**Email Spam Detection Using**

**Machine Learning Algorithms**

***Abstract:***

**Email Spam has become a major problem nowadays, with Rapid growth of internet users, Email spams is also increasing. People are using them for illegal and unethical conducts, phishing and fraud. Sending malicious link through spam emails which can harm our system and can also seek in into your system. Creating a fake profile and email account is much easy for the spammers, they pretend like a genuine person in their spam emails, these spammers target those peoples who are not aware about these frauds. So, it is needed to Identify those spam mails which are fraud, this project will identify those spam by using techniques of machine learning, this paper will discuss the machine learning algorithms and apply all these algorithm on our data sets and best algorithm is selected for the email spam detection having best precision and accuracy.**

1. INTRODUCTION

In recent years, the issue of unwanted bulk emails, commonly referred to as spam, has become a significant challenge on the internet. Spammers, the individuals responsible for sending these emails, often collect email addresses from websites, chatrooms, and malicious software [1]. Spam emails disrupt users by consuming valuable time, storage space, and network resources. The sheer volume of spam can overwhelm email servers, reduce communication efficiency, and waste computational resources [2]. Alarmingly, spam accounts for over 77% of global email traffic, with recipients finding these unsolicited messages both frustrating and intrusive. Many users have suffered financial losses due to fraudulent spam emails designed to deceive them into revealing sensitive personal information, such as passwords, bank details, or credit card numbers.

Reports from Kaspersky Lab indicate a decline in spam email volume in 2015, reaching its lowest level in 12 years. For the first time since 2003, spam emails accounted for less than 50% of global email traffic, dropping to 46.4% in mid-2015. This reduction was linked to a decrease in activity from major botnets that previously sent billions of spam emails. However, malicious spam emails containing malware and ransomware remained steady during this period. By late 2015, spam volumes began to rise again, with a sharp increase in December. This trend continued into 2016, with Kaspersky Lab identifying a fourfold increase in spam emails compared to the previous year. By March 2016, the number of spam emails detected reached nearly 23 million, making up 56.92% of global email traffic in the first quarter of that year. Healthcare and dating spam were among the most common types of spam emails. Additionally, spam imposes a burden on Simple Mail Transfer Protocol (SMTP) servers, which must process large volumes of unsolicited emails, further straining resources [52]. The prevalence of spam containing malicious code and attachments continued to grow between late 2016 and early 2018.

1. LITERATURE SURVEY

Spam emails have emerged as a significant challenge in the digital age, resulting in the wastage of resources and posing security risks to users. This section reviews previous studies, reports, and statistics on the evolution of spam emails and their impact on internet users and email infrastructure.

Spam emails are described as unsolicited bulk messages sent by spammers who harvest email addresses from online sources such as websites, chatrooms, and malicious software [1]. These emails negatively affect users by consuming their time, reducing storage capacity, and depleting network bandwidth. The overwhelming volume of spam traffic has detrimental effects on the performance of email servers, communication networks, and user productivity. Research reveals that spam accounts for more than 77% of global email traffic, emphasizing the magnitude of the problem [3]. Additionally, spam emails are often associated with fraudulent activities, including phishing scams where attackers impersonate reputable companies to extract sensitive user information, such as passwords, banking details, and credit card numbers.

Reports from Kaspersky Lab and Symantec indicate notable fluctuations in spam volume over the years. In 2015, the volume of spam emails fell to a 12-year low, accounting for less than 50% of global email traffic. This decline, attributed to the reduction in botnet activity, marked a significant milestone. However, malicious spam containing harmful attachments such as malware, ransomware, and JavaScript remained consistent. By December 2015, spam volume began to rise again, driven by an increase in emails carrying pernicious attachments. This upward trend continued into 2016, with Kaspersky Lab detecting nearly 23 million spam emails in March alone, representing 56.92% of global email traffic for the first quarter of that year.

Further studies highlight the shift in the nature of spam emails, with healthcare and dating-related spam becoming the most prevalent. The impact of spam extends beyond user inconvenience, as it imposes a heavy burden on Simple Mail Transfer Protocol (SMTP) servers, which must handle the substantial processing requirements of unsolicited emails [52]. Between late 2016 and early 2018, the volume of spam emails containing malicious code saw a significant increase, underscoring the growing sophistication of spamming techniques and the need for enhanced detection and mitigation strategies.

The findings from this survey indicate that while the volume of spam has fluctuated over time, its persistent presence and evolving nature require continuous monitoring and advancements in anti-spam technologies to safeguard users and networks from its adverse effects.

.

1. Survey of Existing System

**Neural Network-Based Methods** Neural networks have shown promising results in spam detection due to their ability to learn complex patterns. For instance, Radial Basis Function (RBF) neural networks combined with optimization techniques such as Particle Swarm Optimization (PSO) enhance classification accuracy by optimizing the network's parameters. Similarly, convolutional neural networks (CNNs) have been explored for their effectiveness in processing email content and identifying spam patterns.

**Support Vector Machines (SVMs)** SVMs have been widely used in spam detection due to their robust classification capabilities. They excel in distinguishing between spam and legitimate emails by leveraging discrete mixture-based kernels, as demonstrated in applications involving text categorization and email filtering. These methods are effective in handling high-dimensional data typically associated with email content.

**Bayesian Filtering** Bayesian filtering remains a classic approach to spam detection. By analyzing the probabilistic relationship between words and spam, this method classifies emails based on the likelihood of their content being spam. While effective, it can struggle with evolving spam tactics that introduce novel vocabulary.

**Instance-Based Reasoning Systems** Instance-based reasoning systems, such as SpamHunting, use historical examples to classify incoming emails. By comparing new emails with previously labeled examples, these systems provide accurate results. However, their dependency on extensive labeled datasets can pose scalability challenges.

**Hybrid Techniques** Hybrid systems combine multiple approaches to leverage their strengths. For example, integrating neural networks with clustering algorithms enhances spam detection by addressing the weaknesses of individual methods. These systems often achieve higher accuracy and robustness compared to single-method approaches.

**Machine Learning Approaches** Various machine learning algorithms, including Random Forests and clustering techniques, have been applied to spam detection. These algorithms excel in processing large datasets and identifying intricate patterns in email metadata and content. However, their performance heavily depends on feature selection and dataset quality.

1. Limitation of existing system

**Data Preprocessing Assumptions**: The code assumes that all missing values in the dataset can be safely replaced with empty strings. This might not be the most effective approach in certain cases, leading to data distortion or loss of important information.

**Limited Feature Extraction**: The TfidfVectorizer is configured with basic parameters like min\_df=1 and stop\_words='english', which may not capture all nuances of the text. This can limit the performance of the model, especially for more complex language patterns or domain-specific spam.

**Single Model Approach**: The system uses only a Logistic Regression model, which may not perform as well for all types of spam detection tasks. Exploring other models (e.g., Random Forest, Naive Bayes, or deep learning models) might provide better results, especially for complex spam classification tasks.

**No Hyperparameter Tuning**: The model is trained with default settings without any hyperparameter optimization. Tuning hyperparameters (e.g., regularization strength for Logistic Regression) could improve the model's accuracy and generalization.

**Overfitting Risk**: Without cross-validation or more robust validation strategies, there is a risk of overfitting the model to the training data, leading to poor generalization on unseen data.

**Imbalanced Classes**: If the dataset has a significant imbalance between spam and ham emails, the model's performance may be biased toward predicting the majority class. Techniques like class weighting or oversampling might be needed to address this issue.

**No Evaluation on Test Set**: The prediction and evaluation steps are done on training data (prediction\_on\_training\_data) but not on the test data. This might lead to an overestimation of the model's performance.

**Scalability**: If the dataset grows significantly in size, the current approach might not scale well, especially during the feature extraction phase. More efficient techniques or using more powerful models may be required for larger datasets.

**Limited Metrics for Evaluation**: While accuracy is computed, other metrics like precision, recall, and F1-score are only mentioned for the test data but aren't included in the evaluation loop.

1. .Problem Statement and Objectives

**5.1 Problem Statement:**

The primary goal of this project is to build an effective spam email detection system using machine learning techniques. Spam emails, also known as junk emails, have a significant impact on both personal and business communication. These emails often contain irrelevant content, advertisements, or malicious links that can harm the recipient's system. The objective is to create a classification model that can accurately distinguish between spam (unwanted) and ham (legitimate) emails to automate email filtering systems and improve user experience.

The current systems might struggle with classifying complex spam emails, and they often rely on manual rule-based methods or simple heuristics, which may be less effective as spam tactics evolve. By leveraging machine learning models like Logistic Regression, this project aims to improve the accuracy of spam classification and provide a scalable solution that can handle large datasets with dynamic characteristics.

**5.2 Objectives:**

1. **Data Preprocessing**: Clean and preprocess the email dataset to handle missing values and ensure that the input data is ready for machine learning.
2. **Feature Extraction**: Use TF-IDF (Term Frequency-Inverse Document Frequency) to convert email content (text) into numerical feature vectors that can be used as input to the machine learning model.
3. **Model Training**: Train a machine learning model (Logistic Regression) on the preprocessed data to classify emails as spam or ham.
4. **Model Evaluation**: Evaluate the model's performance using appropriate metrics such as accuracy, precision, recall, and F1-score to ensure that it can generalize well to unseen data.
5. **Optimize Model Performance**: Identify and address any issues related to class imbalance, overfitting, or underfitting by exploring techniques such as hyperparameter tuning, cross-validation, and data augmentation.
6. **Deployment**: Save the trained model and prepare it for deployment, so it can be used in real-time email filtering applications.
7. **Scalability and Flexibility**: Ensure the system can scale and adapt to new types of spam emails by periodically retraining the model with updated data.

By achieving these objectives, the system will help users automatically filter out spam emails, reducing the risk of malware, enhancing productivity, and improving the overall email experience.

6.0 Scope

The scope of this project focuses on developing a machine learning-based spam email detection system with the following key areas:

1. **Email Classification**:
   * The primary focus is on classifying emails into two categories: **spam** and **ham** (legitimate).
   * The system will rely on existing datasets, such as the commonly used "spam.csv" dataset, to train and test the model.
   * Emails are represented as text, and only textual content is considered for classification, excluding any attachments, images, or links.
2. **Preprocessing**:
   * The project will preprocess the email data by handling missing values, normalizing the text (e.g., converting to lowercase), and removing stop words to improve model accuracy.
   * Tokenization and feature extraction will be done using TF-IDF to capture relevant textual information.
3. **Model Development**:
   * The Logistic Regression model will be used as the primary classifier to distinguish between spam and ham emails.
   * Other classification models (e.g., Naive Bayes, Random Forest) may be explored for comparison, but Logistic Regression will be the primary focus.
4. **Evaluation Metrics**:
   * The performance of the model will be assessed using metrics like accuracy, precision, recall, and F1-score.
   * A confusion matrix will also be used to visualize the model's classification performance.
5. **Limitations**:
   * The project will focus on emails in the English language and may not handle multilingual datasets or domain-specific emails well.
   * The dataset size is fixed (based on the available data in the CSV file), which may limit scalability.
   * The project will not include features like detecting phishing links, malware, or attachments, as it focuses solely on the text classification of spam emails.
6. **Deployment**:
   * The trained model will be saved and exported as a pickle file, making it ready for deployment in email filtering systems.
   * The project does not include a fully-fledged email client interface but focuses on the backend model for spam detection.
7. **Scalability and Future Improvements**:
   * While the current system is built for a single static dataset, it can be extended to incorporate dynamic learning techniques where the model is retrained periodically with new data to adapt to evolving spam tactics.
   * Potential future enhancements could include multi-class classification (e.g., categorizing spam into various types), integration with real-time email systems, or the use of deep learning models for more accurate predictions.

In summary, the scope is confined to creating a spam detection system using text-based machine learning techniques, with a focus on classification, evaluation, and deployment of the model for practical use in filtering spam emails.

7.0 Proposed System

**Proposed System: Gmail Spam Detection System with Flask and React Integration**

**System Overview:**

The proposed system is designed to automatically classify incoming emails from Gmail as either **spam** or **ham** (legitimate) using a pre-trained machine learning model. It integrates Gmail with OAuth2 authentication and IMAP protocols for accessing the user's inbox and utilizes a **Logistic Regression model** trained on a spam dataset. The system's architecture includes a **Flask backend** for server-side logic and a **React frontend** for a user-friendly interface. The spam classification process is done using **TF-IDF** vectorization to convert the email subjects into numerical features, which are then classified using the trained model.

**7.1 Key Components:**

1. **Gmail Authentication (OAuth2 + IMAP)**:
   * The system allows users to authenticate with their Gmail accounts using **OAuth2**. This process ensures secure access to the user's inbox without exposing sensitive login credentials.
   * The system uses the **IMAP protocol** to fetch emails from the user's Gmail inbox once the user is authenticated, enabling the system to read email metadata like subject, sender, and date.
2. **Machine Learning Model (Spam Detection)**:
   * A pre-trained **Logistic Regression** model is used to classify email subjects as either spam or ham. The model has been trained using a dataset of labeled emails (spam and ham) and has learned to identify patterns in the email subjects.
   * The **TF-IDF (Term Frequency-Inverse Document Frequency)** technique is used for feature extraction, converting the text of email subjects into a sparse matrix of features that can be fed into the logistic regression model for classification.
3. **Backend (Flask)**:
   * **Flask** serves as the backend framework for handling requests, authenticating Gmail users, fetching emails, and running the classification model.
   * It handles **API requests** from the React frontend and returns the results, including the list of emails and their spam classification.
   * The backend also manages the integration with **Google APIs** for Gmail access and **model loading** using the joblib package.
4. **Frontend (React)**:
   * **React** is used to build an interactive web interface where users can log in to their Gmail account, view their email subjects, and see whether each email is classified as spam or ham.
   * The frontend communicates with the Flask backend via API calls to fetch the emails and display their classification results in real-time.

**7.2System Workflow:**

1. **User Authentication**:
   * The user logs in to the system via Google OAuth2, granting the application permission to access their Gmail inbox.
   * Once authenticated, the system stores the access and refresh tokens to allow seamless and secure access to the Gmail account without needing to log in again.
2. **Fetching Emails from Gmail**:
   * After authentication, the system uses the **IMAP protocol** to fetch all emails from the user's inbox.
   * The system extracts metadata such as the subject and sender from each email to prepare it for classification.
3. **Spam Detection**:
   * The email subjects are passed through the **TF-IDF Vectorizer** to convert the text into numerical features.
   * The **Logistic Regression model** is used to predict whether each email is spam or ham based on the extracted features.
   * The prediction result is either "Spam" or "Not Spam," which is returned by the Flask backend to the React frontend.
4. **Frontend Display**:
   * The **React frontend** displays the list of emails with their subject lines and their classification status (Spam/Not Spam).
   * The user can interact with the interface, view detailed email information, and check the spam status of each email.
5. **Continuous Model Improvement**:
   * The system can be periodically retrained with updated datasets to adapt to changing spam patterns, ensuring that the spam detection remains accurate over time.

**7.3 Detailed Architecture:**

1. **Backend (Flask)**:
   * **API Endpoint**: A /api/emails endpoint is exposed by Flask, which the React frontend will call to fetch the emails and their spam classification results.
   * **Google Authentication**: The backend handles the OAuth2 authentication flow and token management, allowing secure access to Gmail.
   * **IMAP Email Fetching**: Once authenticated, the system uses the imaplib library to fetch emails from the Gmail inbox and extract the relevant metadata (subject).
   * **Model Inference**: The pre-trained spam detection model is loaded from disk using joblib, and the Flask backend applies the model to the email subjects to classify them as spam or ham.
2. **Frontend (React)**:
   * **User Interface**: The frontend provides a user-friendly interface where users can view a list of their emails along with their spam classification status.
   * **Real-time Data Fetching**: Using axios or fetch, the frontend makes asynchronous API requests to the Flask backend to retrieve the email data and display it dynamically in the UI.
   * **Responsive Design**: The frontend is designed to be responsive, ensuring a smooth experience on both desktop and mobile devices.
3. **Spam Detection Model**:
   * **Preprocessing**: Email subjects are processed using **TF-IDF** to convert them into feature vectors that the model can understand.
   * **Model**: The **Logistic Regression model** is trained on a labeled dataset (spam vs. ham emails) and is saved using joblib. This allows for easy loading and use during inference.
   * **Prediction**: The model outputs a binary classification (1 for ham, 0 for spam), which is sent back to the frontend for display.

**7.4 Technologies Used:**

* **Backend**:
  + **Flask**: For serving API requests, handling authentication, and managing model inference.
  + **IMAP**: To fetch emails from Gmail using the imaplib library.
  + **joblib**: For loading the pre-trained machine learning model.
  + **Google APIs**: For Gmail authentication using OAuth2.
* **Frontend**:
  + **React**: For building the dynamic user interface.
  + **Axios/Fetch**: For making API calls to the Flask backend.
  + **HTML/CSS**: For designing the frontend layout and ensuring a user-friendly interface.
* **Machine Learning**:
  + **Scikit-learn**: For training the **Logistic Regression** model and using **TF-IDF** for feature extraction.

14. Bibliograph

Awad, M. ∙ Foqaha, M.

**Email spam classification using hybrid approach of RBF neural network and particle swarm optimization**

*Int. J. Netw. Secur. Appl.* 2016; **8**

[Google Scholar](https://scholar.google.com/scholar?q=M.AwadM.FoqahaEmail+spam+classification+using+hybrid+approach+of+RBF+neural+network+and+particle+swarm+optimizationInt.+J.+Netw.+Secur.+Appl.842016)

[2.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref2)

Fonseca, D.M. ∙ Fazzion, O.H. ∙ Cunha, E. ...

**Measuring characterizing, and avoiding spam traffic costs**

*IEEE Int. Comp.* 2016; **99**

[Google Scholar](https://scholar.google.com/scholar?q=D.M.FonsecaO.H.FazzionE.CunhaI.Las-CasasP.D.GuedesW.MeiraM.ChavesMeasuring+characterizing%2C+and+avoiding+spam+traffic+costsIEEE+Int.+Comp.992016)

[3.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref3)

Visited on May 15, 2017*Kaspersky Lab Spam Report.* 2017;, 2017 2012

<https://www.securelist.com/en/analysis/204792230/Spam_Report_April_2012>

[Google Scholar](https://scholar.google.com/scholar?q=Visited+on+May+15%2C+2017Kaspersky+Lab+Spam+Report20172012https%3A%2F%2Fwww.securelist.com%2Fen%2Fanalysis%2F204792230%2FSpam_Report_April_2012)

[4.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref4)

Bahgat, E.M. ∙ Rady, S. ∙ Gad, W.

**An e-mail filtering approach using classification techniques**

**The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November 28-30, 2015**

Springer International Publishing, BeniSuef, Egypt, 2016; 321-331

[Crossref](https://doi.org/10.1007/978-3-319-26690-9_29)

[Scopus (17)](https://www.cell.com/servlet/linkout?suffix=e_1_5_1_2_4_2&dbid=137438953472&doi=10.1016%2Fj.heliyon.2019.e01802&key=2-s2.0-84951960978)

[Google Scholar](https://scholar.google.com/scholar_lookup?doi=10.1007%2F978-3-319-26690-9_29)

[5.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref5)

Bouguila, N. ∙ Amayri, O.

**A discrete mixture-based kernel for SVMs: application to spam and image categorization**

*Inf. Process. Manag.* 2009; **45**:631-642

[Crossref](https://doi.org/10.1016/j.ipm.2009.05.005)

[Scopus (47)](https://www.cell.com/servlet/linkout?suffix=e_1_5_1_2_5_2&dbid=137438953472&doi=10.1016%2Fj.heliyon.2019.e01802&key=2-s2.0-69249211134)

[Google Scholar](https://scholar.google.com/scholar_lookup?doi=10.1016%2Fj.ipm.2009.05.005)

[6.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref6)

Cao, Y. ∙ Liao, X. ∙ Li, Y.

**An e-mail filtering approach using neural network**

**International Symposium on Neural Networks**

Springer Berlin Heidelberg, 2004; 688-694

[Crossref](https://doi.org/10.1007/978-3-540-28648-6_110)

[Scopus (15)](https://www.cell.com/servlet/linkout?suffix=e_1_5_1_2_6_2&dbid=137438953472&doi=10.1016%2Fj.heliyon.2019.e01802&key=2-s2.0-24944452752)

[Google Scholar](https://scholar.google.com/scholar_lookup?doi=10.1007%2F978-3-540-28648-6_110)

[7.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref7)

Fdez-Riverola, F. ∙ Iglesias, E.L. ∙ Diaz, F. ...

**SpamHunting: an instance-based reasoning system for spam labelling and filtering**

*Decis. Support Syst.* 2007; **43**:722-736

[Crossref](https://doi.org/10.1016/j.dss.2006.11.012)

[Scopus (74)](https://www.cell.com/servlet/linkout?suffix=e_1_5_1_2_7_2&dbid=137438953472&doi=10.1016%2Fj.heliyon.2019.e01802&key=2-s2.0-33947513003)

[Google Scholar](https://scholar.google.com/scholar_lookup?doi=10.1016%2Fj.dss.2006.11.012)

[8.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref8)

Mason, S.

**New Law Designed to Limit Amount of Spam in E-Mail**

2003

<http://www.wral.com/technolog>

[Google Scholar](https://scholar.google.com/scholar?q=S.MasonNew+Law+Designed+to+Limit+Amount+of+Spam+in+E-Mail2003http%3A%2F%2Fwww.wral.com%2Ftechnolog)

[9.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref9-1)

Stuart, I. ∙ Cha, S.H. ∙ Tappert, C.

**A neural network classifier for junk e-mail**

**Document Analysis Systems VI**

Springer Berlin Heidelberg, 2004; 442-450

[Crossref](https://doi.org/10.1007/978-3-540-28640-0_42)

[Scopus (26)](https://www.cell.com/servlet/linkout?suffix=e_1_5_1_2_9_2&dbid=137438953472&doi=10.1016%2Fj.heliyon.2019.e01802&key=2-s2.0-35048826302)

[Google Scholar](https://scholar.google.com/scholar_lookup?doi=10.1007%2F978-3-540-28640-0_42)

[10.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref10)

Han, J. ∙ Kamber, M. ∙ Pei, J.

**Data Mining: Concepts and Techniques**

Elsevier, 2011

[Google Scholar](https://scholar.google.com/scholar?q=J.HanM.KamberJ.PeiData+Mining%3A+Concepts+and+Techniques2011Elsevier)

[11.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref11)

Qasem, S.N. ∙ Shamsuddin, S.M. ∙ Zain, A.M.

**Multi-objective hybrid algorithms for radial basis function neural network design**

*Knowl. Based Syst.* 2012; **27**:475-497

[Crossref](https://doi.org/10.1016/j.knosys.2011.10.001)

[Scopus (69)](https://www.cell.com/servlet/linkout?suffix=e_1_5_1_2_11_2&dbid=137438953472&doi=10.1016%2Fj.heliyon.2019.e01802&key=2-s2.0-84855977752)

[Google Scholar](https://scholar.google.com/scholar_lookup?doi=10.1016%2Fj.knosys.2011.10.001)

[12.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref12)

Schaffer, J.D. ∙ Whitley, D. ∙ Eshelman, L.

**Combinations of genetic algorithms and neural networks: a survey of the state of the art**

Combinations of Genetic Algorithms and Neural Networks1992 1-37

[Google Scholar](https://scholar.google.com/scholar?q=J.D.SchafferD.WhitleyL.EshelmanCombinations+of+genetic+algorithms+and+neural+networks%3A+a+survey+of+the+state+of+the+artCombinations+of+Genetic+Algorithms+and+Neural+Networks1992137)

[13.](https://www.cell.com/heliyon/fulltext/S2405-8440(18)35340-4#body-ref-sref13)

Elbeltagi, E. ∙ Hegazy, T. ∙ Grierson, D.

**Comparison among five evolutionary-based optimization algorithms**

*Adv. Eng. Inf.* 2005; **19**:43-53

[Crossref](https://doi.org/10.1016/j.aei.2005.01.004)

[Scopus (1171)](https://www.cell.com/servlet/linkout?suffix=e_1_5_1_2_13_2&dbid=137438953472&doi=10.1016%2Fj.heliyon.2019.e01802&key=2-s2.0-22344432656)

[Google Scholar](https://scholar.google.com/scholar_lookup?doi=10.1016%2Fj.aei.2005.01.004)