

EMAILTRACKER: AN INTELLIGENT ANALYTICAL SYSTEM TO ASSIST EMAIL EVENT TRACKING USING ARTIFICIAL INTELLIGENCE AND BIG DATA

Joyce Zheng¹ and Yu Sun²

¹Central High School, 1700 W Olney Ave, Philadelphia, PA 19141

²California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

Recent studies have shown an increasing demand for software that assists in social media applications, such as Gmail [1] [2]. This paper develops a software that utilizes a pixel image in order to assist Gmail users with the status of their email [3]. This intelligent analytical system can be used to tell whether an email has been opened or not. After conducting a qualitative evaluation of this approach, the results provided evidence of the system's usability and the reliability of the system to give accurate results and data.

KEYWORDS

Chrome Extension, AI, Machine Learning.

1. INTRODUCTION

The technology of instantaneous communication over the internet began with Ray Tomlinson's creation of the electronic mail, which we refer to as Gmail [4]. At the time, messages could only be left on the same computer, however while working for ARPANET, Tomlinson came up with the idea to create a medium on the ARPANET system that allowed users to send electronic messages across multiple computers [5]. However, as user activity increased, problems relating to the status of emails began to appear. This application utilizes an invisible pixel image to assist users in tracking and understanding the status of their sent emails. Each pixel image contains its individual tracking number, which triggers the sender's system, thus allowing them to precisely see whether the email has been opened or not. Not only does this application save time for both the recipient and the sender, it also reduces the rate of follow-up emails [6].

Most of the existing systems that have been proposed are paid applications that notify users on the status of their email and allow users to view the data and analytic of multiple emails at once. However, these proposals don't take into consideration the rate of consumers that are willing to pay for the application, which is rarely the case in most practices. According to a study of 100 million users, the results have shown that only about 5% of current app users spend money on in-app purchases. Their implementations are also limited in scale, with samples given for ... Other techniques, such as ... They ... Because ..., the method/algorithm used cannot be too sophisticated and often results in ... A second practical problem is that some users find it hard to understand...

The rest of the paper is organized as follows: Section 2 gives insights on the challenges that we met during the designing of the email tracking application; Section 3 introduces solutions to the challenges mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, followed by a comparison of other similar works in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. How to ensure accurate results despite Gmail's security system

One challenge while creating the application was being able to get accurate results in spite of Gmail's security system [7]. Gmail's security system scans through emails, including attachments and links, for malicious content. The system might give a false positive because Gmail's security system had scanned and loaded the pixel image. This interferes with the application's ability to give accurate insight into the status of the email.

2.2. Difficulties with email tracking for group emails and forwarded emails

Because this email tracker uses an individual pixel image to track an email, it is meant for one person. Consequently, the extension is unable to create a tracking link for an email sent to multiple recipients. Because the image tracking link is sent with an email, it only tracks the message that includes the pixel image. This means that forwarded emails are unable to be tracked. However, chrome extensions like MailTrack have used other codes in their system to allow them to track group emails [8].

2.3. Recipients might have tracking blockers and other extensions that might render the accuracy of the email tracking

Email tracking blockers like Ghostery and Smart Pixel can prevent the pixel image in the email from loading, making tracking emails sent to recipients who have these blockers impossible [9] [10]. Additionally, if a recipient previews an email, the pixel image might still load, which gives the system a false result.

3. SOLUTION

Email Tracker is a chrome extension based on a system of pixel images that notifies users on the status of their email and allows users to view the data and analytic of multiple emails at once. The system uses the recipient's email address as well as the subject in regards to the email in order to create an individual tracking link, which is extremely helpful in identifying and tracking certain emails. This tracking link generates a pixel image, which allows senders to precisely see whether the email has been opened by the intended recipient or not. When the receiving end opens the email containing the pixel image and the pixel image is loaded, the image sends a message back to the server indicating that the email has been opened by the recipient. As shown in the graph above, the sender would compose an email, by inserting the recipient's email, email subject, and the message. Then, the sender would click on the email tracker extension in the upper right corner, which opens a window that generates a unique tracking link. Before sending the email, the user would insert the link into the photo section of the email, which generates an invisible pixel image. When the recipient clicks on the email, the pixel image is loaded, which sends the message back to the main system, allowing the sender to see that the email has been opened.

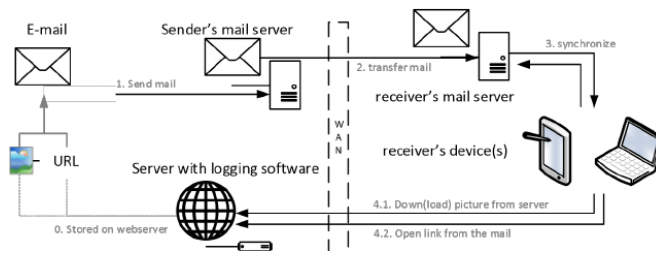


Figure 1. Overview of the system

The system uses the code to the right to create a window that creates the url for the pixel image file. The window utilizes the inputted subject and recipient to create the individual tracking link.

```
@app.route('/test')
def testImage():
    email = request.args.get("email")
    subj = request.args.get("subject")

    ip_list = request.access_route
    print(ip_list[0])
    temp = ip_list[0].split('.')
    my_ip = temp[0] + '.' + temp[1]

    if my_ip == '130.211':
        print("I opened the image!")
    else:
        currEmail.append([email, subj])

    return send_file("smallipixel.png", mimetype = "image/gif")
```

Figure 2. Screenshot of code 1

The system uses the code to the right in the background to request the data from the content's send message (such as the recipient name, recipient's email, and the subject) and send it back to the content.js to create the tracking link

The sender would then copy the tracking link and insert it into the photo section of the email and send it. Once the recipient opens the email, the pixel image will load and send a message back to the system to let them know that their email has been opened.

```
chrome.runtime.onMessage.addListener(
  function(request, sender, response){
    console.log(request);
    const arr = request.split("&")
    var d = JSON.stringify({"name": arr[0], "subj": arr[1]});
```

Figure 3. Screenshot of code 2

4. EXPERIMENT

4.1. Experiment 1

This experiment uses the results of 10 sent emails to show the accuracy of the system's ability of showing the status of an email (before the email is opened and after the email is opened). This experiment shows that despite the challenges faced, such as overcoming Google's security system, the email tracker still gives reliable results.

Table 1. Result of experiment 1

Email Trials	status before the email is opened	status after the email is opened
#1	unopened	seen
#2	unopened	seen
#3	unopened	seen
#4	unopened	seen
#5	unopened	seen
#6	unopened	unopened
#7	unopened	seen
#8	unopened	seen
#9	unopened	seen
#10	unopened	seen

Based on the results of the experiment, out of the 10 sent emails, almost all of them give accurate results. For the 6th trial run, after the email is opened, the email initially gives a status of unopened, but after reloading the page, it gives “seen”. Consequently, this proves that our system is reliable and capable of giving correct results.

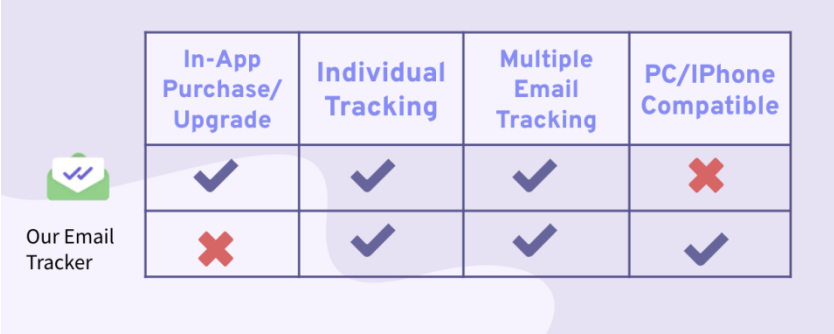
4.2. Experiment 2

This experiment uses Mailtrack and our system to compare and contrast the results and abilities of different email tracking systems by sending 10 emails with both systems (before the email is opened and after the email is opened).

Table 2. Result of experiment 2

Email Trials	Our System: status before email is opened	Our System: status after email is opened	Mailtrack: status before email is opened	Mailtrack: status after email is opened
#1	unopened	seen	not opened	opened
#2	unopened	seen	not opened	opened
#3	unopened	seen	not opened	opened
#4	unopened	seen	not opened	opened
#5	unopened	seen	not opened	opened
#6	unopened	unopened	not opened	opened
#7	unopened	seen	not opened	opened
#8	unopened	seen	not opened	opened
#9	unopened	seen	not opened	opened
#10	unopened	seen	not opened	opened

The results of the experiment shows that although both systems give accurate results, the abilities of the two email trackers vary. For example, our email tracker uses an invisible pixel image that tracks the email without the recipient knowing. However, Mailtrack includes a watermark in their emails, which is visible to both the sender and recipient. Additionally, Mailtrack requires the user to pay for upgraded services, however, our system does not.



	In-App Purchase/ Upgrade	Individual Tracking	Multiple Email Tracking	PC/iPhone Compatible
Our Email Tracker	✓	✓	✓	✗
	✗	✓	✓	✓

Figure 4. Vary results of Email Tracker

The experiment of the system 's accuracy proves that despite the challenges faced, such as overcoming Google 's security system, the email tracker still gives reliable results. The experiments also show that despite many other competitors in the email tracking field, our system has abilities and tools, such as not having to pay for upgraded versions, individual tracking, and anonymous tracking, that many other email tracking extensions don't have.

5. RELATED WORK

MailTrack is an email tracking chrome extension that utilizes pixel images to notify users on the status of their email and allow users to view the data and analytics of multiple emails at once [11]. While our system requires the user to copy and paste the given tracking link, MailTrack automatically tracks an email by pressing "compose". You can choose to not track the email by deleting the MailTrack watermark included in the email. However, this also means that the recipient can see that the email is tracked.

Mixmax is an email tracking software for Gmail that gives users email tracking alerts [12]. It also allows users to see which individual opened the email in cases with group emails. This feature is something not available on our software. Like our email tracking system, mixmax offers individual tracking. They also offer statistics and insights into how many times the email has been opened. However, the only disadvantage is that mixmax uses a watermark visible to both the recipient and sender.

Snovio is another email tracking extension that tracks emails using a pixel image system [13]. Like Mailtrack and Mixmax, it offers desktop push notification, something that our email tracker extension doesn't offer. It also allows users to see the full history and insights of an email. This includes the date and time they viewed the emails and the number of times the email is opened. However, based on some of the reviews, many users experience false results. For example, by opening the email you sent, it gives the system a false "seen".

6. CONCLUSIONS

Email Tracker is a free chrome extension that utilizes an invisible pixel image to help users track and understand the status of their email. The pixel images are devices of individual tracking links, which will trigger the sender's system when the images are loaded. The experiments shown above have verified the accuracy of the email tracking (in giving accurate results) and shown the email tracker in comparison to other email tracking devices. The experiment of the system's accuracy shows that out of 10 sent emails, 9 emails give accurate initial results. The results also show that in spite of Gmail's security system, which scans emails (and unintentionally loads the

pixel images) for malicious content, our system still gives accurate results. Additionally, in comparison to MailTrack, it has more pros, such as having individual and multiple tracking, iPhone and Android compatibility, and allows users to use the extension (and all of its tools) without an app purchase [14].

The experiment on the system's accuracy shows that out of 10 sent emails, 9 of them give initial accurate results. For the last email, after the email is opened, the email initially gave a status of unopened, but after reloading the page, it gave "seen". This shows that the user might have to reload the page a few times to get accurate results. Compared to other email tracking applications, our software requires a few extra steps to retrieve the tracking link, such as entering the email's subject and recipient to generate a tracking link.

In the future, I plan to fix the code in order to get the system to give accurate results immediately, without reloading the page [15]. Additionally, I plan to change the layout of the tracking software, by allowing users to press track on an email without having to physically generate a link.

REFERENCES

- [1] Carr, Caleb T., and Rebecca A. Hayes. "Social media: Defining, developing, and divining." *Atlantic journal of communication* 23.1 (2015): 46-65.
- [2] Chen, Mia Xu, et al. "Gmail smart compose: Real-time assisted writing." *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019.
- [3] Fisher, Peter. "The pixel: a snare and a delusion." *International Journal of Remote Sensing* 18.3 (1997): 679-685.
- [4] Sajithra, Karin, and Rajindra Patil. "Social media—history and components." *Journal of Business and Management* 7.1 (2013): 69-74.
- [5] Hauben, Michael. "History of ARPANET." *Site de l'Instituto Superior de Engenharia do Porto* 17 (2007).
- [6] MacFarlane, Andrew, et al. "Sender vs. recipient-orientated information systems revisited." *Journal of Documentation* (2021).
- [7] Dunning, Ted E. "Accurate methods for the statistics of surprise and coincidence." *Computational linguistics* 19.1 (1993): 61-74.
- [8] Liu, Lei, et al. "Chrome Extensions: Threat Analysis and Countermeasures." *NDSS*. 2012.
- [9] MacBeth, Sam. "Tracking the trackers: analysing the global tracking landscape with GhostRank." Retrieved October 3 (2017): 2018.
- [10] Hinton, H. Scott. "Progress in the smart pixel technologies." *IEEE journal of selected topics in quantum electronics* 2.1 (1996): 14-23.
- [11] Xu, Haitao, et al. "Privacy risk assessment on email tracking." *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018.
- [12] Savvidy, Konstantin G. "The MIXMAX random number generator." *Computer Physics Communications* 196 (2015): 161-165.
- [13] Cselle, Gabor, Keno Albrecht, and Rogert Wattenhofer. "BuzzTrack: topic detection and tracking in email." *Proceedings of the 12th international conference on Intelligent user interfaces*. 2007.
- [14] Remneland-Wikhamn, Björn, et al. "Open innovation, generativity and the supplier as peer: The case of iphone and android." *International Journal of Innovation Management* 15.01 (2011): 205-230.
- [15] Haupt, Johannes, et al. "Robust identification of email tracking: A machine learning approach." *European Journal of Operational Research* 271.1 (2018): 341-356.