OPTIMIZING SPAM FILTERING WITH MACHINE LEARNING

Submitted by

m.kalaivani

BSc.,(CS)

1. INTRODUCTION

Over recent years, as the popularity of mobile phone devices has increased, Short Message Service (SMS) has grown into a multi-billion dollar industry. At the same time, reduction in the cost of messaging services has resulted in growth in unsolicited commercial advertisements (spams) being sent to mobile phones. Due to Spam SMS, Mobile service providers suffer from some sort of financial problems as well as it reduces calling time for users. Unfortunately, if the user accesses such Spam SMS they may face the problem of virus or malware. When SMS arrives at mobile it will disturb mobile user privacy and concentration. It may lead to frustration for the user. So Spam SMS is one of the major issues in the wireless communication world and it grows day by day.

1.1 OVERVIEW

A business requirement for an SMS spam classification system would include the ability to accurately identify and flag spam messages, protect customers from unwanted or harmful messages, and comply with industry regulations and laws regarding spam messaging. Additionally, the system should be able to handle a high volume of messages, integrate with existing systems and databases, and provide reporting and analysis capabilities to track performance and improve the system over time. The system should also have an easy-to-use interface and be easy to maintain and update.

1.2 PURPOSE

Social Impact:- it can help protect individuals from unwanted and potentially harmful messages. Spam messages can include phishing attempts, scams, and fraud, which can have serious financial and personal consequences for recipients. By accurately identifying and flagging spam messages, the system can help prevent these types of attacks and protect individuals from falling victim to them.

Business Model/Impact:- it can help protect their customers and improve their reputation. Spam messages can harm a business's reputation and lead to customer complaints and lost business. By accurately identifying and flagging spam messages, the system can help protect businesses and improve their customer's trust.

2.LITERATURE SURVEY

Globally, short messaging service (SMS) is one of the most popular and also most affordable telecommunication service packages. However, mobile users have become increasingly concerned regarding the security of their client confidentiality. This is mainly due to the fact that mobile marketing remains intrusive to the personal freedom of the subscribers [1]. SMS spamming has become a major nuisance to the mobile subscribers given its pervasive nature. It incurs substantial cost in terms of lost productivity, network bandwidth usage, management, and raid of personal privacy [2]. Thus, in short spamming threatens the profits of the service providers [3], [4]. Mobile SMS spams frustrate the mobile phone users, and just like e-mail spams, they cause new societal frictions to mobile handset devices [5]. Email spam is sent or received via the World Wide Web, while the SMS mobile spam is typically broadcasted via a mobile network.

Spam can be described as unwanted or unsolicited electronic messages sent in bulk to a group of recipients. The messages are characterized as electronic, unsolicited, commercial, mass constitutes a growing threat mainly due to the following factors: 1) the availability of low-cost bulk SMS plans; 2) reliability (since the message reaches the mobile phone user); 3) low chance of receiving responses from some unsuspecting receivers; and 4) the message can be personalized. Mobile SMS spam detection and prevention is not a trivial matter. It has taken on a lot of issues and solutions inherited from relatively older scenarios of email spam detection and filtering [8]. Unsolicited SMS text messages are a common occurrence in our daily life and consume communication time, bandwidth and resources. Although the existing spam filters provide some level of performance, the spams misinform receivers by manoeuvring data samples [9].

3. THEORITICAL ANALYSIS

A business requirement for an SMS spam classification system would include the ability to accurately identify and flag spam messages, protect customers from unwanted or harmful messages, and comply with industry regulations and laws regarding spam messaging. Additionally, the system should be able to handle a high volume of messages, integrate with existing systems and databases, and provide reporting and analysis capabilities to track performance and improve the system over time. The system should also have an easy-to-use interface and be easy to maintain and update.

3.1 HARDWARE / SOFTWARE DESINING

The hardware required for the development of this project is:

Processor : Intel core TM i5-9300H
Processor speed : 2.4GHz
RAM size :8GB DDR
System Type :x64- based processor

Software desining

The software required for the development of this project is:

Desktop GUI : Anaconda Navigator
Operating system : Windows 10
Front end : HTML,CSS,JAVASCRIPT
Programming : PYTHON
Cloud computing : IBM cloud services

4. EXPRIMENTAL INVESTIGATION

IMPORTING LIBRARIES

First step is usuallyimporting the libraries that will be needed in the program.

Pandas: It is a python library mainly used for data manipulation.

Numpy: This python library is used for numerical analysis.

Matplotlib and Seaborn: Both are the data visualization library used for plotting graph which will help us for understanding the data.

NLTK : a standard python library with prebuilt functions and utilities for the ease of use and implementation. It is one of the most used libraries for natural language processing and computational linguistics.

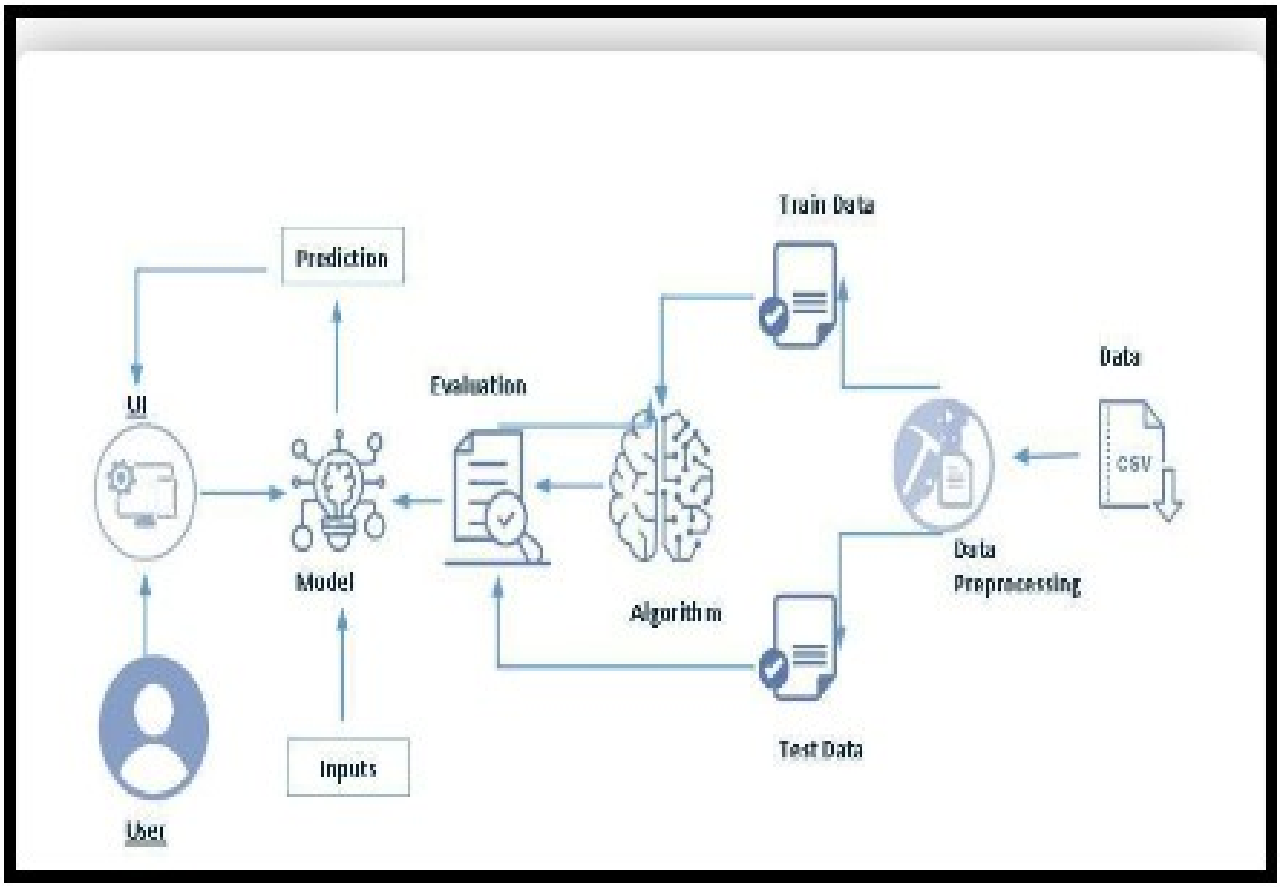
READING THE DATASET

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of the csv file.

1. Import the required packages.
2. Loading the Dataset.
3. Remove the unwanted data columns.
4. Preprocessing and Exploring the Dataset.
5. Build word cloud to see which message is spam and which is not.
6. Remove the stop words and punctuations.
7. Convert the text data into vectors.

5. FLOW CHART


Train Data
Prediction
Data
UI
Evaluation
CSV
Data
Preprocessing
Model
Algorithm
Inputs
Test Data
User

PROJECT FLOW

- User interacts with the Ul(User interface) to upload the input features.
- Entered input is analysed by the model which is integrated.
- Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

- Define Problem / Problem Understanding

- Specify the business problem
- Business requirement
- Literature Survey
- Social or Business requirements

- Exploratory Data

- Descriptive statistical
- Visual Analysis

- Model Building

  - Training the model in multiple algorithms
  - Testing the model

- Performance Testing and Hyperparameter Tuning

  - Tesing model with multiple evaluation metrics
  - Comparing model accuracy before and after applying hyperparameter tuning

- Model Deployment

  - Save the best model
  - Integrate with Web Framework

- Project Demonstration and Documentation

  - Record explanation Video for project end to end solution
  - Project development procedure documentation-Step by step project

6. RESULT

Out[2]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

Out[4]:

```
v1              0
v2              0
Unnamed: 2    5522
Unnamed: 3    5560
Unnamed: 4    5566
dtype: int64
```

Out[6]:

	lable	text	Unnamed: 2	Unnamed: 3	Unnamed: 4
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

Out[9]:
```
1114    No no:)this is kallis home ground.amla home to...
3589    I am in escape theatre now. . Going to watch K...
3095    We walked from my moms. Right on stagwood pass...
1012       I dunno they close oredi not... ÌÏ v ma fan...
3320                                Yo im right by yo work
                               ...
4931                Match started.india  &lt;#&gt;  for 2
3264    44 7732584351, Do you want a New Nokia 3510i c...
1653    I was at bugis juz now wat... But now i'm walk...
2607    :-) yeah! Lol. Luckily i didn't have a starrin...
2732    How dare you stupid. I wont tell anything to y...
Name: text, Length: 4457, dtype: object
```

```
Before OverSampling, counts of lable '1': 581
Before OverSampling, counts of lable '0': 3876
```

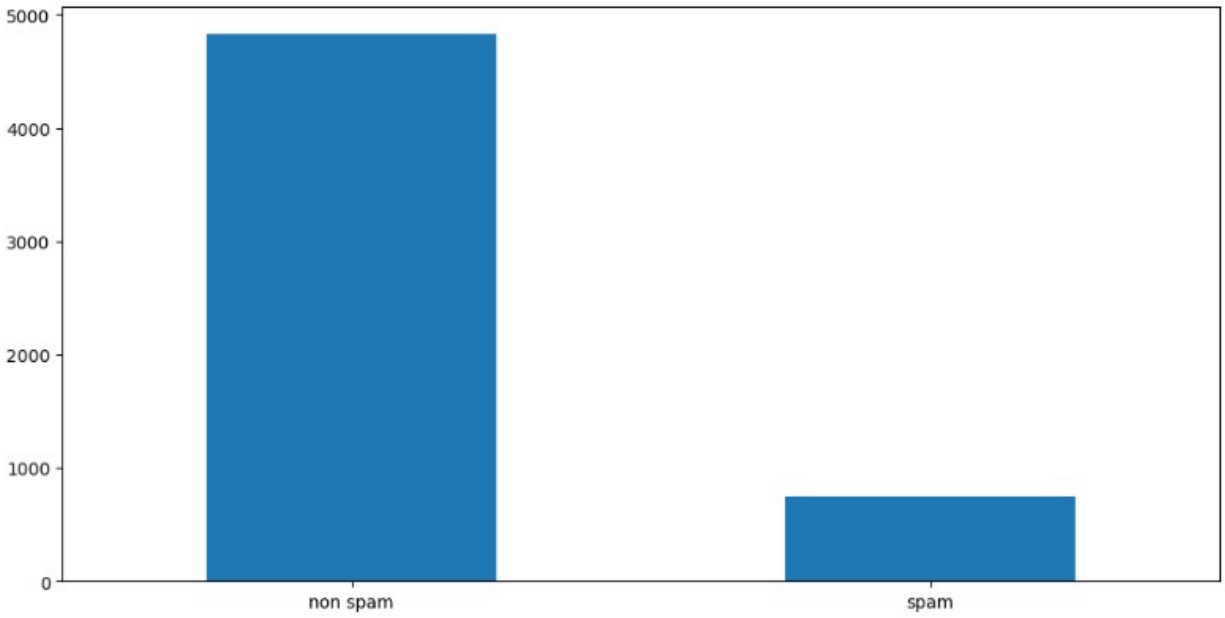
Out[17]:
```
True
```

Out[21]:
```
['go jurong point crazi avail bugi n great world la e buffet cine got amor wat',
 'ok lar joke wif u oni',
 'free entri 2 wkli comp win fa cup final tkt 21st may 2005 text fa 87121 receiv entri question std
txt rate c appli 08452810075over18',
 'u dun say earli hor u c alreadi say',
 'nah think goe usf live around though',
 'freemsg hey darl 3 week word back like fun still tb ok xxx std chg send 1 50 rcv',
 'even brother like speak treat like aid patent',
 'per request mell mell oru minnaminungint nurungu vettam set callertun caller press 9 copi friend
callertun',
 'winner valu network custom select receivea 900 prize reward claim call 09061701461 claim code kl3
41 valid 12 hour',
 'mobil 11 month u r entitl updat latest colour mobil camera free call mobil updat co free 08002986
030',
 'gonna home soon want talk stuff anymor tonight k cri enough today',
 'six chanc win cash 100 20 000 pound txt csh11 send 87575 cost 150p day 6day 16 tsandc appli repli
hl 4 info',
 'urgent 1 week free membership 100 000 prize jackpot txt word claim 81010 c www dbuk net lccltd po
box 4403ldnw1a7rw18',
```

Out[24]:

	lable
count	5572.000000
mean	0.134063
std	0.340751
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

Out[25]: (5572, 5)



Out[29]:

DecisionTreeClassifier

DecisionTreeClassifier()

Out[30]:

DecisionTreeClassifier

DecisionTreeClassifier()

Out[31]:

```
▾ RandomForestClassifier
RandomForestClassifier()
```

Out[32]:

```
▾ RandomForestClassifier
RandomForestClassifier()
```

Out[34]:

```
▾ MultinomialNB
MultinomialNB()
```

Out[35]: sklearn.naive_bayes.MultinomialNB

Out[41]: (4457, 7163)

```
Epoch 1/10
69/69 [==============================] - 67s 895ms/step - loss: 0.1576 - accuracy: 0.9554
Epoch 2/10
69/69 [==============================] - 58s 844ms/step - loss: 0.0066 - accuracy: 0.9982
Epoch 3/10
69/69 [==============================] - 58s 838ms/step - loss: 0.0012 - accuracy: 0.9998
Epoch 4/10
69/69 [==============================] - 58s 836ms/step - loss: 3.2633e-04 - accuracy: 1.0000
Epoch 5/10
69/69 [==============================] - 58s 843ms/step - loss: 9.9362e-05 - accuracy: 1.0000
Epoch 6/10
69/69 [==============================] - 58s 848ms/step - loss: 6.4709e-05 - accuracy: 1.0000
Epoch 7/10
69/69 [==============================] - 59s 862ms/step - loss: 4.3521e-05 - accuracy: 1.0000
Epoch 8/10
69/69 [==============================] - 59s 855ms/step - loss: 3.1069e-05 - accuracy: 1.0000
Epoch 9/10
69/69 [==============================] - 59s 860ms/step - loss: 2.3374e-05 - accuracy: 1.0000
Epoch 10/10
69/69 [==============================] - 59s 857ms/step - loss: 1.8360e-05 - accuracy: 1.0000
```

```
35/35 [==============================] - 6s 33ms/step
```

Out[48]:

```
array([[5.8226783e-12],
       [2.1026846e-05],
       [1.4567069e-12],
       ...,
       [1.0731280e-07],
       [2.5747346e-13],
       [9.0983861e-12]], dtype=float32)
```

```
Out[50]: 4456    0
         690     0
         944     0
         3768    0
         1189    0
                ..
         2906    0
         1270    0
         3944    0
         2124    0
         253     0
         Name: lable, Length: 1115, dtype: int32


[[948   1]
 [ 13 153]]
Accuracy Score Is:-  98.7443946188341


 Enter new review...hello fg hou hkl
 [[0 0 0 ... 0 0 0]]
 1/1 [==============================] - 4s 4s/step
 [[0.00268606]]


 [[948   1]
  [ 13 153]]
 Accuracy Score Is Naive Bayes:-  98.7443946188341


 [[948   1]
  [ 13 153]]
 Accuracy Score Is:-  98.7443946188341
```

```
 * Serving Flask app '__main__'
 * Debug mode: off
```

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
[2023-04-21 12:41:32,537] ERROR in app: Exception on / [GET]
Traceback (most recent call last):
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\app.py", line 2525, in wsgi_app
    response = self.full_dispatch_request()
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\app.py", line 1822, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\app.py", line 1820, in full_dispatch_request
    rv = self.dispatch_request()
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\app.py", line 1796, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
  File "C:\Users\GASCCS23\AppData\Local\Temp\ipykernel_1416\2211322710.py", line 14, in home
    return render_template('home.html')
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\templating.py", line 146, in render_template
    template = app.jinja_env.get_or_select_template(template_name_or_list)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\jinja2\environment.py", line 1081, in get_or_select_template
    return self.get_template(template_name_or_list, parent, globals)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\jinja2\environment.py", line 1010, in get_template
    return self._load_template(name, globals)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\jinja2\environment.py", line 969, in _load_template
    template = self.loader.load(self, name, self.make_globals(globals))
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\jinja2\loaders.py", line 126, in load
    source, filename, uptodate = self.get_source(environment, name)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\templating.py", line 62, in get_source
    return self._get_source_fast(environment, template)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\templating.py", line 98, in _get_source_fast
    raise TemplateNotFound(template)
jinja2.exceptions.TemplateNotFound: home.html
127.0.0.1 - - [21/Apr/2023 12:41:33] "GET / HTTP/1.1" 500 -
127.0.0.1 - - [21/Apr/2023 12:41:33] "GET /favicon.ico HTTP/1.1" 404 -
```

7. ADVANTAGES AND DISADVANTAGES

Advantages

- ✓ the security risk can be reduced since the user gets in hand the emails that have gone through various spam checks.
- ✓ these email spam filters throw out malware, malicious, and virus-infested emails and protect user security.

Disadvantages

- ✓ Thousands of spam emails may reach Inboxes before a spammer's email address, IP or domain is blacklisted.
- ✓ In a case of using large blocks of legitimate text

8. APPLICATIONS

- o Preparing the text data.
- o Creating word dictionary.
- o Feature extraction process
- o Training the classifier

9. CONCULATION

Spam emails have become a major concern for the internet community as it poses a threat to integrity and productivity of the users. Filtering of email is very much necessary for email communication. The accurate detection of spam emails is a big issue and many filtering methods have been proposed by variousresearchers.

. After analyzing different papers given by different researchers, we observed as follows SVM algorithm was not able to give better result in

- terms of accuracy. Naïve Bayes has better performance than other

• algorithms such as Support Vector Machines (SVM's) and DecisionTrees. Decision Tree Classifier was taking large memory
• space which is a great matter of concern for the researchers. Size of dataset used by many authors is very small
• and needs to be expanded. Most of the proposed model has four basic steps,
• preprocessing of data, feature extraction, training and testing. Some models used pattern matching technique also
• for classification by using regular expression. Spammers are now evolving and sending spam
• emails containing pictures and pdf to pass the filter.

10.FUTURESCOPE

Once the dictionary is ready, we can extract word count vector (our feature here) of 3000 dimensions for each email of training set.
Each **word count vector** contains the frequency of 3000 words in the training file. Of course you might have guessed by now that most of them will be zero. Let us take an example. Suppose we have 500 words in our dictionary. Each word count vector contains the frequency of 500 dictionary words in the training file. Suppose text in training file was "Get the work done, work done" then it will be encoded as [0,0,0,0,0,…….0,0,2,0,0,0,……,0,0,1,0,0,…0,0,1,0,0,……2,0,0,0,0,0]. Here, all the word counts are placed at 296th, 359th, 415th, 495th index of 500 length word count vector and the rest are zero.

APPENDIX

A Sources code of Flask:

```
from flask import Flask, render_template, request

import  pickle

import numpy as np

import re

import nltk

from nltk.corpus import stopwords

from ntlk.stem import PorterStemmer

from tensorflow.keras.models. import load_model
```

Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument.

```
Load_model = load_model('spam.h5)

Cv = pickle.load(open('cv1.pkl','rb'))

App = Flask(_name_)
```

REFERANCE

[1] S. K. Tuteja, “Classification Algorithms for Email Spam Filtering", 2016.

[2] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, "Email Classification Research Trends: Review and Open Issues", 2017.

[3] S. Ajaz, M. T. Nafis, and V. Sharma, "Spam Mail Detection Using Hybrid Secure Hash Based Naive Classifier, 2017.

[4] Shafi’i Muhammad Abdul Hamid, M. S., Osho, O., Ismaila, I., & Alhassan, J. K. “Comparative Analysis of Classification Algorithms for Email Spam Detection”, 2018.

[5] Rusland, N. F., Wahid, N., Kasim, S., & Hafit, H.. “AnalysisofNaïveBayes Algorithmfor Email Spam Filtering across Multiple Datasets”, 2017.

[6] Yüksel, A. S., Cankaya, S. F., & Üncü, İ. S. “Design of a Machine Learning Based Predictive Analytics System for Spam Problem”, 2017