



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 9, September 2021

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.569



Anti-Phishing Program (Spam Mail Detector)

Rutvij Mayank Dave¹, Asim Upendra Chitre², Shaunak Milind Alshi³, Heet Chheda⁴, Amey Gawde⁵

U.G. Student, Department of Electronics and Telecommunication Engineering, KJ Somaiya College of Engineering,

Vidyavihar, Maharashtra, India^{1,2,3,4}

Assistant Professor, Department of Electronics and Telecommunication Engineering, KJ Somaiya College of

Engineering, Vidyavihar, Maharashtra, India⁵

ABSTRACT: The majority of the emails we receive daily are spam and are unproductive. After hours of coding to remove spam emails from inboxes and store them in spam folders, spambots and spam accounts continue to find new ways to get into inboxes and do the tasks for which they were created. A spam email can be as easy as a message announcing the launch of a new product, or a message from someone asking for money to aid relatives who have been affected by a natural disaster, or even a threat from a particular group. To address these issues, we decided to develop a model that would prevent such emails from reaching our mailbox. To forecast whether the mail subject and body correspond to a spam or ham letter, we will use Keras, a deep learning framework. We looked at spam mail trends and utilised them as features in our model to assess if the mail we received was spam or not. Finally, we will demonstrate the correctness of our model and explain why it is a superior functional model for spam email filtering.

KEYWORDS: spam, deep learning, Keras

I. INTRODUCTION (P)

Junk email or unsolicited bulk emails sent through the email system are referred to as email spam. It refers to the use of an email system to send unsolicited emails to a set of recipients, particularly advertising emails. Unsolicited messages indicate that the receiver has not been permitted to receive them.

Email spam takes several forms, the most common of which is to advertise outright scams or shady business ventures. Spam is frequently used to market low-cost pharmaceuticals, weight-loss programmes, online degrees, job possibilities, and online gambling.

Spam may be a typical method of committing email fraud. A well-known example is an advance-fee scam, in which a user receives an email with an offer that purports to result in a reward. The fraudster provides a scenario during which the victim is required to supply immediate financial help so as for the fraudster to get away the bigger sum of cash, which they will subsequently share. The fraudster will either devise new fees or cease answering once the victim has made the money.

Phishing emails, which are emails posed as official communication from banks, online payment processors, or any other business a user may trust, are another type of fraudulent spam. Phishing emails usually drive recipients to a spoof version of the company's website, where they are asked to enter personal information such as login and credit card information.

II. RELATED WORK (P)

When an email is sent, it enters into the messaging system and is routed from one server to another till it reaches the recipient's mailbox. An email depends on certain protocols, such as SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol) and IMAP (Internet Message Access Protocol). The transmission details are specified by SMTP protocol whereas POP3 and IMAP are used to receive messages. A Message Agency (MTA) receives mails from a sender MUA or another MTA then determines the acceptable route for the mail. The recipients MTA delivers the incoming mail to the incoming mail server Mail Delivery Agent (MDA) which is essentially a POP/IMAP server. MUAs (e.g. Mozilla Thunderbird, Microsoft Outlook, etc.) are e-mail clients and help the user to read and write e-mails. Spam filters can be deployed at strategic places in both clients and servers. Many Internet Service Providers (ISPs) and organizations deploy spam filters at the e-mail server level, the well-liked places to deploy being at the gateways, mail routers, etc. They can be deployed in clients, where they will be installed at proxies or as plug-ins. Some spam filters can be deployed on both server and client-side.[1]



To effectively handle the threat posed by email spam, leading email providers like Gmail, Yahoo mail and Outlook have employed a mixture of various machine learning (ML) techniques such as Neural Networks in their spam filters. These ML techniques can find out and identify spam emails and phishing messages by analyzing a lot of such messages throughout a huge collection of computers. Since machine learning can adapt to varying conditions, Gmail and Yahoo mail spam filters do quite just checking junk emails using pre-existing rules. They generate new rules themselves based on what they have learnt as they continue in their spam filtering operation. The machine learning model employed by Google has now advanced to the purpose that it can detect and filter spam and phishing emails with about 99.9 per cent accuracy. The implication of this is often that one out of thousand messages achieve evading their email spam filter. Statistics from Google revealed that between 50-70% of emails that Gmail receives are spam. Google's detection models have also incorporated tools called Google Safe Browsing for identifying websites that have malicious URLs. The phishing-detection performance of Google has away been enhanced by the introduction of a system that delays the delivery of some Gmail messages for a while to carry out additional comprehensive scrutiny of the phishing messages since they are easier to detect when they are analyzed collectively. The purpose of delaying the delivery of a number of these suspicious emails is to conduct a deeper examination while more messages arrive in due course of your time and therefore the algorithms are updated in real-time. Only about 0.05 % of emails are affected by this deliberate delay.[2]

We have seen different types of spam messages being sent on our mail. It can be text related spam mail, image related spam mail or even video related spam mail. In our project, we have focused on text related spam mails by identifying the language of the sender, some stop-words and even some disturbing words from the subject as well as body to distinguish between a spam mail and a ham mail.

III. METHODOLOGY

The first thing which pops up is the login page (as shown in fig 5.1.1) which is made using the PyQt5 library where you would be able to write the email credentials, that is email id and password. It is made to protect the user's privacy while entering their credentials.

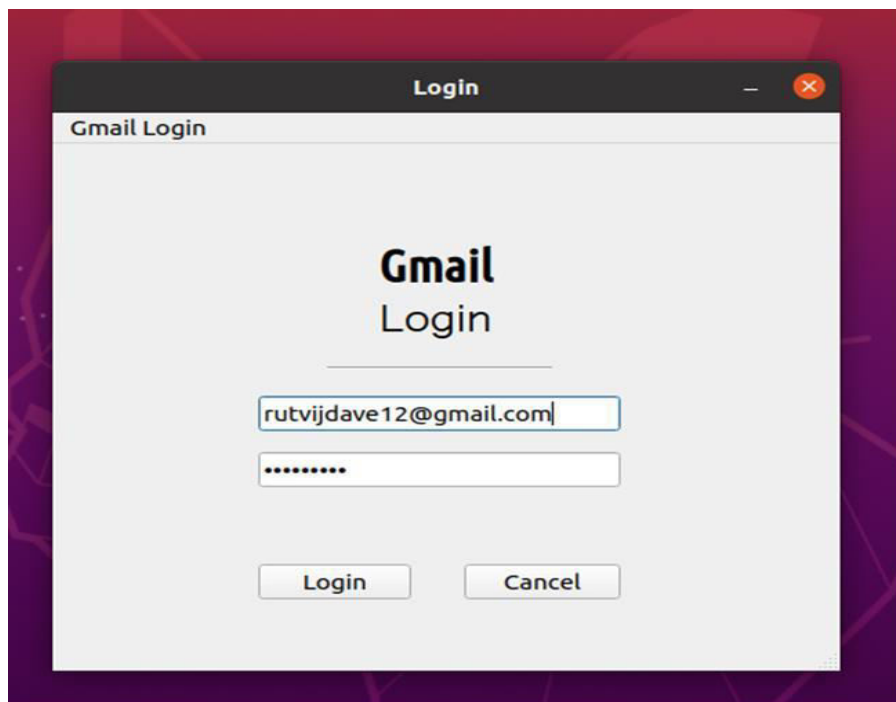
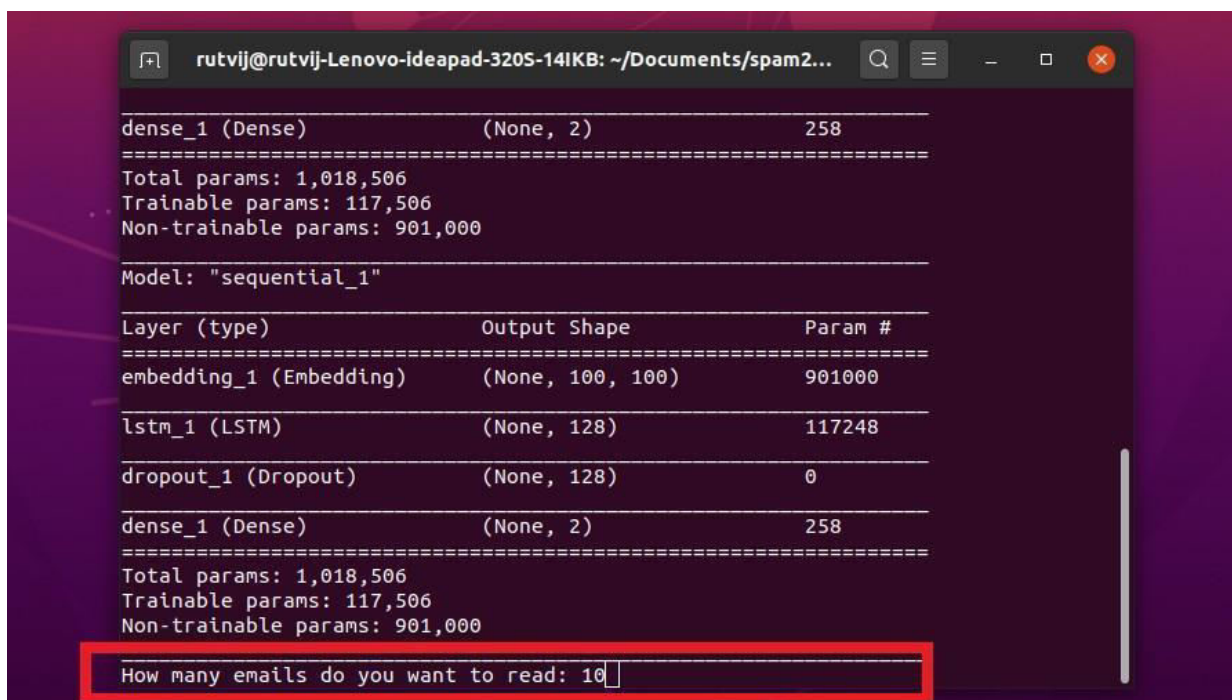


Fig 1. Login page with email and password added

The next step is a prompt you see on the command line asking "How many emails do you want to read:". The user enters an integer value (10 in this case) as in fig 2



```

rutvij@rutvij-Lenovo-ideapad-320S-14IKB: ~/Documents/spam2...
dense_1 (Dense)          (None, 2)          258
=====
Total params: 1,018,506
Trainable params: 117,506
Non-trainable params: 901,000
=====
Model: "sequential_1"
Layer (type)              Output Shape        Param #
=====
embedding_1 (Embedding)   (None, 100, 100)    901000
lstm_1 (LSTM)              (None, 128)         117248
dropout_1 (Dropout)       (None, 128)         0
dense_1 (Dense)           (None, 2)          258
=====
Total params: 1,018,506
Trainable params: 117,506
Non-trainable params: 901,000
=====
How many emails do you want to read: 10
  
```

Fig 2. Command prompt showing how an effective way of several of emails are read

After this step, 10 newest mails will be scraped from your email account and the contents will be void of all the HTML tags. All these data will be stored in an excel file (.csv format) named "my_emails_unfiltered.csv" which will contain the sender's email id, subject of the mail and body of the mail and a spam column which is labelled as "0" (ham) by default. (fig 5.1.3)

	A	B	C	D	E
1		Sender	Subject	Body	Spam
2	0	asimchitre@gmail.com	Football	back positive Football European pandemic nations fight sign	0
3	1	racle-acct_ww@oracle.co	Oracle Password Account Reset	ACCOUNT Privacy request Submit 2020 HELP Thank Account us Team email	0
4	2	racle-acct_ww@oracle.co	Oracle Password Account Reset	ACCOUNT Privacy request link used 2020 message HELP Thank Account Tea	0
5	3	no-reply@swiggy.in		Email Unsubscribe Dont delivered inbox want convenience Template	0
6	4	digest-noreply@quora.com		things big least like systems ob hired share Adam First question taking first Q	0
7	5	news@edx.org	getting ahead Online career learning	ence ahead Lean motivated Management learner Digital Sigma Decision Agi	0
8	6	rutvij.dave@somaiya.edu	Hi	Hey 20	0
9	7	change@e.change.org	Thank	least Affairs build restored week orking workers chain safeTwinkle ment hor	0
10	8	no-reply@swiggy.in		Email Unsubscribe Dont delivered inbox want convenience Template	0
11	9	noreply@login.ubuntu.com	Finish Ubuntu registration One	log following start moment address link used Thank right mistake us incorre	0
12					
13					
14					
15					
16					

Fig 3. CSV of unfiltered emails

The model is then imported from the results folder, and the messages in the CSV file (my emails unfiltered.csv) are now given through the model, with the projected output (whether the message is spam or not) visible in the “Spam” column of another CSV file, “my emails filtered.csv”) (fig 5.1.4).

	A	B	C	D	E	F
1		Unnamed: Sender		Subject	Body	Spam
2	0	0 asimchitre@gmail.com		Football	back positive Football European pandemic nations fight sign	ham
3	1	1 racle-acct_vw@oracle.co		Oracle Password Account Reset	ACCOUNT Privacy request Submit 2020 HELP Thank Account us Team email	ham spam
4	2	2 racle-acct_vw@oracle.co		Oracle Password Account Reset	ACCOUNT Privacy request link used 2020 message HELP Thank Account Team	ex: spam
5	3	3 no-reply@swiggy.in			Email Unsubscribe Dont delivered inbox want convenience Template	ham
6	4	4 digest-noreply@quora.com			things big least like systems ob hired share Adam First question taking first Questi	spam
7	5	5 news@edx.org		getting ahead Online career learning	ence ahead Lean motivated Management learner Digital Sigma Decision Agili	W: spam
8	6	6 rutvij.dave@somaiya.edu		Hi	Hey 20	ham
9	7	7 change@e.change.org		Thank	least Affairs build restored week orking workers chain safeTwinkle ment home	st spam
10	8	8 no-reply@swiggy.in			Email Unsubscribe Dont delivered inbox want convenience Template	ham
11	9	9 noreply@login.ubuntu.com		Finish Ubuntu registration One	log following start moment address link used Thank right mistake us incorrecly	U: spam
12						
13						

away

Fig 4. CSV effective of filtered emails

IV. EXPERIMENTAL RESULTS

We discovered (as shown in fig. 6.1) that out of a total of 5000+ emails that were checked as spam or ham:

1. The accuracy (percentage of right predictions) was 98.85%.
2. The accuracy (the percentage of spam emails that were genuinely spam) was 99 %.
3. The recall rate (the percentage of spam emails properly predicted) was 99.82 %.

```

mails-R.csv  reduce.py  spam_classifier.py x  test.py  predict.py  get_prediction.py  email  ...
spam_classifier.py > ...
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
2: Python
2624/4180 [=====>.....] - ETA: 3s - loss: 0.0081 - accuracy: 0.9970 - precision: 0.9983 - recall: 0.
2688/4180 [=====>.....] - ETA: 3s - loss: 0.0079 - accuracy: 0.9970 - precision: 0.9982 - recall: 0.
2752/4180 [=====>.....] - ETA: 3s - loss: 0.0082 - accuracy: 0.9967 - precision: 0.9982 - recall: 0.
2816/4180 [=====>.....] - ETA: 3s - loss: 0.0081 - accuracy: 0.9968 - precision: 0.9982 - recall: 0.
2880/4180 [=====>.....] - ETA: 3s - loss: 0.0079 - accuracy: 0.9969 - precision: 0.9982 - recall: 0.
2944/4180 [=====>.....] - ETA: 2s - loss: 0.0080 - accuracy: 0.9969 - precision: 0.9982 - recall: 0.
3008/4180 [=====>.....] - ETA: 2s - loss: 0.0079 - accuracy: 0.9970 - precision: 0.9981 - recall: 0.
3072/4180 [=====>.....] - ETA: 2s - loss: 0.0079 - accuracy: 0.9971 - precision: 0.9981 - recall: 0.
3136/4180 [=====>.....] - ETA: 2s - loss: 0.0078 - accuracy: 0.9971 - precision: 0.9981 - recall: 0.
3200/4180 [=====>.....] - ETA: 2s - loss: 0.0076 - accuracy: 0.9972 - precision: 0.9981 - recall: 0.
3264/4180 [=====>.....] - ETA: 2s - loss: 0.0076 - accuracy: 0.9972 - precision: 0.9981 - recall: 0.
3328/4180 [=====>.....] - ETA: 1s - loss: 0.0076 - accuracy: 0.9973 - precision: 0.9981 - recall: 0.
3392/4180 [=====>.....] - ETA: 1s - loss: 0.0075 - accuracy: 0.9973 - precision: 0.9981 - recall: 0.
3456/4180 [=====>.....] - ETA: 1s - loss: 0.0074 - accuracy: 0.9974 - precision: 0.9981 - recall: 0.
3520/4180 [=====>.....] - ETA: 1s - loss: 0.0073 - accuracy: 0.9974 - precision: 0.9981 - recall: 0.
3584/4180 [=====>.....] - ETA: 1s - loss: 0.0071 - accuracy: 0.9975 - precision: 0.9980 - recall: 0.
3648/4180 [=====>.....] - ETA: 1s - loss: 0.0070 - accuracy: 0.9975 - precision: 0.9980 - recall: 0.
3712/4180 [=====>.....] - ETA: 1s - loss: 0.0069 - accuracy: 0.9976 - precision: 0.9980 - recall: 0.
3776/4180 [=====>.....] - ETA: 0s - loss: 0.0068 - accuracy: 0.9976 - precision: 0.9980 - recall: 0.
3840/4180 [=====>.....] - ETA: 0s - loss: 0.0069 - accuracy: 0.9974 - precision: 0.9980 - recall: 0.
3904/4180 [=====>.....] - ETA: 0s - loss: 0.0068 - accuracy: 0.9974 - precision: 0.9980 - recall: 0.
3968/4180 [=====>.....] - ETA: 0s - loss: 0.0067 - accuracy: 0.9975 - precision: 0.9980 - recall: 0.
4032/4180 [=====>.....] - ETA: 0s - loss: 0.0066 - accuracy: 0.9975 - precision: 0.9980 - recall: 0.
4096/4180 [=====>.....] - ETA: 0s - loss: 0.0066 - accuracy: 0.9976 - precision: 0.9980 - recall: 0.
4160/4180 [=====>.....] - ETA: 0s - loss: 0.0067 - accuracy: 0.9976 - precision: 0.9980 - recall: 0.
4180/4180 [=====>.....] - 11s 3ms/step - loss: 0.0067 - accuracy: 0.9976 - precision: 0.9980 - recall: 0.9985
l: 0.9985 - val loss: 0.0790 - val accuracy: 0.9885 - val precision: 0.9896 - val recall: 0.9982

Epoch 00020: val_loss did not improve from 0.05854
1394/1394 [=====] - 1s 953us/step
[+] Accuracy: 98.85%
[+] Precision: 99.00%
[+] Recall: 99.82%
(base) rutvij@rutvij-Lenovo-ideapad-320S-14IKB:~/Documents/spam2/Spam-Detector/prediction$

```

Fig 5. Result Parameters

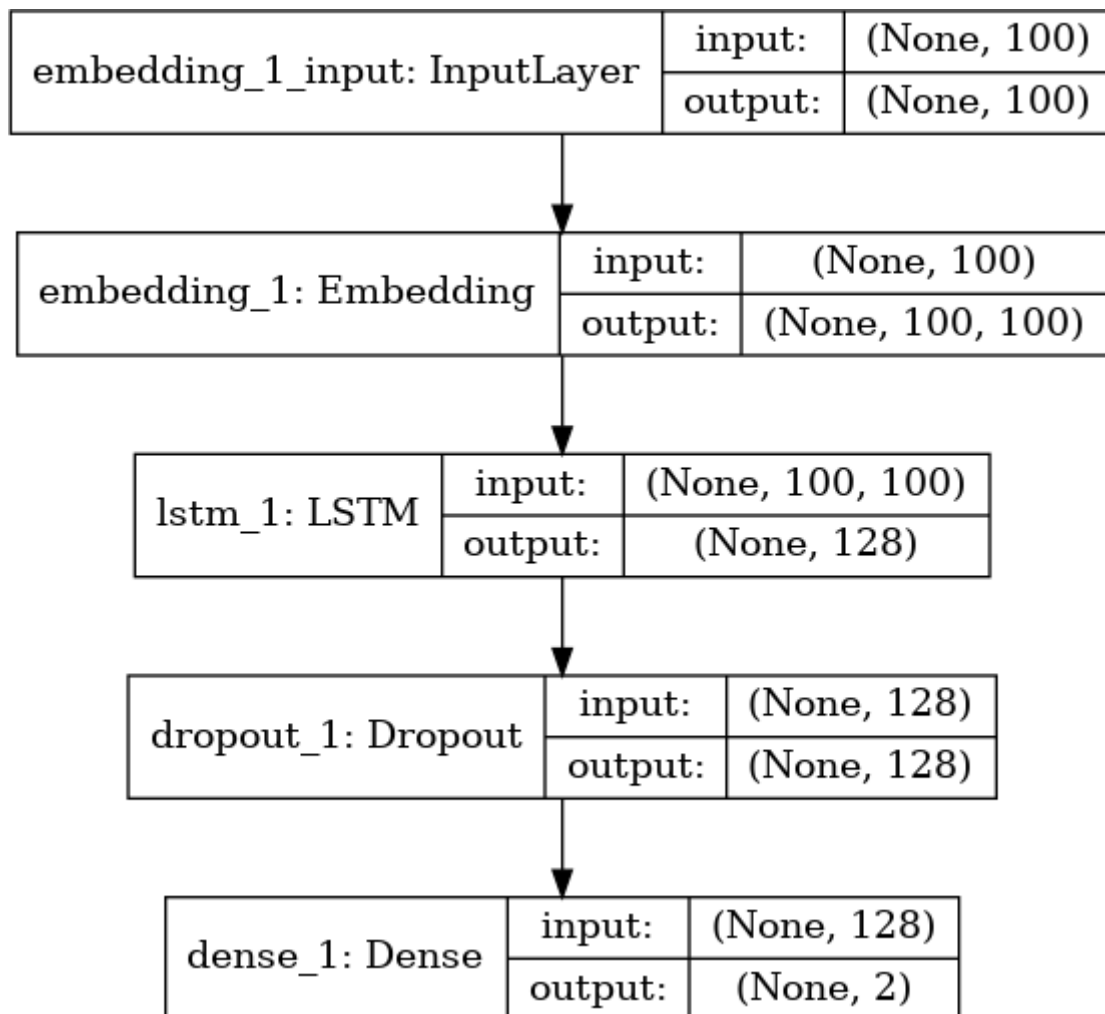


Fig 6. Neural Network graph of our model

V. CONCLUSION

Future research must address the reality that e-mail spam filtering is a co-evolutionary problem, with spammers attempting to outperform the classifiers while the filter attempts to improve its forecast accuracy. As a result, a good strategy must find a viable technique for detecting spam feature drift or evolution. Among all the traditional tactics we've seen so far, content-based spam filtering is the one that has had the most effective way of combating spam. Fortunately, machine learning-based solutions allow systems to learn and adapt to new threats, as well as spammers' countermeasures.

Thus, we were successful in implementing a model that will determine whether the emails we get are spam or ham mails using python.

REFERENCES

- [1] Alexy Bhowmick · Shyamanta M. Hazarika et al. Machine Learning for E-mail Spam filtering: Review, Techniques and Trends: 3 June 2016
- [2] Machine learning for email spam filtering: review, approaches and open research problems



**Impact Factor:
7.569**



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details