Technische Hochschule Ingolstadt

Seminar zu Themen der Informatik

Sommersemester 2022

Seminararbeit

Analyse von PST-Dateien

von

Alexander Pilz

1 Einleitung

E-Mails sind im privaten wie auch im geschäftlichen Umfeld nicht mehr als Kommunikationsmedium wegzudenken. Dies ist klar erkennbar an den versandten und empfangenen E-Mails weltweit. So lag diese Zahl im Jahr 2021 bei 319,6 Milliarden E-Mails und steigt laut einer Prognose bis ins Jahr 2025 auf 376,4 Milliarden E-Mails [4].

E-Mails werden heutzutage aber nicht nur für die Kommunikation verwendet, sondern auch zum Versenden von Spamnachrichten. So beläuft sich der Anteil an Spam-Mails am weltweiten Anteil der versandten E-Mails auf 46 Prozent [2]. Unter Spam versteht man Nachrichten, die ohne Aufforderung und unterwünscht zugestellt werden. Meist haben diese Nachrichten den Zweck Werbung zu verbreiten. Jedoch gibt es auch weitaus bedenklichere Nachrichten. So wird mithilfe von Spam-Mails versucht, unvorsichtige Nutzer dazu zu bringen, persönliche Daten preiszugeben oder finanzielle Gewinne zu erzielen.

Im geschäftlichen Umfeld wird hier dem Nutzer meist die Arbeit abgenommen, weil professionelle Spam-Filter eingerichtet sind und so fast keine unerwünschten Nachrichten mehr ankommen. Im privaten Umfeld werden solche Spamfilter oft nicht bzw. nicht richtig angewandt. Hinzu kommt, dass sich viele Nutzer mit dieser Thematik gar nicht auseinandersetzen. Dies führt häufig dazu, dass die Postfächer der Nutzer hier regelrecht durch Spam-Mails überladen werden.

Im Rahmen dieser Seminararbeit sollen diese unerwünschten E-Mails mithilfe des .PST-Dateiformats analysiert und die Ergebnisse dokumentiert werden.

2 Allgemeines

3 PST-Datei

Das .pst-Dateiformat ist ein proprietäres Dateiformat, welches von Microsoft im Mailprogramm „Outlook“ verwendet wird. PST steht dabei für Personal Storage Table. Microsoft nutzt das Dateiformat zum Speichern von Nachrichtenkopien, Kalendereinträgen und Kontakten. Nutzt man den Microsoft Exchange Server werden die Daten an den Server übermittelt und dort gespeichert. Im Gegensatz dazu speichert Microsoft Outlook ohne Exchange Server diese Elemente auf dem lokalen Computer. Dabei

werden .pst-Dateien meist zum Speichern archivierter Elemente verwendet.

3.1 Aufbau einer PST-Datei

3.2 Sicherheitsprobleme mit PST-Dateien

PST-Dateien bieten den Vorteil, dass sie leicht übertragbar sind. Auch ohne fundierte IT-Kenntnisse können sie zwischen verschiedenen Outlook-Clients übertragen werden. Abgesehen von diesem Mobilitätsvorteil bringen sie jedoch auch Sicherheitsprobleme mit sich [1]:

- Beschädigungs- bzw. fehleranfällig, was zu Datenverlusten führen kann
- Stromausfälle, PC-Abstürze oder versehentliches Schließen trennen eine PST-Datei vom Outlook-Profil (abgekoppelte oder verwaiste PSTs entstehen, die für die IT-Abteilung nicht sichtbar sind, auch wenn sie wertvolle Informationen enthalten, die benötigt werden)
- Inhalte von PST-Dateien sind nur an der Quelle verfügbar und somit bei Analysen von zentralen Punkten aus eventuell nicht verfügbar
- Zugangsbeschränkungen oder eine falsche Klassifizierung können dazu führen, dass PST-Objekte falsch kategorisiert werden
- Passwortschutz vorhanden, kann aber sehr leicht entschlüsselt werden
- Native PST-Verschlüsselung bietet keinen ausreichenden Schutz

4 Schritte bei der Analyse der PST-Datei

In diesem Kapitel werden die durchgeführten Analyseschritte näher erläutert. Dabei wird auf eine Senderanalyse, eine zeitliche Analyse sowie eine Analyse bestimmter Schlagwörter eingegangen.

4.1 Datensatz

Zuerst wird näher auf den verwendeten Datensatz für die Analyse eingegangen. Dabei war die Auswahl eines geeigneten Datensatzes schwieriger als anfangs erwartet, da der Datensatz auch Kriterien für die Analyse erfüllen sollte. Der Datensatz sollte eine relativ hohe Anzahl an E-Mails enthalten, es sollte ein schwacher bzw. kein Spam-Filter vorhanden sein und der Datensatz sollte in ein .pst-Dateiformat überführt werden können. Eine Online-Suche nach geeigneten Datensätzen ergab eine Anzahl an Treffern, die E-Mail-Postfächer simulieren, diese eigneten sich aufgrund ihrer Formate jedoch nicht für die Weiterverarbeitung die in dieser Seminararbeit geplant war.

Aufgrund dessen wurde im privaten Umfeld nach verfügbaren Postfächern gesucht, die ich für die Analyse verwenden durfte. Dabei wurde mir ein E-Mail Konto vom Anbieter ”Web.deßur Verfügung gestellt, welches 4102 E-Mails enthält. Dieses Konto wurde dann in Microsoft Outlook eingebunden um mithilfe des Mail-Programmes einen .pst-Export erstellen zu lassen, der dann als Grundlage für die Analyse verwendet wurde.

4.2 Parsing der PST-Datei

Für die Analyse wurde die Python Bibliothek „libpff“ in Version 20211114 verwendet. Diese Bibliothek wurde speziell für den Zugriff auf das Personal Folder File (PFF) und das Offline Folder File (OFF)

```
pst = pypff.file()
pst.open("C:\\Users\\Alexp\\PycharmProjects\\pstfileanalyse\\PSTFiles\\LRKO.pst")
```

Abbildung 1: Python Code - Öffnen der .pst-Datei mithilfe von libpff

```
def parse_folder(base):
    messages = []
    countEmails = 0
    for folder in base.sub_folders:
        if folder.number_of_sub_folders:
            messages += parse_folder(folder)
        print(folder.name)
        for message in folder.sub_messages:
            countEmails += 1
            messages.append({
                "subject": message.subject,
                "sender": message.sender_name,
                "datetime": message.client_submit_time,
                "text": message.plain_text_body
            })
    print(countEmails)
    return messages
```

Abbildung 2: Python Code - Extraktion der Eigenschaften aus .pst-Datei

entwickelt. Diese Formate werden von Microsoft Outlook verwendet, um E-Mail, Kontakte und andere Daten zu speichern. PFF und OFF werden dabei in mehreren Dateitypen verwendet, unter anderem auch bei PST [3].

Zuerst wurde die PST-Datei mithilfe von des Moduls pypff der Bibliothek libpff geöffnet (siehe Abb. 1). Im nächsten Schritt habe ich eine rekursive Funktion mit dem Namen parse_folder definiert. Diese durchläuft die Ordner der .pst-Datei zählt dabei die enthaltenen E-Mails in den jeweiligen Ordnern, was in Abbildung 3 zu sehen ist, wo der Ordner Unbekannt 4102 E-Mails enthält. Des Weiteren erzeugt die Funktion eine Print-Ausgabe auf der Konsole (zu sehen in Abbildung 3), um den Aufbau der Ordnerstruktur des .pst-Files zu sehen. Die wichtigste Aufgabe der Funktion besteht jedoch darin, eine Liste mit den gewünschten Eigenschaften der E-Mails zu erstellen. In diesem Fall wurden von mir die Eigenschaften „subject“, „sender“, „datetime“ und „text“ ausgewählt. Der Aufbau dieser Funktion sowie die darin aufgeführten Eigenschaften der E-Mail sind in Abbildung 2 abgebildet. Nach der Funktionsdefinition wird die Funktion wie in Abbildung 4 zu sehen ist, aufgerufen. Die somit entstehende Liste „messages“ wird anschließend mithilfe der Bibliothek pandas in einen DataFrame umgewandelt und dann in eine .CSV-Datei mit dem Namen „MessagesPD.csv“ exportiert um für die nachfolgende Weiterverarbeitung in einem geeigneterem Format bereitzuliegen. Abschließend ist hierbei noch zu erwähnen dass die hier definierte Funktion relativ langsam ist und bei größeren Datensätzen eventuell optimiert werden müsste.

```
SPAM Search Folder 2
Gelöschte Elemente
Papierkorb
Posteingang
Postausgang
Gesendet
Lokale Fehler (Nur dieser Computer)
0
Synchronisierungsprobleme (Nur dieser Computer)
Entwurf
Spam
Junk-E-Mail
Unbekannt
4101
Oberste Ebene der Outlook-Datendatei
Suchpfad
0

Process finished with exit code 0
```

Abbildung 3: Print Ausgabe - Aufbau der .pst-Datei

```
messages = parse_folder(root)

df = pd.DataFrame(messages)
df.to_csv('MessagesPD.csv', index=False)
```

Abbildung 4: Python Code - Export in .CSV Datei

```
with open(file, 'r', encoding='UTF8') as in_file:
    for row in csv.reader(in_file):
        if not row:
            test = 0
        else:
            if row[0] in emails:
                index = emails.index(row[0])
                emails[index + 1] = emails[index + 1] + 1
            else:
                tmp_word = row[0]
                emails.append(tmp_word)
                emails.append(1)

with open('FromCount.csv', mode='w', encoding='UTF8', newline='') as out_file:
    header = ['Sender', 'Received Mails']

    writer = csv.writer(out_file)  # create the writer
    writer.writerow(header)  # write the header rows
    length = int(len(emails) / 2)
    for i in range(0, length):
        writer.writerow([emails[(i * 2)], emails[(i * 2) + 1]])
```

Abbildung 5: Python Code - Zählen der erhaltenen E-Mails und auflisten nach Absender

4.3 Senderanalyse

Eine Analyse der Sender wurde durchgeführt, um festzustellen, wie viele verschiedene Absender einer E-Mail an das Analysierte Postfach gesendet haben und welche davon am häufigsten vorhanden waren.

4.4 Zeitliche Analyse

Die nächste Analyse bezieht sich auf das zeitliche Eintreffen von Emails. Durch das Parsen der .pst-Datei lag die Eigenschaft „datetime“, welche das Datum und die Zeit der empfangenen E-Mail darstellt in dem Format vor, dass in Abbildung 7 zu sehen ist. Mithilfe des Python Codes aus Abbildung 8 werden die Daten in die korrekte Zeitzone konvertiert, weil sie standardmäßig im UTC-Format gespeichert werden. Hierzu wird die Zeitzone erst als UTC deklariert und dann in die gewünschte Zeitzone umgewandelt. Im nachfolgenden Schritt werden die extrahierten Daten als Punktewolke, welche die Ankunftszeiten der E-Mails nach Datum darstellt, geplottet. Dazu werden zwei Spalten mit den Koordinaten der zu verfolgenden Punkte erstellt. Als letzter Schritt wird dann die Punktewolke erstellt. Dazu wurden die Python Bibliotheken „matplotlib“ und „seaborn“ verwendet. In dieser Seminararbeit wurde der vorhandene Datensatz vom Jahr 2020 bis Mitte 2022 analysiert. Daraus entstand der Graph, der in Abbildung 9 abgebildet wird. Klar erkennbar ist hier die Häufung von empfangenen E-Mails zwischen 16 und 18 Uhr. Ebenso ist ab dem Jahr 2022 eine Häufung der Nachrichten im Bereich um 14 bis 15 Uhr erkennbar. Auffällig ist, das zwischen 22 Uhr und 5 Uhr fast keine E-Mails eingetroffen sind. Daraus lässt sich ableiten, dass tatsächlich fast nur Mails gesendet werden, zu Zeiten bei denen der Nutzer auch selbst aktiv ist.

4.5 Analyse von Spam-Wörtern

In diesem Kapitel wird auf die Analyse bestimmter Schlagwörter in den E-Mails eingegangen.

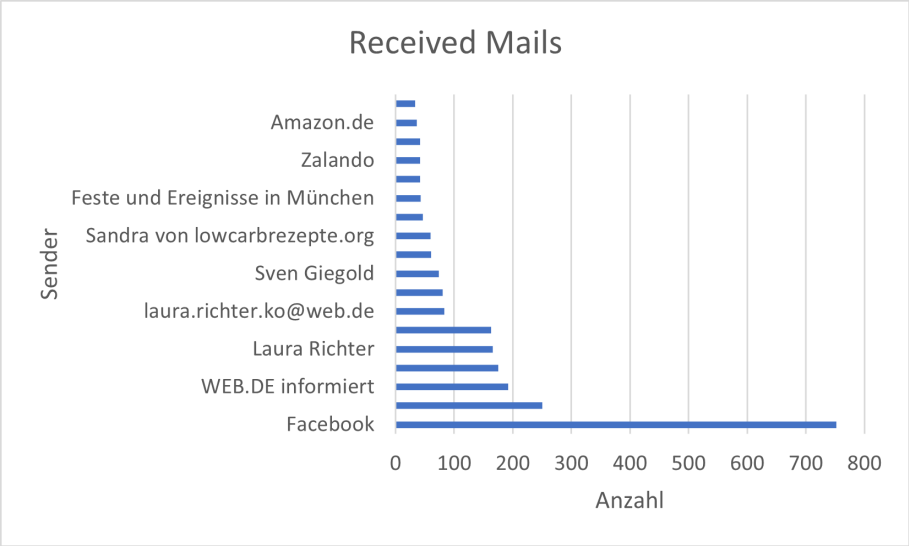

Abbildung 6: Absender mit mehr als 100 gesendeten Mails

5 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Analyse näher erläutert.

6 Fazit

In dieser Seminararbeit wurde eine forensische Analyse einer .pst-Datei durchgeführt. Hierfür wurden verschiedene Analysen durchgeführt. Die Analysen wurde mithilfe von eigens geschriebenen Python-Skripten durchgeführt, da Tools für die Analyse von .pst-Dateien nur kostenpflichtig zur Verfügung standen bzw. bei den kostenfreien Versionen nur sehr eingeschränkte Funktionalitäten besessen haben. Durch die Verwendung professioneller Tools wären mit großer Wahrscheinlichkeit noch viel detailliertere Ergebnisse zustande gekommen.

Zum Parsing der .pst-Datei wurde die Python Bibliothek ”libpff”verwendet, die sich speziell für den Zugriff auf Dateiformate von Microsoft Outlook eignet. Mithilfe dieser Bibliothek wurden dann die wichtigsten Eigenschaften der E-Mails extrahiert, um sie für die weitere Analyse aufzubereiten. Im Anschluss daran wurden dann eine Senderanalyse, eine zeitliche Analyse und eine Spamwortanalyse durchgeführt.

Die Senderanalyse hat gezeigt, dass Facebook mit 752 E-Mails auf Platz 1 der meisten gesendeten E-Mails liegt. Dies liegt höchstwahrscheinlich daran, dass der Besitzer des Postfaches seine Facebook Benachrichtigungen aktiviert hat. Auf dem zweiten Platz liegt jedoch ”Lidl Insider”mit 250 E-Mails. Dieses Ergebnis ist erschreckend, da ich nach einer Rücksprache mit dem Besitzer des E-Mail Kontos erfahren habe, dass genau eine Bestellung vor einigen Jahren durchgeführt wurde und seitdem zahlreiche E-Mails mit Werbung empfangen werden.

Die Analyse der Empfangszeiten der E-Mails hat gezeigt, dass deutliche Muster von Häufungen der empfangenen E-Mails zu bestimmten Zeiträumen erkennbar sind. In dem verwendeten Datensatz war klar erkennbar, dass sich eine Häufung zwischen 16 und 18 Uhr abzeichnet. Dies ist für viele Personen die Zeit um Feierabend zu machen. Das ist insofern eine gute Zeit, da viele Leute dann erschöpft und somit unvorsichtiger sind und leichter auf eine Spam-Mail hereinfallen.

Bei der Analyse der auftretenden Spam-Wörter in den E-Mails wurde eine mithilfe häufig verwendeter Wörter eine SSpam-Wort-Listeërstellt. Die Inhalte der E-Mails wurden dann mit der Liste überprüft um zu sehen wie häufig bestimmte Wörter auftreten. Dabei war Ängebot”mit 5092 Treffern klar auf Platz 1

Abbildung 7: E-Mail Eigenschaft datetime

```
messages = parse_folder(root)

df = pd.DataFrame(messages)
df.to_csv('MessagesPD.csv', index=False)

df['datetime'] = df['datetime'].dt.tz_localize(tz='UTC')
df['datetime'] = df['datetime'].dt.tz_convert(tz='Europe/Paris')

df['hour'] = df['datetime'].dt.hour + df['datetime'].dt.minute / 60
df['date'] = df['datetime'].dt.year + df['datetime'].dt.dayofyear / 365

plt.clf()
ax = sns.scatterplot(x="date", y="hour", s=10, alpha=.3, linewidth=0, marker=".", data=df)
ax.set(xlim=(2020, 2022.5), ylim=(1, 24))
ax.invert_yaxis()
sns.despine()
ax.get_figure().savefig("plot.png", dpi=1200)
```

Abbildung 8: Python Code - Auswertung hinsichtlich der Empfangszeiten

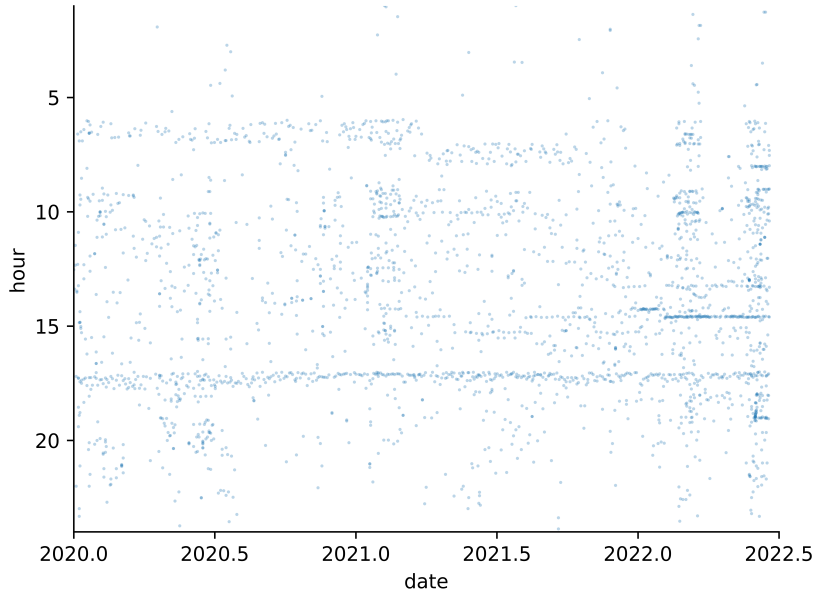

Abbildung 9: Zeitliche Verteilung der E-Mails

```
spamWordList = ['€€€',
                '$$$', '100% gratis', '100% kostenlos', 'Sichere Anlage', 'Bargeld', 'Bargeld-günstig', 'Begünstigter',
                'Cash', 'Einkommen verdoppeln',
                'Einkommen von Zuhause', 'Extra Bargeld verdienen', 'Fondsmanagement', 'Geld verdienen leicht gemacht',
                'Günstiger Kredit', 'Günstige Refinanzierung',
                'Hypothek', 'Ihr Zahlungsverzug', 'Kostenlos', 'Kontosicherheit', 'Paypal', 'Rechnung', 'Rendite',
                'Senken Sie Ihre Hypothek', 'Schulden beseitigen',
                'Verdienen Sie "x" pro Woche', 'Versteckte Kosten', 'Viel Geld sparen', 'Visa/Mastercard',
                'Völlig kostenlos', 'Von Zuhause arbeiten',
                '100% unentgeltlich', 'Vergütung', 'Geschenk', 'Rückzahlung', 'gebührenfrei',
                'kostenlos anmelden', 'gratis testen', 'Gewinn',
                'garantiert', '100% sicher', 'zertifiziert', 'risikofrei', 'Angebot', 'Bonus',
                'Schnäppchen', 'gesehen im TV', 'Deal', 'es funktioniert', 'keine Abzocke',
                'kein Spam', 'keine versteckten Kosten', 'Passwort', 'persönliche Angaben',
                'machen Sie schnell', 'sofort', 'jetzt profitieren', 'limitiertes Angebot', 'Ab jetzt', 'nur heute',
                'zögern Sie nicht', 'greifen Sie zu', 'läuft bald ab',
                'Sie haben gewonnen', 'Bravo', 'klicken', 'um Ihr Geschenk zu erhalten', 'außergewöhnliches Geschenk',
                'gewinnen', 'Sie wurden ausgewählt', 'Glückwunsch', 'Belohnung',
                'Gewicht verlieren', 'schnell abnehmen', 'zu viele Kilos', 'wie abnehmen', 'Diät', 'Verjüngung',
                'Schonkost', 'Erektion', 'Falten', 'Schnarchen', 'Altern', 'Glatze', 'ohne Aufwand', 'Ausdauer',
                'Abnehmen über Nacht', 'Besser im Bett werden', 'Cellulite weg in XXX', 'Drogen legal kaufen',
                'Falten entfernen', 'Glatze weg', 'Gras günstig kaufen', 'Günstige Medikamente',
                'Haarausfall behandeln', 'Leicht abnehmen', 'Prognose', 'Schnarchen behandeln',
                'Schnell Gewicht verlieren', 'Sixpack über Nacht', 'Sofort Gewicht verlieren', 'Kilos verlieren',
                'Therapie', 'Valium',
                'Viagra', 'Weed günstig kaufen', 'Wunderheilung',
                'Abverkauf', 'Alles muss raus', 'Angebot endet heute', 'Angebot läuft „x" ab', 'Ausverkauf',
                'Begrenzte Zeit', 'Chance nicht verpassen', 'Countdown läuft', 'Discount',
                'Eilig', 'Greifen Sie zu', 'Jetzt kaufen', 'Jetzt sichern', 'Jetzt zugreifen', 'Jetzt zuschlagen',
                'Kaufen, kaufen, kaufen', 'Kräftige Rabatte', 'Limitiertes Angebot',
                'Nur heute verfügbar', 'Nur solange der Vorrat reicht', 'Preissensation', 'Preisknüller', 'Schnäppchen',
                'Special Deal', 'Sonderangebot', 'Attraktiv', 'Date', 'Exklusives Kennenlernen', 'Fetisch',
                'Freunde finden', 'Heiße Männer/Frauen', 'Lieber Freund', 'Nicht mehr allein sein',
                'Nicht mehr einsam sein', 'Nude', 'Partner finden', 'Partnerschaftsanfrage', 'Sex', 'Sexy',
                'Sexy Männer/Frauen', 'Singles kennenlernen', 'Traummann/Traumfrau', 'Treffen', 'Völlig harmlos',
                'Völlig unverbindliches Treffen', 'Achtung!', 'Bitte helfen Sie mir', 'Dies ist kein Spam', 'Dringend',
                'Endlich online', 'Freier Zugang', 'Für Sie', 'Hier klicken', 'Ihre angeforderten Informationen',
                'Jetzt anrufen', 'Jetzt handeln', 'Jetzt öffnen', 'Jobangebot', 'Kostenlose Info',
                'Neue Herausforderungen', 'Nicht löschen', 'Profis', 'Sehen Sie sich dies an', 'STOP', 'Vergleichen',
                'Werden Sie Ihr eigener Chef', 'Wie im Fernsehen gesehen', 'Wir haben eine Stelle für Sie', 'XXX'

            ]
```

Abbildung 10: Spamwortliste

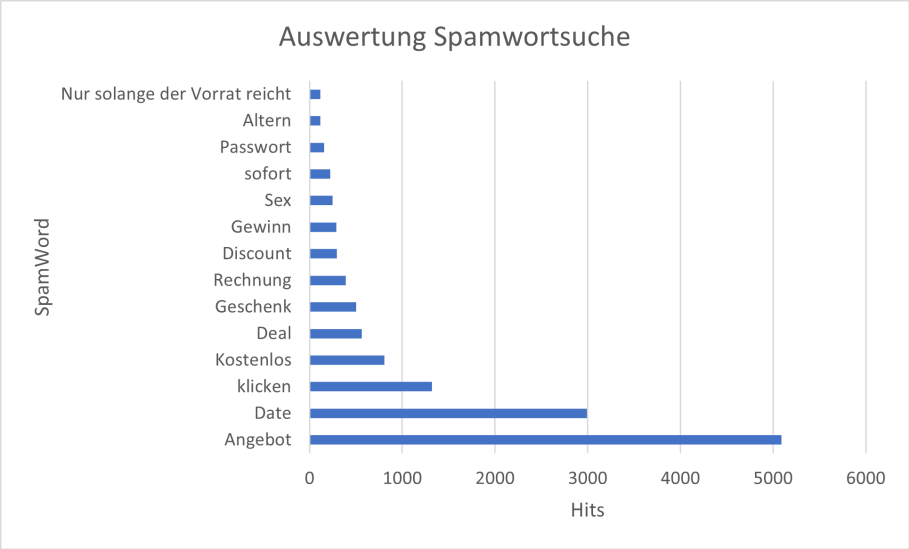

Abbildung 11: Wörter mit mehr als 100 Treffern

und ”Date”mit 2991 Treffern auf Platz 2. Somit kam das Wort Angebot im Schnitt in 1,2 E-Mails vor.

Zu erwähnen ist jedoch, dass diese Ergebnisse nicht repräsentativ sind. Um bessere Ergebnisse zu erzielen bräuchte man zum einen mehr E-Mails und zum Anderen mehrere E-Mail Konten verschiedener Benutzer um andere Verhaltensmuster im zu analysierenden Datensatz vorzufinden.

Literaturverzeichnis

[1] Barracuda networks, 26.06.2022.

[2] A-SIT Zentrum für sichere Informationstechnologie – Austria. Onlinesicherheit - spam, 26.06.2022.

[3] GitHub. Home · libyal/libpff wiki, 26.06.2022.

[4] Statista. Anzahl der e-mails pro tag weltweit 2025, 26.06.2022.