y.; Lee, D.G.; Seo, Y.S. Hybrid Features by Combining Visual and Text Information to Improve Spam Filtering Performance. *Electronics* **2022**, *11*, 2053. [CrossRef]
12. Afzal, S.; Asim, M.; Javed, A.; Beg, M.; Baker, T. URLdeepDetect: A Deep Learning Approach for Detecting Malicious URLs Using Semantic Vector Models. *J. Netw. Syst. Manag.* **2021**, *29*, 21. [CrossRef]
13. Ozcan, A.; Catal, C.; Donmez, E.; Senturk, B. A hybrid DNN–LSTM model for detecting phishing URLs. *Neural Comput. Appl.* **2021** . [CrossRef] [PubMed]
14. Roy, S.S.; Awad, A.I.; Amare, L.A.; Erkihun, M.T.; Anas, M. Multimodel Phishing URL Detection Using LSTM, Bidirectional LSTM, and GRU Models. *Future Internet* **2022**, *14*, 340. [CrossRef]
15. 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]
16. Li, T.; Kou, G.; Peng, Y. Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods. *Inf. Syst.* **2020**, *91*, 101494. [CrossRef]
17. Iyengar, A.; Kalpana, G.; Kalyankumar, S.; GunaNandhini, S. Integrated SPAM detection for multilingual emails. In Proceedings of the 2017 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, 23–24 February 2017; pp. 1–4. [CrossRef]
18. Rastenis, J.; Ramanauskaitė, S.; Suzdalev, I.; Tunaityte, K.; Janulevicius, J.; Cenys, A. Multi-Language Spam/Phishing Classification by Email Body Text: Toward Automated Security Incident Investigation. *Electronics* **2021**, *10*, 668. [CrossRef]
19. Gulli, A.; Kapoor, A.; Pal, S. *Deep Learning with TensorFlow 2 and Keras*; Packt Publishing Ltd.: Birmingham, UK, 2019.
20. Patterson, J.; Gibson, A. *Deep Learning. A Practitional Approach*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2017.