[1] Sardarov Yagub Bali

[2] Nadir Bakhtiyar Rahimov

Implementation of a Spam Detector in Python for Corporate Companies

Journal of Electrical Systems

Abstract: - Many companies, while not categorizing spam as a form of attack, may still consider it a significant threat. When referring to spam, it represents an excessive volume of unwanted and unsolicited electronic mail sent to company email accounts [1]. Although it is often used for marketing purposes, in many cases, it is also employed by malicious actors for detrimental purposes against the company. For instance, spam can be used to propagate viruses or infiltrate corporate computers [1] [2]. Furthermore, spammers may send emails with the aim of directing users to phishing web pages, attempting to obtain confidential information from authorized personnel within the company. In general, spam is, at its simplest, an annoying form of mail, and in its worst cases, it serves as an enabler for spyware, viruses, intrusion, and phishing threats [3].

Keywords: Spam, protection methods, informational resources, unwanted email, malicious.

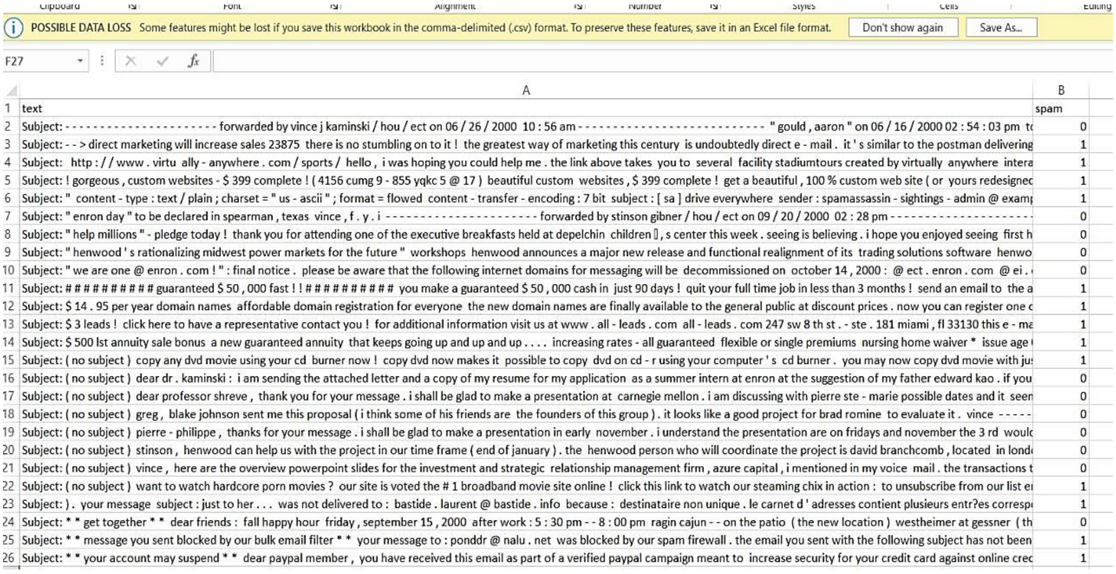
I. INTRODUCTION

Spam can be identified based on the following characteristics:

- Mass email distribution: In this case, electronic mail is directed towards multiple and diverse targets, rather than a specific individual.
- Anonymity: In this scenario, the true identity of the sender or senders is kept confidential.
- Unwanted email: The sent electronic messages are either unnecessary for the company or not anticipated by any entity within the organization [2] [3].

Now, let's consider the implementation of the system in the Python environment that identifies these harmful electronic emails sent to the company as spam.

Firstly, messages characterized as spam are presented as shown below. Their total count is 5728. This data is of particular importance for the system in terms of how it classifies messages as spam (Figure 1).



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

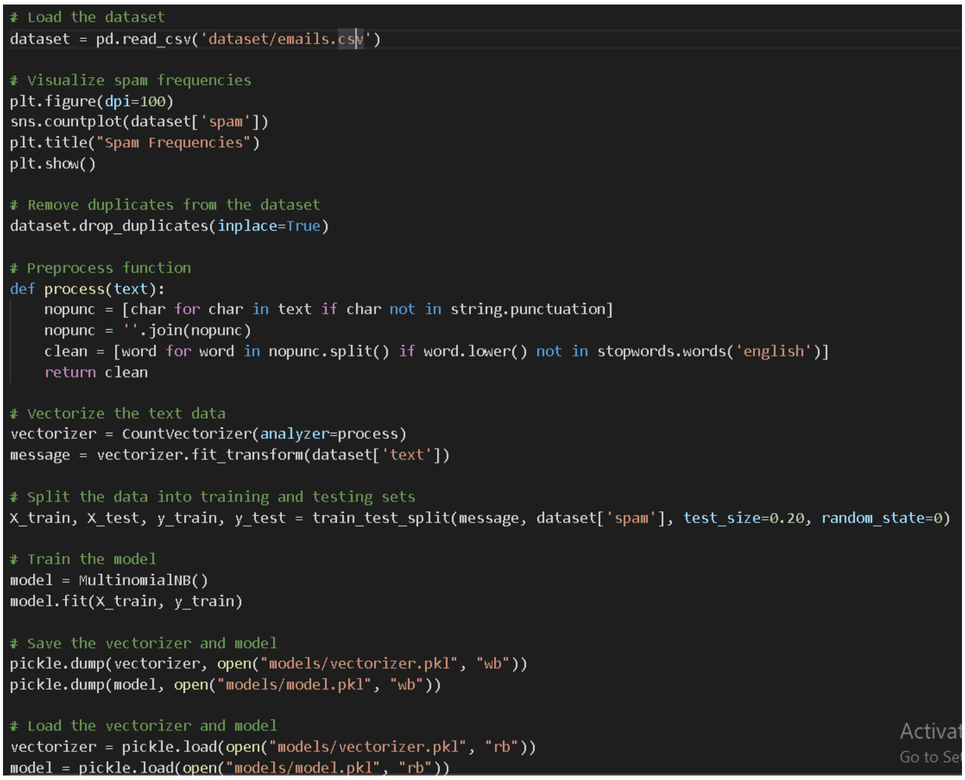
	A	B
1	text	spam
2	Subject: - forwarded by vince j kaminski / hou / ect on 06 / 26 / 2000 10 : 56 am - " gould , aaron " on 06 / 16 / 2000 02 : 54 : 03 pm t	0
3	Subject: - - > direct marketing will increase sales 23875 there is no stumbling on to it ! the greatest way of marketing this century is undoubtedly direct e - mail . it ' s similar to the postman delivering	1
4	Subject: http : / / www . virtu ally - anywhere . com / sports / hello , i was hoping you could help me . the link above takes you to several facility stadiumtours created by virtually anywhere intera	1
5	Subject: ! gorgeous , custom websites - $ 399 complete ! (4156 cumg 9 - 855 yqkc 5 @ 17) beautiful custom websites , $ 399 complete ! get a beautiful , 100 % custom web site (or yours redesignec	1
6	Subject: " content - type : text / plain ; charset = " us - ascii " ; format = flowed content - transfer - encoding : 7 bit subject : [sa] drive everywhere sender : spamassassin - sightings - admin @ examp	1
7	Subject: " enron day " to be declared in spearman , texas vince , f . y . i - forwarded by stinson gibner / hou / ect on 09 / 20 / 2000 02 : 28 pm -	0
8	Subject: " help millions " - pledge today ! thank you for attending one of the executive breakfasts held at depelchin children ▯ , s center this week . seeing is believing . i hope you enjoyed seeing first h	0
9	Subject: " henwood ' s rationalizing midwest power markets for the future " workshops henwood announces a major new release and functional realignment of its trading solutions software henwo	0
10	Subject: " we are one @ enron . com ! " : final notice . please be aware that the following internet domains for messaging will be decommissioned on october 14 , 2000 : @ ect . enron . com @ ei .	0
11	Subject: # # # # # # # # # # guaranteed $ 50 , 000 fast ! ! # # # # # # # # # # you make a guaranteed $ 50 , 000 cash in just 90 days ! quit your full time job in less than 3 months ! send an email to the a	1
12	Subject: $ 14 . 95 per year domain names affordable domain registration for everyone the new domain names are finally available to the general public at discount prices . now you can register one c	1
13	Subject: $ 3 leads ! click here to have a representative contact you ! for additional information visit us at www . all - leads . com all - leads . com 247 sw 8 th st . - ste . 181 miami , fl 33130 this e - ma	1
14	Subject: $ 500 lst annuity sale bonus a new guaranteed annuity that keeps going up and up and up . . . . increasing rates - all guaranteed flexible or single premiums nursing home waiver * issue age	1
15	Subject: (no subject) copy any dvd movie using your cd burner now ! copy dvd now makes it possible to copy dvd on cd - r using your computer ' s cd burner . you may now copy dvd movie with ju	1
16	Subject: (no subject) dear dr . kaminski : i am sending the attached letter and a copy of my resume for my application as a summer intern at enron at the suggestion of my father edward kao . if you	0
17	Subject: (no subject) dear professor shreve , thank you for your message . i shall be glad to make a presentation at carnegie mellon . i am discussing with pierre ste - marie possible dates and it seen	0
18	Subject: (no subject) greg , blake johnson sent me this proposal (i think some of his friends are the founders of this group) . it looks like a good project for brad romine to evaluate it . vince - - - - -	0
19	Subject: (no subject) pierre - philippe , thanks for your message . i shall be glad to make a presentation in early november . i understand the presentation are on fridays and november the 3 rd woulc	0
20	Subject: (no subject) stinson , henwood can help us with the project in our time frame (end of january) . the henwood person who will coordinate the project is david branchcomb , located in lond	0
21	Subject: (no subject) vince , here are the overview powerpoint slides for the investment and strategic relationship management firm , azure capital , i mentioned in my voice mail . the transactions t	0
22	Subject: (no subject) want to watch hardcore porn movies ? our site is voted the # 1 broadband movie site online ! click this link to watch our steaming chix in action : to unsubscribe from our list e	1
23	Subject:) . your message subject : just to her . . . was not delivered to : bastide . laurent @ bastide . info because : destinataire non unique . le carnet d ' adresses contient plusieurs entr?es corresp	1
24	Subject: * * get together * * dear friends : fall happy hour friday , september 15 , 2000 after work : 5 : 30 pm - - 8 : 00 pm ragin cajun - - on the patio (the new location) westheimer at gessner (th	0
25	Subject: * * message you sent blocked by our bulk email filter * * your message to : ponddr @ nalu . net was blocked by our spam firewall . the email you sent with the following subject has not been	1
26	Subject: * * your account may suspend * * dear paypal member , you have received this email as part of a verified paypal campaign meant to increase security for your credit card against online crec	1

Figure 1. Dataset

II. METHODS

As shown in the figure, these data are marked as 0 and 1. Where 0 denotes that the email is not spam, and 1 characterizes it as spam and unsafe. After instructing the system with this data, it can autonomously determine which emails are spam without human intervention [4]. The code used to create this system is presented in main.py (Figure 2).

[1, 2] dept. name: Computer engineering, Azerbaijan State Oil and Industry University, Baku, Azerbaijan



```
# Load the dataset
dataset = pd.read_csv('dataset/emails.csv')

# Visualize spam frequencies
plt.figure(dpi=100)
sns.countplot(dataset['spam'])
plt.title("Spam Frequencies")
plt.show()

# Remove duplicates from the dataset
dataset.drop_duplicates(inplace=True)

# Preprocess function
def process(text):
    nopunc = [char for char in text if char not in string.punctuation]
    nopunc = ''.join(nopunc)
    clean = [word for word in nopunc.split() if word.lower() not in stopwords.words('english')]
    return clean

# Vectorize the text data
vectorizer = CountVectorizer(analyzer=process)
message = vectorizer.fit_transform(dataset['text'])

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(message, dataset['spam'], test_size=0.20, random_state=0)

# Train the model
model = MultinomialNB()
model.fit(X_train, y_train)

# Save the vectorizer and model
pickle.dump(vectorizer, open("models/vectorizer.pkl", "wb"))
pickle.dump(model, open("models/model.pkl", "wb"))

# Load the vectorizer and model
vectorizer = pickle.load(open("models/vectorizer.pkl", "rb"))
model = pickle.load(open("models/model.pkl", "rb"))
```

Figure 2. Calling a dataset in Python, training a model

After importing the Python packages and libraries to be used, the dataset is incorporated into the project. Based on the existing datasets, visualization is performed. Visualization is a method that represents the quantity of spam within the data that the system has learned with clarity and numerical values [3] [4]. This is illustrated as follows (Figure 3).

The orange quantity represents the amount of spam, while the blue quantity indicates the count of regular emails. Since the computer environment recognizes words as numbers, this information is later converted into numbers through a vectorization process. All words within the emails are included here. Subsequently, within the transferred dataset, the system learns in the background how often each word is used in the email data [5].

After the execution of these processes, the converted data into numbers are divided into two parts: test data and training data. A necessary model is built using the training data. Once this model is established, it must be validated using the test data. It is important to note that the data used for testing should be previously unseen by the established model. Otherwise, the model will not function accurately. To prevent such a situation, test data is kept separate from the model [6].

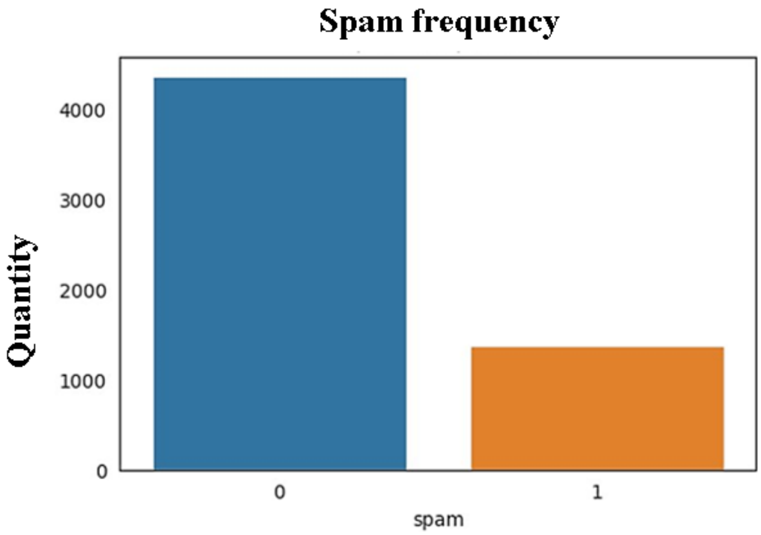
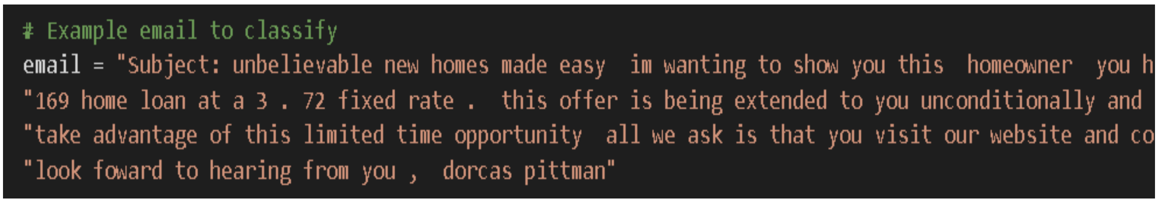


Figure 3. A visual representation of the amount of spam

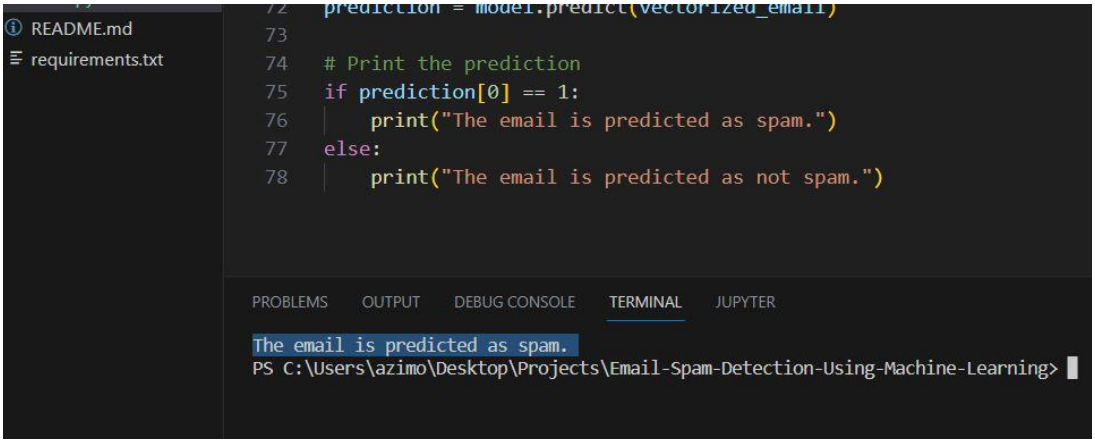
Once the model is created and configured, it is stored in memory. Subsequently, the same model is loaded. Then, we use the test data to evaluate the established model [6] [7]. An example of the code for checking spam in the model is shown as follows (Figure 4).



```
# Example email to classify
email = "Subject: unbelievable new homes made easy  im wanting to show you this  homeowner  you h
"169 home loan at a 3 . 72 fixed rate .  this offer is being extended to you unconditionally and
"take advantage of this limited time opportunity  all we ask is that you visit our website and co
"look foward to hearing from you ,  dorcas pittman"
```

Figure 4. Test data

The resulting output is displayed below (Figure 5). If the submitted data is not spam, then the output will indicate that the data is a safe message and not spam [7] [10].

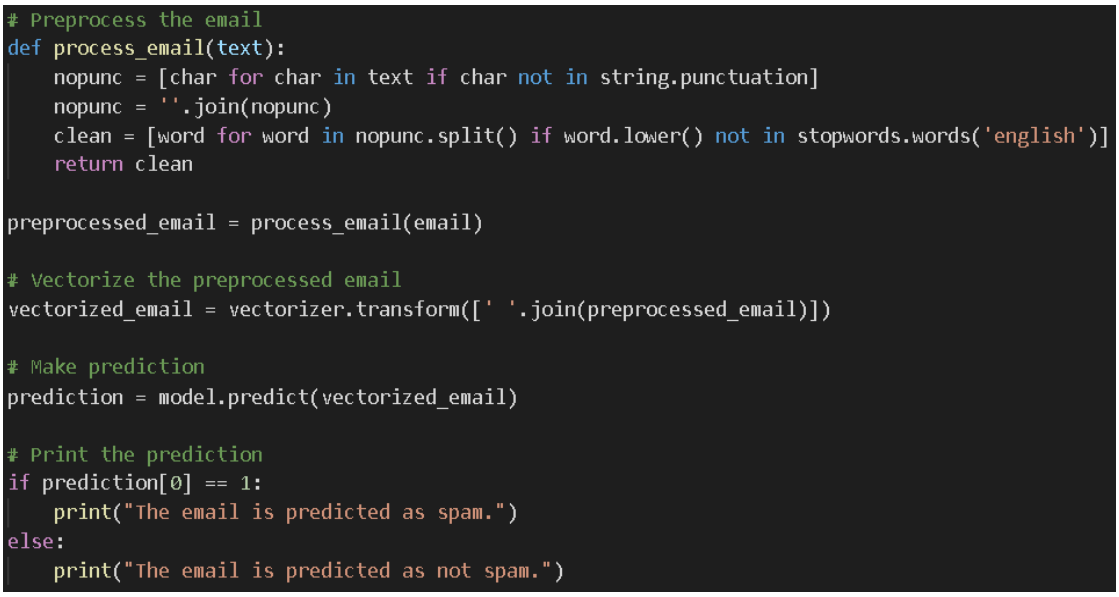


```
prediction = model.predict(vectorized_email)

# Print the prediction
if prediction[0] == 1:
    print("The email is predicted as spam.")
else:
    print("The email is predicted as not spam.")
```

```
The email is predicted as spam.
PS C:\Users\azimo\Desktop\Projects\Email-Spam-Detection-Using-Machine-Learning>
```

Figure 5. Spam email detection

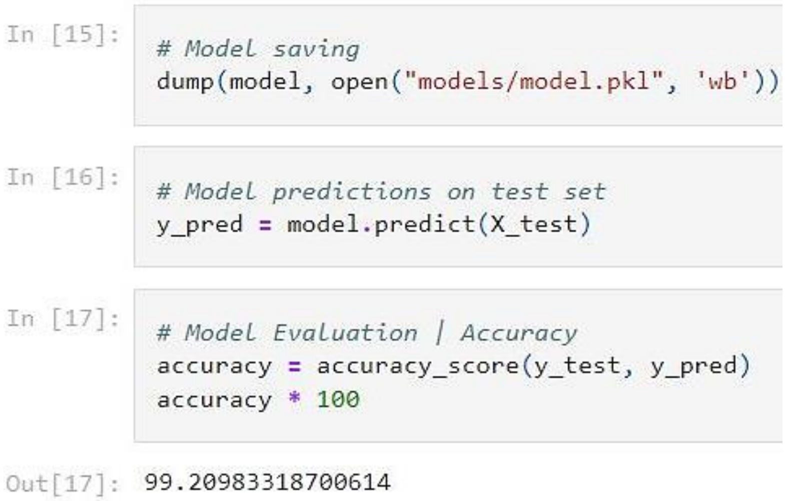
In the presented figures, in addition to the mentioned elements, the codes incorporating the prediction condition are also provided (Figure 6).



```
# Preprocess the email
def process_email(text):
    nopunc = [char for char in text if char not in string.punctuation]
    nopunc = ''.join(nopunc)
    clean = [word for word in nopunc.split() if word.lower() not in stopwords.words('english')]
    return clean

preprocessed_email = process_email(email)

# Vectorize the preprocessed email
vectorized_email = vectorizer.transform([' '.join(preprocessed_email)])

# Make prediction
prediction = model.predict(vectorized_email)

# Print the prediction
if prediction[0] == 1:
    print("The email is predicted as spam.")
else:
    print("The email is predicted as not spam.")
```

Figure 6. Data recognition through a vectorizer

After the model is constructed, it is essential to assess the extent to which it makes accurate and precise decisions. To achieve this, the following code example should be executed (Figure 7).

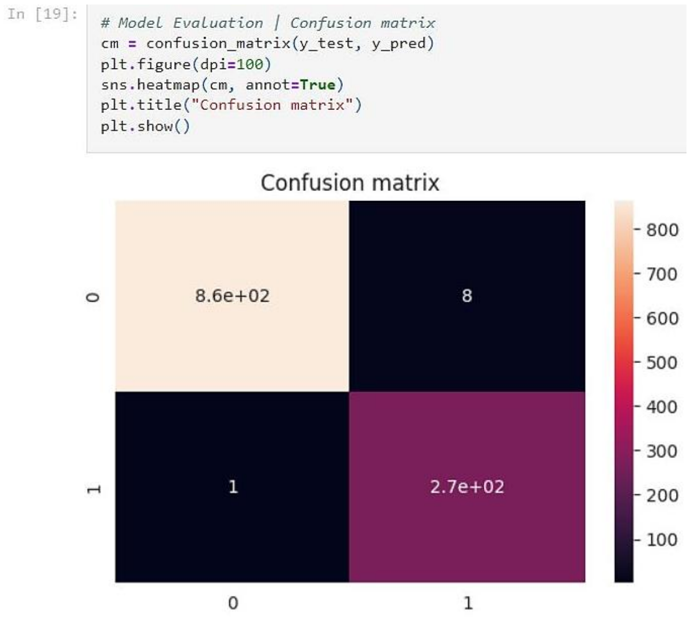


```
In [15]:  # Model saving
          dump(model, open("models/model.pkl", 'wb'))

In [16]:  # Model predictions on test set
          y_pred = model.predict(X_test)

In [17]:  # Model Evaluation | Accuracy
          accuracy = accuracy_score(y_test, y_pred)
          accuracy * 100

Out[17]:  99.20983318700614
```

Figure 7. Accuracy of the model

As evident from the figure, the accuracy metric of the created model is 99%. This implies that it can accurately classify 99 out of every 100 incoming emails.

Furthermore, it is possible to create a matrix that describes how accurately the model identified the normal and safe emails, how many spam emails were misclassified as non-spam, how many normal emails were identified as spam, and how many spam emails were correctly classified as spam among all the test data entered into this model [7] [8] [9]. The mentioned matrix is presented below (Figure 8).



```
In [19]:  # Model Evaluation | Confusion matrix
          cm = confusion_matrix(y_test, y_pred)
          plt.figure(dpi=100)
          sns.heatmap(cm, annot=True)
          plt.title("Confusion matrix")
          plt.show()
```

Figure 8. Model evaluation

III. RESULTS

The protection of informational resources in companies holds significance not only for potential users but also for the companies themselves. When it comes to safeguarding databases and information sources, it is not merely achieved through software solutions. Hardware infrastructure establishment and utilization should also be considered. Corporations and organizations that ensure complete security safeguard their reputations and stand out sharply from their active competitors.

Each security model possesses distinct characteristics, and consideration should be given to their component composition when utilizing these models. Furthermore, to protect the company's assets from potential computer threats, security measures, such as maintaining and regularly updating signatures, should be diligently upheld. In the Python environment, spam protection detectors can effectively mitigate the problems encountered by companies. These detectors facilitate a more dynamic and robust defense against relevant attacks, thereby enhancing the overall security of the company.

REFERENCES

[1] https://techvera.com/4-ways-to-protect-your-business-information-and-data/

[2] https://www.endpointprotector.com/blog/5-ways-big-companies-protect-their-data/

[3] https://www.fcc.gov/communications-business-opportunities/cybersecurity-small-businesses

[4] Upton, David M.; Osborn, William J. Harvard Business School Cases. “Companies should understand where cybercrime thrives”, Mar 01, 2022, p1-880. 880p.

[5] Dobrygowski, Daniel. Harvard Business Review Digital Articles. “Why companies are forming alliances?”, 9/11/2019, p2-5. 4p.

[6] Watson, Rachel. Grand Rapids Business Journal. “Cyberattacks are gaining momentum”, 5/2/2022, Vol. 40 Issue 9, p1-4. 4p.

[7] Glassman, James K. Kiplinger's Personal Finance. “An urgent need for cybersecurity”, Jun2022, Vol. 76 Issue 6, p39-40. 2p.

[8] Aeilts Tony. FBI Law Enforcement Bulletin. “Defending against cybercrime and terrorism: A new role for universities.” Volume: 74 Issue: 1 Dated: January 2005 Pages: 14-20

[9] Peralta, Paola. Employee Benefit News. “Protect your company data from the Great Resignation”, Mar/Apr2022, Vol. 36 Issue 2, pN.PAG-N.PAG. 1p.

[10] Shaker, Bob. CFO. “Do You Even Know What Cyber Defense Is?”, Jul/Aug2017, Vol. 33 Issue 6, p19-19. 2/3p.