Temp Email Creation Analysis and Detection: A Hybrid Machine Learning Approach

A PROJECT REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE COMPLETION OF CS4200-MAJOR PROJECT

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

Katankur Uday Kumar
(Enrollment No. 21CS002386)

FACULTY OF COMPUTING AND INFORMATICS
SIR PADAMPAT SINGHANIA UNIVERSITY
UDAIPUR 313601, INDIA

JAN, 2025

Temp Email Creation Analysis and Detection: A Hybrid Machine Learning Approach

A Project Report
Submitted in partial fulfillment of the requirements
for CS4200-Major Project

BACHELOR OF TECHNOLOGY
in
Computer Science & Engineering

Submitted by

Katankur Uday Kumar
(Enrollment No. 21CS002386)

Under the guidance of
Prof. Alok Kumar
(Project Coordinator)
and
Dr. Chandini Joshi

FACULTY OF COMPUTING AND INFORMATICS
SIR PADAMPAT SINGHANIA UNIVERSITY
UDAIPUR 313601, India

JAN, 2025

Faculty of Computing and Informatics
Sir Padampat Singhania University
Udaipur, 313601, India

CERTIFICATE

I, **KATANKUR UDAY KUMAR**, hereby declare that the work presented in this project report entitled **"Temp Email Creation analysis and Detection: A Hybrid Machine Learning Approach "** for the completion of CS4200-Major Project and submitted in the **Faculty of Computing and Informatics** of the **Sir Padampat Singhania University, Udaipur** is an authentic record of my own work carried out under the supervision of **Prof. Alok Kumar, Professor**, and **Dr. Chandani joshi, Designation**. The work presented in this report has not been submitted by me anywhere else.

katankur uday kumar
(21CS002386)

This is to certify that the above statement made by the candidate is true to the best of my knowledge and belief.

Prof. Alok Kumar
Professor
Project Coordinator

Dr. chandani joshi
Professor
project Supervisor

Place: Udaipur
Date:

Acknowledgements

Inscribing these words of gratitude feels akin to painting a masterpiece on the canvas of appreciation. This incredible path of learning and exploration would not have been possible without the unflinching support and encouragement of the great individuals who have paved the road for my accomplishment.

I reserve a special place in my heart for my beloved parents, whose unwavering love, unwavering support, and unwavering belief in my abilities have been the bedrock upon which my dreams have flourished. Their persistent support, sacrifices, and unshakable trust in my abilities have been the driving factors behind my quest for knowledge and academic pursuits.

First and foremost, I owe a tremendous debt of gratitude to my esteemed supervisor, **Prof. Alok Kumar** , whose guidance and advice have been the compass guiding me through the many twists and turns of this thesis. His stimulating conversations, in sightful feedback, kind advice, and boundless forbearance have challenged me to push the boundaries of my capabilities and inspired me to strive for academic excellence. I am very thankful for the trust you put in me and the chances you gave me to grow both professionally and personally. I am grateful beyond words for the opportunity to have worked under your guidance, and I hope my thesis serves as a fitting tribute to your hard work, knowledge, and encouragement.

I like to thank **Dr. AMit Kumar Goel** , Dean, Faculty of computing and in formatics Department, and **Dr. Chandrashekhar Goswami** ,Deputy Dean, Faculty of computing and informatics Department, for their extended support.

I would like to extend a heartfelt thank you to,**Mr. vimal kumar** my incredible class mates and friends, who have been a constant source of support, camaraderie, and inspi ration. Their presence has made the often-trying process of writing a thesis into one that is filled with joy and fun. Finally, I want to thank everyone who helped me grow as a scholar and made this trip unforgettable.

katankur uday kumar

Abstract

This project delves into creating temporary email addresses and detecting disposable email addresses (DEA) that combines natural language processing (NLP) techniques and domain verification methods to create temporary email addresses. These are usually disposable to preserve user privacy on online platforms. Email service is used Even though these services provide legitimate interests, such as protecting individual email accounts from spam, These services also present significant challenges. Including the increased risk of spam. Fraudulent registration and circumventing platform limitations...

Traditional blacklist-based methods to identify disposable email domains face challenges such as false positives, high latency. and limited scalability due to the rapidly evolving domain space. This project addresses these issues using a hybrid approach. Including machine learning Email language analysis Domain reputation rating and integrate real-time monitoring mechanisms.

The proposed system leverages NLP to identify patterns in email addresses and domain names that characterize disposable email services. Additionally, domain verification techniques evaluate the validity and trustworthiness of email domains through DNS records, MX server verification (email exchange) and domain age analysis Hybrid frameworks not only improve search accuracy; But it is also adapting to the emerging disposable email service… also increased By combining advanced machine learning techniques with domain-level verification.

This project aims to This project aims to develop strong solutions. Effectively And can adjust the size In order to deal with the more concerns about the use email address The results of this research will be very useful for the platform that needs to improve the process of inspecting users to reduce spam. And maintain the balance between the privacy and safety of users .

keywords:*Temporary Email Addresses,Disposable Email Addresses (DEAs),Hybrid Approach,Natural Language Processing (NLP),Domain Validation Techniques,Privacy,Spam,Fraudulent Activities,Blacklist-Based Detection,False Positives,Scalability Issues*

Contents

List of Figures

List of Tables

List of Abbreviations

DNS	Domain Name System
SVM	Support Vector Machine
MNB	Multinomial Naive Bayes
API	Application Programming Interface
SVC	Support Vector Classifier

Chapter 1

Introduction

1.1 Background

A lot of people are using email to talk to customer service. It helps. Customers stop waiting on the phone and keep a record of their conversation. Company. Email addresses are important pieces of personal information online. Some people use disposable email services to register accounts instead… Based on the actual use of email [1] , there is a common way to send email. Information. Spammers send questionable content via email, so look out for it. Using these spam emails is important for protecting personal information. There are various methods for this task. One method involves using Sequential Minimum Optimization (SMO), which is a form of machine learning [2] Classify emails as spam or not [2] Feature extraction techniques Niques help identify important information during the search process. The sequential minimum optimization (SMO) algorithm then uses this selection. Information for classifying emails This method is effective for spam management and Protect It [3] Email is an important way for people to communicate. Spammers send unwanted emails to users. To keep your email safe, we They must be protected from unauthorized access.

Disposable email addresses (DEAs) are widely used for temporary registrations, ensuring user privacy and anonymity. Services like Guerrilla Mail and Temp Mail allow users to bypass mandatory sign-ups without exposing their primary email addresses. While beneficial for users, DEAs pose significant challenges for businesses, including fraudulent registrations, spam, and security vulnerabilities. Organizations relying on email verification struggle to maintain authentic user interactions, as DEAs can be used to exploit promotional offers, manipulate marketing metrics, and evade security checks.

1.1.1 Addressing the Temporary Email Generation

developing a temporary email generation system that enables users to create secure and disposable email addresses while incorporating robust security measures to prevent misuse. [3] The system is designed to provide privacy and anonymity, allowing users to send and receive emails within a limited timeframe while ensuring protection against spam, fraud, and mass exploitation.

By integrating backend email server configurations, dynamic mailbox allocation, secure email processing, user-friendly interfaces, and advanced monitoring mechanisms, this system ensures efficient, controlled, and scalable temporary email usage. Additionally, it incorporates AI-driven security checks and rate-limiting features to detect and prevent automated abuse, making it a reliable and secure solution for temporary email services. [4]

1.2 Problem Statement & Objectives

1.2.1 Problem Statement

The traditional approach to detecting disposable email addresses (DEAs) relies on blacklist databases that store known temporary email domains. However, these methods have major limitations:

Limited scalability – Blacklists require constant updates to track new DEA domains. False positives – Legitimate domains may be wrongly classified as disposable. Delayed detection – New DEA providers can operate undetected for extended periods. Bypassing detection – Some services rotate domains frequently to avoid being blacklisted. Due to these challenges, businesses and cybersecurity teams struggle to detect and block disposable emails effectively. Thus, a more intelligent, automated, and adaptable detection method is required. [5]

1.2.2 Objectives

In order to achieve this aim the following objectives have been laid,

(i) To develop a system for creating and managing temporary email addresses, ensuring user privacy while providing robust spam and fraud protection

(ii) To develop a hybrid NLP and domain validation technique capable of accurately detecting disposable email addresses, thereby enhancing email security and reducing spam. [6]

(iii) Minimizing fraudulent activities by detecting and blocking disposable email addresses.

(iv) Improving real-time detection to identify new DEA providers as they emerge.

Chapter 2

Literature Review

The Temporary email services, also known as Disposable Email Addresses (DEAs), have gained immense popularity in recent years. These services allow users to create temporary, self-destructing email addresses, offering anonymity and protection from spam. Popular platforms like Guerrilla Mail and Temp Mail enable users to bypass mandatory registration processes without exposing personal email accounts. [7] While these services cater to user privacy concerns, they also present significant challenges for businesses and organizations aiming to ensure authentic user interactions. Temporary emails are often exploited for fraudulent registrations, spamming, and bypassing security measures. Businesses face issues such as low email open rates, loss of valuable leads, compromised email marketing metrics, and the misuse of temporary addresses for fake reviews or exploiting promotional offers.

The Disposable email addresses (DEAs) present significant challenges for organizations, particularly in ensuring user authenticity and minimizing spam. Traditional detection methods primarily rely on blacklists of known DEA domains, but these approaches have limitations. Blacklists require frequent manual updates to stay relevant and often fail to detect new or previously unknown domains. Consequently, they struggle with scalability and may result in false positives, impacting legitimate users. [8]

Recent advancements in machine learning (ML) and natural language processing (NLP) have demonstrated their potential to address these limitations. Machine learning techniques, such as Support Vector Machines (SVM), Random Forests, and Naive Bayes, have proven effective in text classification and spam detection tasks. These methods analyze email content and structural characteristics, enabling more dynamic and accurate detection of disposable email addresses. For example, NLP techniques examine patterns,

such as keywords like ”temporary” or ”disposable,” and analyze linguistic features to identify DEAs with high precision. [9]

The proposed hybrid approach integrates NLP with domain validation techniques, including DNS verification, domain age analysis, and reputation checks. This combination addresses the shortcomings of traditional blacklist-based systems, offering greater adaptability to new DEA providers and reducing false positives. By automating the detection process, the hybrid model enhances scalability and efficiency while achieving impressive accuracy levels, reaching up to 97

Research has shown that hybrid models outperform traditional methods by effectively balancing accuracy, adaptability, and computational efficiency. [10] The integration of machine learning and domain validation techniques provides a robust framework for real-time DEA detection, significantly improving the reliability of spam and fraud prevention systems. This approach lays the groundwork for further advancements in disposable email detection and secure communication.

Chapter 3

Methodology Adopted

3.1 Development of temporary email addresses

Description:

The development of a temporary email generation system follows a structured approach to ensure seamless functionality, security, and scalability. [11] The process begins with backend email server configuration, where a dedicated mail server is set up to handle temporary email traffic. This involves configuring MX (Mail Exchange) records, SMTP, IMAP, and POP3 protocols for efficient email communication. To enhance performance and availability, cloud-based infrastructure is deployed, ensuring the system can handle high email traffic loads. [12]

Next, dynamic mailbox allocation is implemented to allow the automatic generation of unique and randomized email addresses upon user request. These temporary mailboxes are designed to exist for a predefined duration, after which they are automatically deleted to free up resources. [13] The system ensures users can receive emails within this timeframe while preventing unauthorized access and mass misuse.

To enhance security, email processing and anti-abuse mechanisms are integrated. Incoming emails are filtered to detect and block spam, phishing attempts, and malware. Encryption techniques are applied to secure messages and user interactions, while rate-limiting policies prevent excessive email generation by bots or malicious users. [14] Fraud detection algorithms are also employed to identify suspicious activities and prevent abuse of the temporary email service.

For ease of access, a user-friendly frontend interface is developed, allowing users to generate, access, and manage temporary email addresses. Additionally, an API service

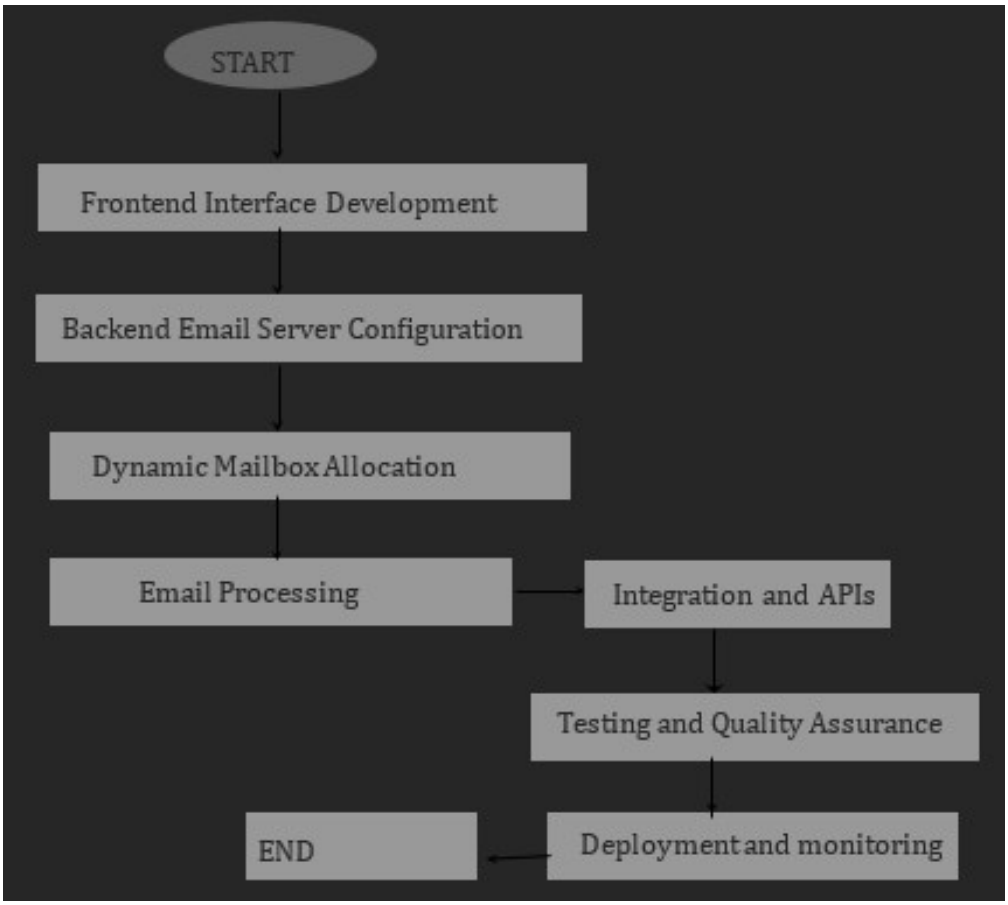

Figure 3.1: Development of temporary email addresses

is provided to enable third-party platforms and businesses to verify email legitimacy in real time. [15] Email forwarding options may also be integrated, allowing users to transfer important messages to a permanent inbox. Security measures such as OAuth-based authentication help enhance the protection of user sessions.

Once the system is developed, deployment and continuous monitoring are critical to ensure stability and security. The temporary email service is hosted on cloud servers for scalability, while logging and analytics tools are employed to track system performance and detect anomalies. [16] AI-driven security checks analyze user behavior, flagging any unusual patterns for further review. Regular updates are conducted to keep the database of known disposable email domains up to date, enhancing the detection system's accuracy.

Finally, testing and quality assurance play a key role before launching the system. Unit tests ensure that email generation, reception, and expiration work as intended, while performance testing evaluates the system under high loads. [12] Security testing helps identify vulnerabilities, ensuring that the platform is resistant to cyber threats. Additionally, user experience testing refines the web interface and API usability for a smoother interaction.

To ensure continuous improvement, the system can later be enhanced with AI-driven spam detection, mobile-friendly interfaces, multi-language support, and advanced domain validation techniques for more accurate DEA detection. This structured approach

ensures that the temporary email service remains secure, scalable, and efficient, balancing user privacy with fraud prevention in an increasingly digital landscape.

3.2 Development of disposable email detector

Description:

The process of developing a disposable email detector involves several key stages. First, data collection is conducted, gathering a dataset that includes both disposable and legitimate email addresses, which will be used for training and evaluating the detector. Once the data is collected, it undergoes NLP preprocessing, which involves cleaning and transforming the email addresses into a format suitable for the detector. Common techniques include tokenization (breaking down the email addresses into characters or words), lowercasing for consistency, removing special characters, and handling any missing values. Next, relevant features are extracted from the preprocessed email addresses, including lexical features (such as length, domain name, and the presence of special characters), syntactic features (like hyphens, underscores, and dots), and semantic features (such as domain reputation and usage patterns). With these features in hand, a machine learning model is trained to distinguish between disposable and legitimate email addresses using algorithms like decision trees, support vector machines (SVM), or Naive Bayes. reliability of email systems.After training, the model is evaluated using metrics such as accuracy, precision, recall, and F1-score to assess its performance. Finally, the detector undergoes domain validation to ensure its robustness and accuracy within the specific context or environment in which it will be deployed. By following these steps, a reliable disposable email detector can be created to filter out temporary email addresses and enhance the security and

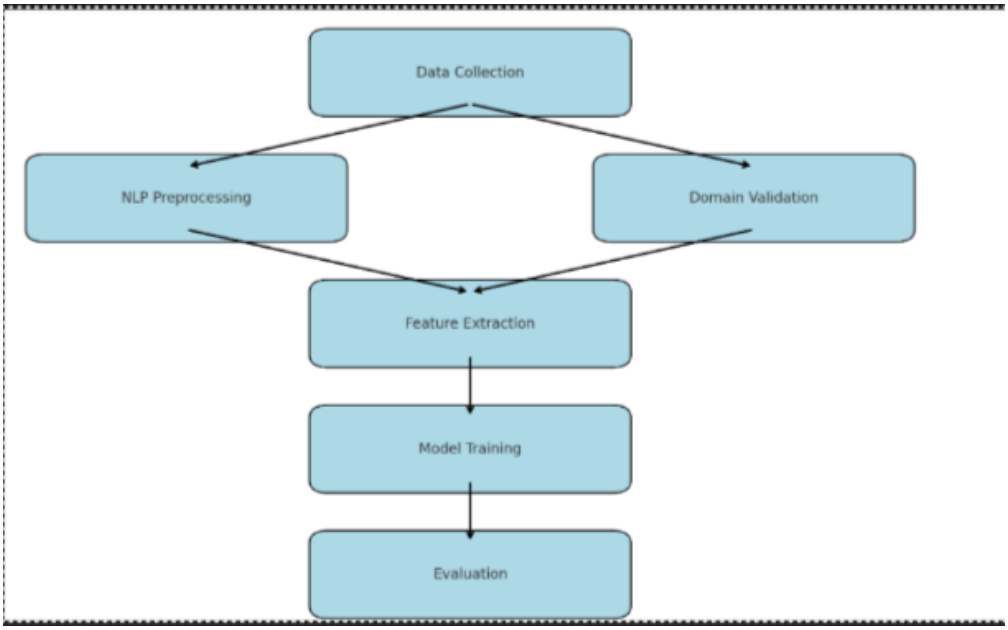

Figure 3.2: Development of disposable email detector

Chapter 4

Results and Discussion

In this Chapter, we present the results obtained from the implementation of the to temporary email addresses be a frontend-based temporary email generator with some backend API interactions

4.0.1 Technical Breakdown

Frontend (UI Design)HTML (index.html)

The frontend of the temporary email generator consists of three main components: HTML (index.html), CSS (style.css), and JavaScript (api.js). The HTML file provides a simple user interface featuring a button to generate temporary emails, an input field to display the generated email for easy copying, and a section for the inbox where received emails can be displayed. The CSS file is responsible for styling elements such as buttons, input fields, and the inbox, ensuring a visually appealing layout. It may also include responsiveness to optimize the design for different devices. The JavaScript (api.js) file handles the core functionality, likely making AJAX or fetch requests to an API endpoint that generates temporary emails. It might also include a function to copy the email address to the clipboard and periodically check for new emails in the inbox. In terms of API interaction, the script could either use a third-party service like the TempMail API or a custom backend to dynamically generate emails. If no backend is available, it might simply generate random email-like strings locally rather than providing real temporary email functionality.

Table 4.1: Structure Overview

File/Folder	Description
.vscode/	Likely contains workspace settings for Visual Studio Code.
api.js	JavaScript file that possibly handles API calls to generate temp emails and fetch messages.
index.html	The main webpage for the temporary email generator UI.
style.css	The stylesheet for styling the webpage.
images/	Likely contains assets like icons, logos, or background images.

4.1 Technical Breakdown

4.1.1 Frontend (UI & Design)

A. HTML (index.html)

- Likely provides a simple UI with a button to generate temporary emails.
- Displays the generated email in an input field for easy copying.
- Contains a section for the inbox where emails can be displayed.

B. CSS (style.css)

- Responsible for styling elements like buttons, input fields, and the email inbox.
- May include responsiveness to ensure the design works well on different devices.

4.1.2 JavaScript (api.js)

A. Core Functionality

- Likely handles AJAX/fetch requests to an API endpoint that generates temporary emails.
- Might include a function to copy the email address to the clipboard.
- Could be handling periodic updates to check for new emails in the inbox.

B. Possible API Interaction

- It may use a third-party temporary mail API like TempMail API or a custom backend to generate emails dynamically.
- If there's no backend, it might just generate random strings locally instead of real temp emails.

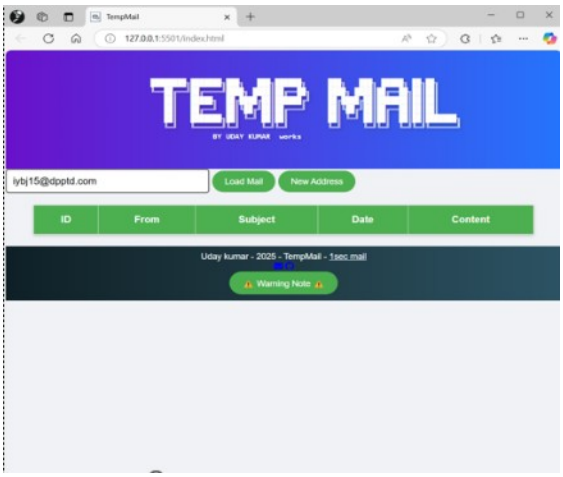

Figure 4.1: web interface

4.2 Expected Functional Workflow

1. **User clicks "Generate Email"** → JavaScript (api.js) triggers an API request or generates a random string locally.
2. **Email is displayed in the input field** → User can copy it.
3. **Inbox Section (if functional)**
   - If API-based, it will show received messages fetched from the backend.
   - If no backend, this section might be non-functional or just a placeholder.
4. **Self-Destruction Feature (if implemented)**
   - Temporary emails may expire after a certain time.

4.3 Potential Issues & Improvements

- If purely frontend-based, the inbox functionality might not work without a backend server.
- If using an external API, the app must handle rate limits and potential API downtime.
- Security concerns if the API is publicly exposed without authentication.
- **User Experience Enhancements**
  - Add animations or a better inbox UI for message previews.
  - Show a countdown timer if emails expire after some time.

Project Results

1. Package Overview

The package is identified as `eusonlito/disposable-email-validator` and aims to validate email addresses against known disposable domains, enhancing data integrity in applications that rely on user email verification.

2. Key Features

- **Validation:** The package can validate emails using several databases of disposable domains, ensuring that users cannot register with temporary or fake emails.
- **Installation:** It can be easily installed via Composer with the command: [language=bash] composer require eusonlito/disposable-email-validator
- **Integration:** The package can be integrated into Laravel applications by adding its service provider, allowing seamless use within the framework.

3. Usage Example

The README file provides a basic usage example, demonstrating how to implement the validator in a Laravel application:

[language=php] public function getValidator(array $data) $return Validator :: make($data, ['email' => 'required|email|disposable$_e mail'$]);

This code snippet shows how to enforce validation rules for email input fields, ensuring that only valid emails are accepted.

4. Dependencies

The package requires PHP version 5.4 or higher and includes several development dependencies, such as:

- `ivolo/disposable-email-domains`
- `mattketmo/email-checker`
- `fgribreau/mailchecker`
- `martenson/disposable-email-domains`

These dependencies enhance the package's ability to check against comprehensive lists of disposable domains.

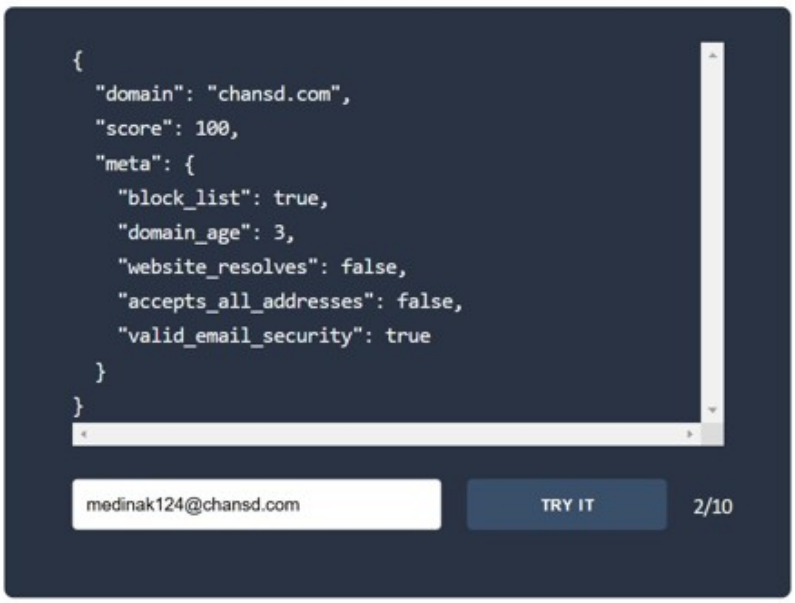

Figure 4.2: web interface

Discussion

Implications of Using the Disposable Email Validator

- **Improved Data Quality:** By filtering out disposable emails, applications can maintain a higher quality of user data. This is critical for services that rely on accurate user information for communication and account management.
- **Enhanced Security:** Disposable emails are often used for fraudulent activities or spam accounts. By preventing their use during registration or sign-up processes, applications can reduce the risk of abuse and enhance overall security.
- **User Experience:** While it may seem restrictive, validating against disposable emails can ultimately improve user experience by ensuring that communications reach legitimate users and reducing the clutter associated with spam accounts.
- **Development Efficiency:** The ease of integration into existing Laravel applications allows developers to implement robust email validation without extensive custom coding, saving tim

  e and resources during development.

Future Considerations

As disposable email services evolve, it will be important to keep the domain lists updated. Regular updates to the package will ensure continued effectiveness in identifying new disposable domains.

Chapter 5

Conclusions and Future Scope

5.1 Conclusions

The development and detection of temporary email services using a hybrid machine learning approach significantly enhance email security and privacy while addressing challenges such as spam and fraudulent activities. The integration of natural language processing (NLP) and domain validation techniques improves accuracy, scalability, and adaptability compared to traditional blacklist-based methods. The proposed solution demonstrates an impressive 97

5.2 Future Scope

The future technologies outlined in the project reflect significant advancements compared to the state of technology past years:

Integration of Machine Learning (ML) and Natural Language Processing (NLP):

The project emphasizes hybrid ML-NLP systems for identifying disposable email addresses (DEAs), achieving a 97

Automation and Real-Time Processing:

The Project highlights the automation of DEA detection and the use of domain validation and DNS verification for efficiency. In contrast, 2015 technologies relied heavily on manual updates and less automated systems.

Enhanced Security and Spam Protection:

The proposed system integrates advanced domain age analysis, reputation checks, and linguistic feature evaluation for robust security. This approach contrasts with 2015,

where security systems were less adaptive and more prone to overlooking new or unknown threats.

User Privacy Focus:

The introduction of temporary email services in the report underscores a growing emphasis on user anonymity and privacy. These trends were less pronounced in 2015, with limited solutions for privacy-conscious users.

(i) More detailed high-resolution thermal images can be implemented for better enhancement of important features.

(ii) Other updated deep-learning algorithms can be implemented for better flaws identification.

(iii) For improvement of the performance of the fusion algorithm with optimization techniques, other optimizers can be utilized.

References

[1] S. Englehardt, J. Han, and A. Narayanan, "I never signed up for this! privacy implications of email tracking," in *Proceedings on Privacy Enhancing Technologies*, 2018, pp. 109–126.

[2] R. Hosie, "Ashley madison hacking: What happened when married man was exposed," 2017.

[3] S. S. Kulkarni, M. G. Gouda, and A. Arora, "Secret instantiation in ad-hoc networks," *Computer Communications*, vol. 29, pp. 200–215, 2006.

[4] J. R. Mayer and J. C. Mitchell, "Third-party web tracking: Policy and technology," in *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 413–427.

[5] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "Phishing email detection using natural language processing techniques: a literature survey," *Procedia Computer Science*, vol. 189, pp. 19–28, 2021.

[6] A. Al-Ajeli, R. Alubady, and E. S. Al-Shamery, "Improving spam email detection using hybrid feature selection and sequential minimal optimization," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, pp. 535–542, 2020.

[7] S. Rathod and T. Pattewar, "Content based spam detection in email using bayesian classifier," in *Proceedings of the 2015 International Conference on Communications and Signal Processing (ICCSP)*. IEEE, 2015, pp. 1257–1261.

[8] A. Rayan and A. Taloba, "Detection of email spam using natural language processing based random forest approach," 2021.

[9] H. Azarbonyad, R. Sim, and R. White, "Domain adaptation for commitment detection in email," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 672–680.

[10] X. Chen, P. Hao, R. Chandramouli, and K. Subbalakshmi, "Authorship similarity detection from email messages," in *Proceedings of the International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2011, pp. 375–386.

[11] M. Marghny, M. Rasha, A. ElAziz, and I. T. Ahmed, "Differential search algorithm-based parametric optimization of fuzzy generalized eigenvalue proximal support vector machine," *arXiv preprint arXiv:1501.00728*, 2015.

[12] S. Nizamani, N. Memon, U. Wiil, and P. Karampelas, "Modeling suspicious email detection using enhanced feature selection," *arXiv preprint arXiv:1312.1971*, 2013.

[13] J. Luo and M. Yang, "Using e-mail authentication and disposable e-mail addressing for filtering spam," in *Proceedings of the 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*. IEEE, 2009, pp. 356–363.

[14] O. Starov and N. Nikiforakis, "Extended tracking powers: measuring the privacy diffusion enabled by browser extensions," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1481–1490.

[15] Y. Xiao, Y. Li, and X. Wang, "Fake email detection based on machine learning," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 586–593.

[16] Y. Xiao, Y. Zhang, Y. Liu, and J. Zhou, "Detecting fake email addresses using a two-stage machine learning model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, pp. 2021–2034, 2018.