iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.51446

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Improving Digital Forensic Investigations through Automated User Entity Correlation

Glenn Nor[1], Dr. Mabrouka Abuhmida[2], Dr. Eric Llewellyn[3]
[1, 2, 3]*Faculty of Computing, Engineering and Science, University of South Wales*

Abstract: ***Digital forensic investigation is a time-consuming process, particularly when it comes to manually correlating information between different custodians. Existing methods have been limited in their ability to provide a complete overview of relevant activities and events. In response, this research project has developed a new framework that uses metadata and document entity correlation to identify correlations between custodians. The resulting insights are novel, providing a unique overview of custodian data and a clearer understanding of document content and revisions. Using this framework, digital forensic investigators can extract relevant activity or event-based data, create custom activity or event-based correlation data, and generate event graphs. This approach is an efficient and practical way to generate actionable insights for large-scale investigations.***

Keywords: Digital forensics, entity correlation, event-based data, framework development, custodians.

I. INTRODUCTION

Digital forensic investigations often require investigators to manually collect and analyze a vast amount of data in order to get an overview of people, activities, and events. One of the most time-consuming tasks for investigators is to correlate the activities of a specific user or custodian to identify any connections between their activities and events in the digital evidence. This process involves manually searching through disparate data sources, filtering, and grouping data, and identifying correlations between entities, such as documents, emails, and file creations.

To address this problem, the research project described in this paper aims to create a new way for digital forensic investigators to get an overview of custodian activities by generating automated correlation data for use in timeline or graph-based visualization. The project's objectives are to extract relevant activity or event-based data for use in timeline or graph generation by researching file metadata and other relevant evidence information, design a framework that allows the creation of custom activity or event-based, custodian-specific correlation data that can be used in the generation of event graphs, and test the theoretical framework by creating proof-of-concept Python implementation code. This approach leverages advances in data mining and visualization techniques to reduce the manual effort required in digital forensic investigations and potentially uncover previously unnoticed connections between entities and events.

Recent research has demonstrated the potential of machine learning and graph-based approaches to streamline digital forensic investigations [1]. By automating the correlation of entity object data and integrating it with visualization tools, investigators can more efficiently explore the relationships between various entities involved in an investigation, improving both the speed and accuracy of the analysis process [2]. This study aims to build upon these advances and further develop the methodology and tools necessary for the automated correlation of entity objects in digital forensic investigations.

II. LITERATURE REVIEW

Digital forensics is a complex process that involves the collection and analysis of electronic data to uncover evidence related to a crime or incident. One of the most critical parts of a digital forensic investigation is obtaining an overview of people, activities, and events. The process of obtaining this overview can be very time-consuming, especially when dealing with large data sets. This is where the concept of automated correlation of user entity objects comes into play.

Carbone and Bean [3] pointed out that the majority of tools in digital forensics have limited timeline visualization capabilities or lack the capabilities altogether. Olsson and Boldt [4], Hales [5], and Osborne and Turnbull [6] also supported this research, showing limited capabilities in visualization and forensic analysis procedures in digital forensics. However, some examples of visualization tools have been developed and used in various digital forensic areas. For instance, Schrenk and Poisel [7] used visualization to detect anomalies and attacks in network forensics, Lowman [8] used visualization to assist in understanding web histories, and Meng et al. [9] visualized emails. These tools, however, do not give investigators a complete picture of timeline events.

©IJRASET: All Rights are Reserved

To address this limitation, Henseler and Hyde [10] used AI techniques such as Graph Neural Networks (GNN) to discover relationships and patterns in digital forensic evidence. They collected forensic artifacts extracted from structured databases maintained by the operating system and applications to build relational graphs of identifiers and a timeline of events. However, the technique is limited and is an important part of taking small-scale event correlation to a more complete, large-scale version.

Another excellent example of a timeline analysis development is from Nisén [11]. He created timeline analysis software for security incident events. By using a combination of data visualization and timeline production, it was possible to get an overview of security incidents by graphically viewing network traffic load, IP communication, and disparate system logs connected and viewed as a single event.

Various attempts have been made to solve this problem using different approaches. For example, Hargreaves and Patterson [12] created high-level timelines of low-level events, and Chatbot et al. [13] reconstructed events using automated timeline creation. Although these projects and others like them have helped the digital forensic community to explore new ways of working with digital forensic evidence, research in this field is limited and mostly focuses on specific sub-branches in digital forensics.

One of the most critical aspects of a digital forensic investigation is file-based events. Hibshi et al. [14] have previously shown interest in visualization techniques that can help reduce manual review. Previous research has presented visualization and abstraction as the best solution to do this [15]. Abstraction reduces irrelevant data and allows for the visualization of a relevant reduced section of the evidence data [16][17]. Furthermore, temporal abstraction identifies system event timestamps and correlates the chain of events [18]. Although the last one is very old, the design and concept still has value when talking about and designing frameworks in this area of digital forensics.

This research project focuses on multiple data sources for extracting relevant time-based information, such as metadata and content-based information such as documents for graph node generation. Parts of this project will, therefore, deal with temporal analysis. There are two temporal analysis methodologies, as described in Inglot et al. [19]: The first one uses file system timestamps, such as Modified, Accessed, Changed (MAC).

The second one extracts timestamps from multiple sources, such as logs, files, registry keys, and registry keys, and others to create a more accurate representation of events. Temporal analysis has also been used in conjunction with timeline creation [20], which is something this project attempts to improve upon.

Adderley [21] has conducted extensive research into the creation of a graph-based temporal analysis for use in digital forensics. The focus was on temporal event reconstruction using a combination of abstraction and visualization techniques. The research is valuable in digital forensic investigations as it provides investigators with a clear view of system and user events found on the forensic image. It presents when events happened, the chain of events that led to them, what system resources were involved, and more. For example, when software is installed, many things occur in conjunction with it, with multiple sources of metadata and other data sources of interest.

While the event will be sent through an abstraction process, it will also be enriched with useful additional information that provides context and insights, such as vendor, version, and error status.

However, Adderley's research does not provide this type of information for file-based user activity and events. We can see what the users did on the machine, but not what the user created or modified throughout the available evidence. This portfolio project aims to address this by providing more intimate user event reconstruction, answering questions about what happened, when, by whom, and in what context.

Hibshi et al. (2011) [14] and Pati & Avinash (2016) [15] have previously shown interest in visualization techniques that can help reduce manual review in digital forensic investigations. The use of abstraction and visualization is presented as the best solution to achieve this goal. Hargreaves & Patterson (2012) [12] and Chatbot et al. (2014) [13] have attempted to solve parts of this problem using various approaches, such as high-level timeline creation of low-level events and event reconstruction using automated timeline creation.

Overall, research in the field of digital forensic investigations with regards to file-based user activity and events has been limited. Previous work has mostly focused on specific sub-branches in digital forensics, such as network forensics or web history visualization. There is a need for more complete timeline/event functionality to provide a more efficient approach to analyzing and evaluating large amounts of digital evidence. This project will focus on multiple data sources for extracting relevant time-based information, such as metadata and content-based information, such as documents for graph node generation, as well as temporal analysis.

III.METHODOLOGIES

The test documents in this research paper will be analysed using a machine learning technique called Named-Entity Recognition (NER), which for simplicities sake we will refer to as entity extraction. The resulting entities will be added to a database along with metadata such as origin location and custodian names.

Entity extraction works by analysing text in documents in order to identify and classify words of certain entity classes, such as: person, organization, place, quantity and more. This part of the design will not only perform entity extraction from multiple documents, but also identify entity classes that are popular and with the help from our next subsection, identify if there are multiple custodians with documents of the same entity classes. For a digital forensic investigator, these entity classes will serve as topics. It tells the digital forensic investigators what kind of documents they have, and if they contain entity classes, or topics, that correlate.

A nice way to illustrate how entity extraction works and looks like, is to view an example of a text with identified entity classes. The following example was taken from Aiimi.com [22] and shows a graphical representation of the technique in action:

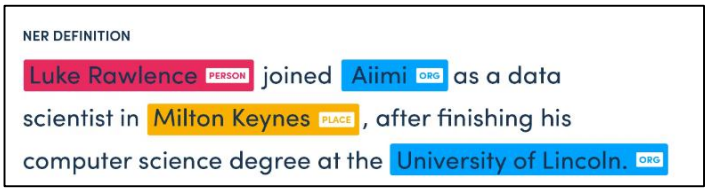


Figure 1: Entity Extraction Example

We leverage entity extraction, and build a model we can use to later create a python program proof-of-concept:

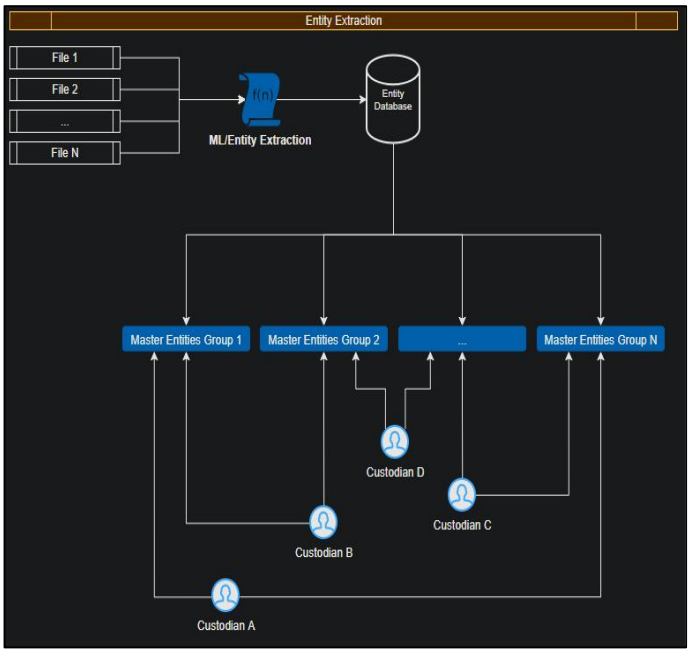


Figure 2: Program design for entity extraction

A. Proof-of-Concept

As we will only focus on proof-of-concept python code, the selection of which Entity Extraction technique (dictionary-based, machine learning) or if machine learning is selected, which machine learning technique (NLTK, Spacy), is not necessarily that important. The purpose of this research project is to illustrate what these techniques can be used for, to create insight for digital forensic investigators.

That being said, we do need an accurate proof-of-concept model in order to show the value of our proof-of-concept. We believe that the machine learning approach will be the best choice in this project as it has shown to be more adaptable. We can also see in literature that when evaluating between NLTK and Spacy, it tends to favour Spacy [23].

B. Evaluation of accuracy

Before we blindly use the Spacy's pre-trained machine learning model in our proof-of-concept, we are going to perform a simplified evaluation of the accuracy of Spacy's model. We will not be using standard metrics like F-Score, Recall etc. but rather just take a look at the confusion matrix [24] to see if we have an acceptable accuracy.

The confusion matrix consists of four defined versions of detected variables: True Positive, False Positive, True Negative and False Negative. These four variables can be displayed in a more logical structure which is defined as confusion matrix:

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 3: Confusion Matrix

We will be focusing on the true positive and false negative.

By feeding our model with pre-classified texts, sentences, or words we can count and calculate the overall accuracy with regards to how many were correctly identified and how many failed to be identified.

Spacy has 18 Entity Extraction Classes [25]:

Table 1: Entity Extraction Classes

Type	Description	Example
PERSON	People, including fictional	Fred Flintstone
NORP	Nationalities or Religious or Political Groups	Republican Party
FAC	Buildings, airports, highways, bridges, etc.	Logan international Airport, The Golden Gate
ORG	Companies, agencies, institutions etc.	Microsoft, FBI, MIT
GPE	Countries, cities, states	France, UAR, Chicago, Idaho
LOC	Non-GPE locations, mountain ranges, bodies of water	Europe, Nile River, Midwest
PRODUCT	Objects, vehicles, foods, etc.	Formula 1
EVENT	Named hurricanes, battles, wars, sports events, etc.	Olympic Games
WORK OF ART	Titles of books, songs, etc.	The Mona Lisa
LAW	Named documents made into laws	Roe v. Wade
LANGUAGE	Any named language	English
DATE	Absolute or relative dates or periods	20 July 1969
TIME	Times smaller than a day	Four hours
PERCENT	Percentage, including "%"	Eighty percent
MONEY	Monetary value, including unit	Twenty cents
QUANTITY	Measurements, as of weight or distance	Several kilometres, 55Kg
ORDINAL	"First", "second", etc.	9th, Ninth
CARDINAL	Numerals that do not fall under another type	2, Two, Fifty-Two

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

For our evaluation we are going to select three of these classes to run tests on: GPE, PERSON and DATE. To test these three classes, we are going to use the Named Entity Recognition Data (NER Data) from Kaggle [26], which was created for testing purposes of pre-trained machine learning models of Entity Extraction. We will use this dataset as a base, and create a python script that will filter out relevant test data words:

```
##########################################################################################
# Take NER Data file - Filter out test words
##########################################################################################
data = pd.read_csv(r'C:\Users\glenor\Documents\PhD_3\TestData\ner.csv')
unique_tags = data.labels.apply(lambda x: pd.value_counts(x.split(" "))).sum(axis = 0)
tags = unique_tags.keys().tolist()[1:]

def generate_test_Clean(CLASS, data):
    Main = []
    for h in range(len(data)):
        # Raw Values
        TextItem = data["text"][h]
        LabelItem = data["labels"][h]

        # Tokenized Values
        TextItem_Token = TextItem.split(" ")
        LabelItem_Token = LabelItem.split(" ")

        try:
            # Test data identification
            Location = LabelItem_Token.index(CLASS)
            Item = TextItem_Token[Location]

            # Test Data List Creation
            Main.append(Item)

        except:
            pass

    return Main
```

Figure 4: Python code for prepping data for entity extraction

We run the function and accumulate the accuracy scores for all three entity extraction classes:

```
# Create Test Lists
GPE = list(set(generate_test_Clean("B-gpe", data)))
PER = list(set(generate_test_Clean("I-per", data)))
DAT = list(set(generate_test_Clean("B-tim", data)))
```

Figure 5: Python Clean List Creation

We write a python function to perform the evaluation:

```
##########################################################################################
# Function for evaluating Spacy model
##########################################################################################
def Evaluate_NER(CLASS_ID, TEST_CLASS_LIST, NER_MODEL):
    Score_Corr = 0
    Score_Incorr = 0

    for t in range(len(TEST_CLASS_LIST)):
        ItemCheck = TEST_CLASS_LIST[t]
        ModelCheck = NER_MODEL(ItemCheck)

        # Will only have relevant result if entity is found.
        # We must therefore handle non-entity situations
        try:
            for ent in ModelCheck.ents:
                ModelResult = ent.label_

            #print("Correct: {} | Test: {} [{}]".format(CLASS_ID, ModelResult, ItemCheck))

            # Count Correct
            if (ModelResult in CLASS_ID):
                Score_Corr += 1

            # Count Incorrect
            else:
                Score_Incorr += 1
        except:
            pass

    Result = dict()

    Result["Correct"] = Score_Corr
    Result["Incorrect"] = Score_Incorr

    return Result
```

Figure 6: Python code for evaluating Spacy Model

The "Correct" variable represents the true positive, and the "Incorrect" variable represents the false negative. Since our lists are only of one (or multiple close) classes, an incorrect count will automatically be a false negative. All words should in a best-case scenario be correct.

To run the evaluation and calculate the scores we create the following python function:

```
##############################################################################################
# Function for running and counting evaluation scores for Spacy model
##############################################################################################
# Spacy Pre-Trained English ML Model
NER = spacy.load('en_core_web_sm')

print()

# Test GPE
print("Testing eval: GPE")
Q = Evaluate_NER(["GPE", "NORP"], GPE, NER)
print(Q)
print("\n")


# Test PERSON
print("Testing eval: PERSON")
R = Evaluate_NER(["PERSON", "ORG"], PER, NER)
print(R)
print("\n")

# Test DATE
print("Testing eval: DATE")
S = Evaluate_NER(["DATE", "CARDINAL", "ORDINAL"], DAT, NER)
print(S)
##############################################################################################
```

Figure 7: Python code for calculating evaluation score

We run the function and accumulate the accuracy scores for all three entity extraction classes:

Table 2: Results of the accuracy evaluation

NER Class	True Positive	False Negative	Percentage Accuracy
GPE	372	33	91.8 %
PERSON	2381	535	81.6 %
DATE	742	150	83.2 %

Considering that there are words in the dataset that could have multiple meanings, and therefore classifications, these scores between 81.6 % and 91.8 % are more than acceptable for our proof-of-concept.

C. Entity Extraction for Word Documents

The first thing we need to create is a base proof-of-concept python code that can handle documents as input and entity classes as output. Later we will build upon this to handle linking entity classes between multiple custodians. For this research project, we have chosen to only look at word documents when we use this design. However, this can be expanded at later point to include just about any document that contains user-generated text. To handle the import of text from docx word files we can use the python module "python-docx" [27]. This module only allows for paragraph-by-paragraph extraction of text, so we need to create a function to handle the extraction of all text. All we need to import text from word and pass it to Spacy's pre-trained machine learning model is the following python code:

```
##################################################################
# Function to handle importing of Word (DOCX) data to string
##################################################################
def ExtractText(document):
    Doc = docx.Document(document)
    allText = []
    for para in Doc.paragraphs:
        allText.append(para.text)
    return '\n'.join(allText)
##################################################################

# Base Location
os.chdir(r"C:\Users\glenor\Documents\PhD_3\TestData\Documents")

# Spacy Pre-Trained English ML Model
NER = spacy.load('en_core_web_sm')
```

Figure 8: Python code for handling word doc and initialize spacy

We can now pass the string generated from the "ExtractText" function to the NER object and it will extract all entities from it.

D. NER Custodian Profiler

Now that we have a way of importing data, convert it to string-format, and a machine learning model to extract the entities, we need to build a profile. The Entity Extraction Profiler, or NER Custodian Profile, which is its official name, will use what we set up in the previous section to generate a custodian specific entity extraction profile.

We can create an implementation of a NER Custodian profiler in python like this:

```
#############################################################################################################################
# Generate Custodian NER Profile
# Run through all word documents and collect
# Entities to a profile database
#############################################################################################################################

# Pandas Dataframe for holding all data
NER_DB = pd.DataFrame(columns=["Custodian", "Doc_Name", "Entities", "Classes", "Doc_Location"])

# Identify all word files
Custodians = glob.glob("*")

# Main Loop Through Custodian and Documents
for v in range(len(Custodians)):

    # Select a custodian to analyse
    CurrCustodian = Custodians[v]

    # Fetch and clean filenames for word documents
    CurrDocs_Full = glob.glob("{}\*.docx".format(CurrCustodian))
    CurrDocs = [x.split("\\")[-1] for x in CurrDocs_Full]

    # Loop for Named Entity Recognition
    for w in range(len(CurrDocs)):
        Current_Document = CurrDocs_Full[w]
        Current_Document_Name = CurrDocs[w]
        Current_Document_String = ExtractText(Current_Document)
        NER_ENTITIES = NER(Current_Document_String)

        NERD = dict()

        # Create lists for Entities and Labels
        for ent in NER_ENTITIES.ents:
            NERD[ent.text] = ent.label_

        Doc_Entity = list(NERD.keys())
        Doc_Classes = list(NERD.values())

        # Add relevant data to database
        NER_DB.loc[NER_DB.shape[0]] = [CurrCustodian, Current_Document_Name, Doc_Entity, Doc_Classes, CurrDocs_Full[w]]
```

Figure 9: Python implementation of NER Custodian Profiler

The NER Custodian Profile database collects all entities extracted from every word document from all available custodians.

E. Custodian Activity Correlation

An important insight that can be generated from the NER Custodian profile database is the overview of which custodians share entities across the datasets. Are there two or more custodians writing about the same locations? About the same person or persons? Entity Extraction by itself does bring valuable information, especially when introduced to the field of digital forensics, but it does not bring new or novel insight. But when we start cross-checking the extracted entities across multiple custodians, and bring together those with shared information, we are introducing new and valuable insight that has not been done in digital forensics before. To do this cross-checking of entities, we first need to collect all extracted entities into a master entity list and remove duplicates. This master list can be used to identify each document, and therefore each custodian, that has the various entities. Once the master list has been used to identify all the entity locations, we can simply remove the unique entries, meaning all entries that can only be found from documents belonging to one custodian, and we have a custodian activity correlator.
We can write a proof-of-concept implementation python code like this:

```
#############################################################################################################################
# Custodian Activity Correlation
#############################################################################################################################

# Pandas Dataframe for CAC Collection
NER_CAC_DB_Raw = pd.DataFrame(columns=["CAC_ID", "CAC_Entity", "Doc_Name", "Doc_Location", "Custodian"])

# Create a master list of all entities
# Regardless of which custodian it comes from
MasterEntity_Raw = []
for t in range(len(NER_DB["Entities"])):
    CurrEntity = NER_DB["Entities"][t]
    MasterEntity_Raw+=CurrEntity
MasterEntity = list(set(MasterEntity_Raw))

for r in range(len(MasterEntity)):
    CurrCheck = MasterEntity[r]
    for s in range(len(NER_DB["Entities"])):
        CurrEntity = NER_DB["Entities"][s]
        if (CurrCheck in CurrEntity):
            ID = [r]
            EntityCheck = [CurrCheck]
            DocName = [NER_DB["Doc_Name"][s]]
            DocLocation = [NER_DB["Doc_Location"][s]]
            Custodian = [NER_DB["Custodian"][s]]
            ItemMerged = ID+EntityCheck+DocName+DocLocation+Custodian
            NER_CAC_DB_Raw.loc[NER_CAC_DB_Raw.shape[0]] = ItemMerged

# We need to remove all unique entries
# As we are only interested in data that is found in multiple custodians
NER_CAC_DB = NER_CAC_DB_Raw[NER_CAC_DB_Raw.duplicated(["CAC_ID"], keep=False)]
```

Figure 10: Proof-of-Concept implementation for Custodian Activity Correlator

IV. TESTING

In this section we are going to use the proof-of-concept python code we wrote earlier, to test both the NER custodian profiler and the NER Activity Correlator on some test data.

A. NER Custodian Profiler Test

The following is an overview of our test data, and test custodians:

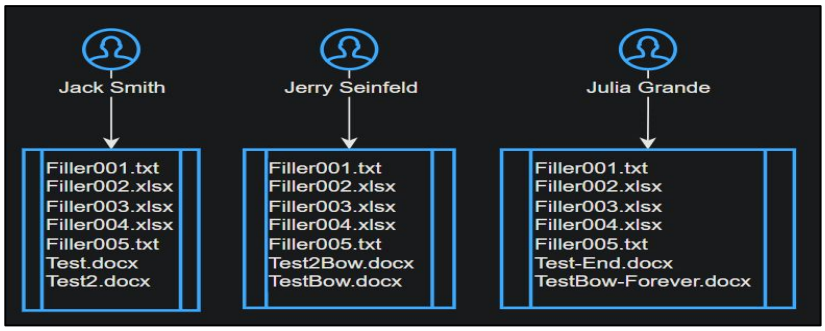


Figure 11: Overview of test data and custodians

When we run the proof-of-concept, the database is populated with metadata and then exported as Excel file.

Custodian	Doc_Name	Entities	Classes	Doc_Location
Jack Smith	Test.docx	['the centennial of the 19th', 'America', 'the past century and a quarter', 'years ago', 'the early 20th century', 'the United States', 'that era', 'just 20 percent', 'the Census Bureau', 'only 5 percent', 'African American', 'Fewer than 2 percent', '24-year-olds', 'just one-third', 'nearly 50 percent', '1930', 'nearly 12 percent', 'these years', 'first', '1920', 'Between the 1930s and', 'mid-1970s', '1970', '50 percent', '40 percent', 'First', 'World War II', 'U.S.', 'Capitol', 'Washington', 'D.C.', 'the decades from 1930 to 1970', 'two', 'the 1970s', 'the Pregnancy Discrimination Act', '1978', '1974', 'the early 1990s', 'between the ages of 25', '54', 'just over 74 percent', 'roughly 93 percent', 'the late 1990s', 'about 76 percent', 'about 89 percent', 'about 17 percent', 'each week', 'about 10 percent']	['DATE', 'GPE', 'DATE', 'DATE', 'DATE', 'GPE', 'DATE', 'PERCENT', 'ORG', 'PERCENT', 'NORP', 'PERCENT', 'DATE', 'CARDINAL', 'PERCENT', 'DATE', 'PERCENT', 'DATE', 'ORDINAL', 'DATE', 'DATE', 'DATE', 'DATE', 'PERCENT', 'PERCENT', 'ORDINAL', 'EVENT', 'GPE', 'ORG', 'GPE', 'GPE', 'DATE', 'CARDINAL', 'DATE', 'LAW', 'DATE', 'DATE', 'DATE', 'DATE', 'DATE', 'PERCENT', 'PERCENT', 'DATE', 'PERCENT', 'PERCENT', 'PERCENT', 'DATE', 'PERCENT']	Jack Smith\Test.docx
Jack Smith	Test2.docx	['3,000-year', 'California', 'Californian', 'Clarke Knight', 'US', 'Menlo Park', 'Klamath Mountains', 'the National Academy of Science', 'Rod Mendes', 'the Yurok Tribe', 'Karuk', 'Yurok', 'thousands of years', 'first', 'Mendes', 'Knight', 'Native', 'Karuk, Yurok and Hoopa Valley Tribe', 'Frank Lake', 'US Forest Service', 'Arcata', 'PhD', '2007', 'decades', 'The Karuk Resources Advisory Board', 'Indigenous', 'two', 'the Klamath Mountains', 'between 1700 and 1900']	['DATE', 'GPE', 'NORP', 'PERSON', 'GPE', 'GPE', 'PERSON', 'ORG', 'PERSON', 'ORG', 'ORG', 'GPE', 'DATE', 'ORDINAL', 'PERSON', 'PERSON', 'NORP', 'ORG', 'PERSON', 'ORG', 'GPE', 'WORK_OF_ART', 'DATE', 'DATE', 'ORG', 'GPE', 'CARDINAL', 'LOC', 'DATE']	Jack Smith\Test2.docx

Figure 12: Example output from NER Custodian Profiler

As we can see, the database assigns metadata and extracted entities to specific custodians, as well as the classes and document location.

B. Custodian Activity Correlator Test

In this section we are going to test the Custodian Activity Correlation (CAC) function. It takes the NER Custodian Profile database and looks for possible correlations between documents belonging to different custodians.

Using the proof-of-concept python code created earlier we can take a look at some of the results:

Example CAC result 1 – "the American Bowling Congress":

CAC_ID	CAC_Entity	Doc_Name	Doc_Location	Custodian
116	the American Bowling Congress	TestBow-Forever.docx	Jerry Seinfeld\TestBow-Forever.docx	Jerry Seinfeld
116	the American Bowling Congress	Test2Bow.docx	Julia Grande\Test2Bow.docx	Julia Grande

Example CAC result 2 – "Europe":

CAC_ID	CAC_Entity	Doc_Name	Doc_Location	Custodian
82	Europe	Test-End.docx	Jerry Seinfeld\Test-End.docx	Jerry Seinfeld
82	Europe	Test2Bow.docx	Julia Grande\Test2Bow.docx	Julia Grande

Example CAC result 3 – "the United States:

CAC_ID	CAC_Entity	Doc_Name	Doc_Location	Custodian
53	the United States	Test.docx	Jack Smith\Test.docx	Jack Smith
53	the United States	Test-End.docx	Jerry Seinfeld\Test-End.docx	Jerry Seinfeld
53	the United States	TestBow-Forever.docx	Jerry Seinfeld\TestBow-Forever.docx	Jerry Seinfeld
53	the United States	Test2Bow.docx	Julia Grande\Test2Bow.docx	Julia Grande

Figure 13: Custodian Activity Correlator (CAC) test results

The Custodian Activity Correlator (CAC) database can be used in many ways. One way which seems natural in a digital forensic investigation is to use the database to generate graphical overview of the results. We can see that a graphical representation of the results shown above, is immediately more intuitive:

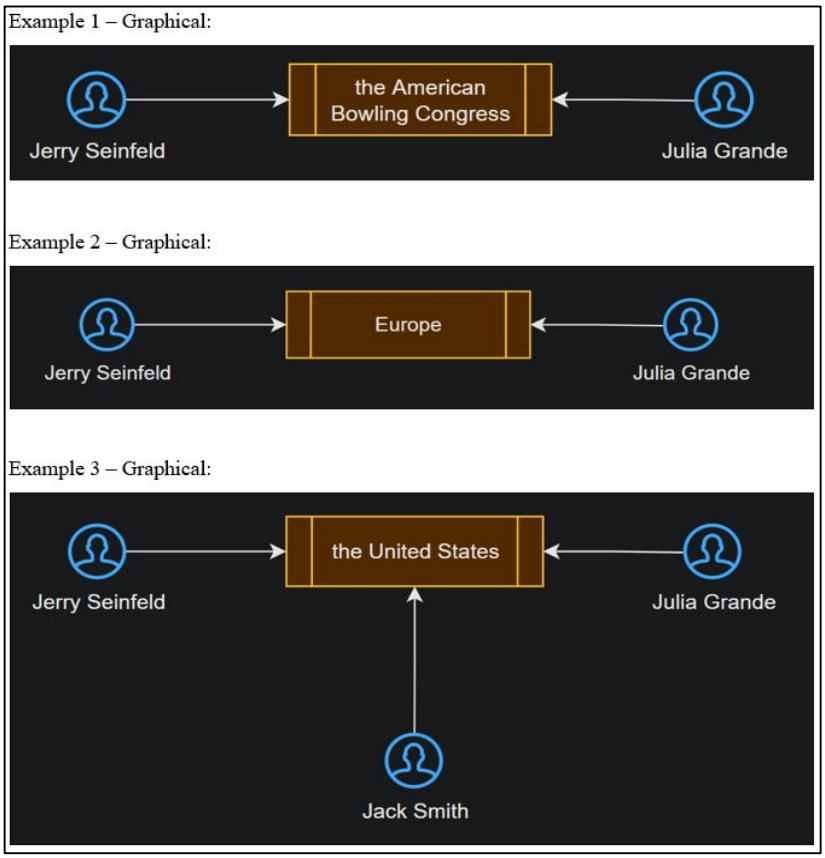


Figure 14: Example use of Custodian Activity Correlator - Graphical Representation

V. CONCLUSION

A. *NER Custodian Profiler*

The NER Custodian Profiler, or Named Entity Recognition Custodian Profiler, provides digital forensic investigators with a unique overview of what custodians have written about in their documentations, such as word files, PDF files etc. This insight by itself is not as powerful as when used together with other custodian profiles but could among other things be used to determine keyword searches for specific custodians. The NER Custodian Profiler opens possibilities for a more nuanced understanding of custodian behaviour and preferences. By analyzing the content of their written documents, investigators can uncover patterns in the language and topics that custodians engage with. This knowledge can provide valuable context to other elements of the investigation, giving investigators a more in-depth understanding of custodian actions and motivations.

B. *Custodian Activity Correlator*

By leveraging the NER custodian profiles, we can use the Custodian Activity Correlator and investigators can uncover potential connections and collaborations between multiple custodians that may otherwise remain hidden. Analyzing the similarities in the content and language used by different custodians can reveal shared interests, common projects, or even potential criminal conspiracies. This information can significantly enhance the efficiency and effectiveness of digital forensic investigations by pinpointing individuals who may warrant further scrutiny. Moreover, the Custodian Activity Correlator enables investigators to uncover patterns of communication and collaboration between custodians over time. By examining changes in the content and themes discussed in their documents, it is possible to identify critical events, shifts in relationships, or the emergence of new trends. This temporal analysis can provide valuable context for understanding the dynamics between custodians and the evolution of their activities.

VI. DISCUSSION & FUTURE WORK

The NER Custodian Profiler provides a unique perspective on the content created by custodians in various document formats, such as Word files and PDFs. By identifying and analyzing the themes and subjects discussed by the custodians, investigators can gain valuable insights into their activities, interests, and potential areas of collaboration. However, the true value of the NER Custodian Profiler becomes evident when it is used in conjunction with other tools like the Custodian Activity Correlator. The combination of these tools enables investigators to correlate activities and content across multiple custodians, offering a more comprehensive understanding of their relationships, shared interests, and collaborative efforts.

One of the primary benefits of these tools is the automation of tedious and time-consuming tasks, such as manually correlating activities and content across various data sources. The automation provided by the NER Custodian Profiler and Custodian Activity Correlator not only saves time but also reduces the risk of human error and oversight. By streamlining the investigation process, investigators can focus on the most relevant evidence, connections, and patterns, leading to more accurate and efficient outcomes.

Despite the numerous advantages offered by these tools, challenges remain in implementing them effectively in real-world digital forensic investigations. One of the primary concerns is the accuracy and reliability of the automated correlations generated by these tools. While the proof-of-concept implementations have shown promising results, there is a need for further research and testing to validate the robustness of these tools in different scenarios and data sets. Additionally, addressing potential issues related to noise, data quality, and false positives or negatives is essential to ensure the effectiveness of these tools.

In terms of future research and development, there are several avenues to explore. First, integrating the NER Custodian Profiler and Custodian Activity Correlator with other digital forensic tools and techniques, such as network forensics and malware analysis, could provide a more holistic and comprehensive view of the digital evidence landscape. Second, the development of more sophisticated machine learning and artificial intelligence algorithms could further enhance the accuracy and efficiency of these tools, enabling them to adapt and learn from different data sets and scenarios. Finally, investigating the potential applications of these tools beyond digital forensics, such as in the fields of cybersecurity, e-discovery, and corporate investigations, could unlock new opportunities and expand their impact.

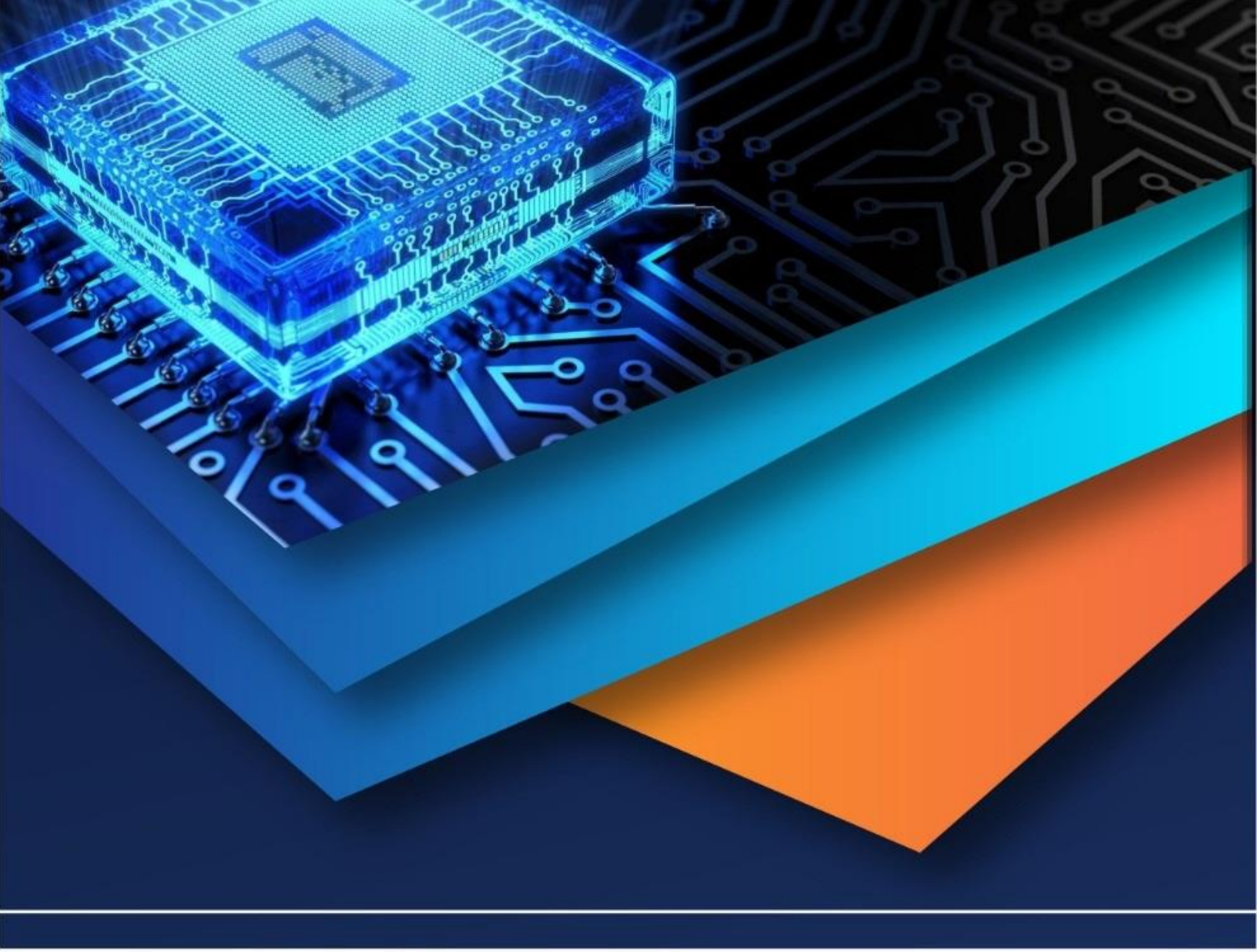
REFERENCES

[1] J. Dykstra and A. T. Sherman, "Acquiring and analyzing data from android devices," IEEE Security & Privacy, vol. 14, no. 4, pp. 54-59, 2016.
[2] M. E. Pollitt, "An admissible forensic analysis of the windows registry," in Proc. of the Digital Forensics Research Workshop, 2008.
[3] Carbone, R., & Bean, C. (2011). “Generating Computer Forensic Super Timelines Under Linux,” Defense Research and Development Canada-Valcartier Technical memorandum, pp. 1–136, 2011.
[4] Olsson, J., & Boldt, M. (2009). Computer forensic timeline visualization tool. Digital Investigation, 6(1), 78-87.
[5] Hales, G. (2017). Visualisation of Device Datasets to Assist Digital Forensic Investigation. Division of Computing Maths, School of Arts, Media and Computer Games, pp. 1–4
[6] Osborne, G., & Turnbull, B. (2009). “Enhancing computer forensics investigation through visualisation and data exploitation,” International Conference on Availability, Reliability and Security, pp. 1012–1017
[7] Schrenk, G., & Poisel, R. (2011). A Discussion of Visualization Techniques for the Analysis of Digital Evidence, International Conference on Availability, Reliability and Security, pp758-763.
[8] Lowman, S. (2010). Web History Visualisation For Forensic Investigations. MSc thesis.
[9] Meng, F., Wu, S., Yang, J., & Yu, G. (2009). Research of an E-mail forensic and analysis system based on visualization. 2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA), pp. 281-284
[10] Henseler, H. & Hyde, J. (2019a). Technology Assisted Analysis of Timeline and Connections in Digital Forensic Investigations. In LegalAIIA@ ICAIL (pp. 32-37).
[11] Nisén, P. (2013). Implementation of a timeline analysis software for digital forensic investigations. Aalto University, School of Science.
[12] Hargreaves, C. and Patterson, J. (2012). An automated timeline reconstruction approach for digital forensic investigations. Digital Investigation, vol. 9, pp. 69–79, 2012.
[13] Chabot, Y., Bertaux, A., Nicolle, C. and Kechadi, T. (2014). Automatic Timeline Construction and Analysis for Computer Forensics Purposes. 2014 IEEE Joint Intelligence and Security Informatics Conference, 2014, pp. 276-279, doi: 10.1109/JISIC.2014.54.
[14] Hibshi, H., Vidas, T., & Cranor, L. F. (2011). Usability of forensics tools: A user study. In Proceedings of the 2011 International Workshop on Systematic Approaches to Digital Forensic Engineering (pp. 81-95). IEEE.
[15] Pati, D. and Avinash, A. (2016). Effective Data Visualization using Tableau. International Journal of Engineering and Management Research, vol. 6, no. 5, pp.306–313, 2016.
[16] Nicolau, B. (2017). Visualization for real time big data. Master Thesis, Department of Information Systems and Electronic Services, Technical Universit TU Darmstadt, Darmstadt, Germany, 2017
[17] Turnbull, B. and Randhawa, S. (2015). Automated event and social network extraction from digital evidence sources with ontological mapping. Digital Investigation, vol. 13, pp. 94–106, 2015.
[18] Carrier, B. D. and E. H. Spafford. (2004). Defining event reconstruction of digital crime scenes. Journal of forensic sciences, vol. 49, pp. 1291–1298, 2004.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[19] Inglot, B., Liu, L., and Antonopoulos, N. (2012). A framework for enhanced timeline analysis in digital forensics. IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing, pp. 253–256, 2012.
[20] Casey, E. (2010). Digital evidence and computer crime: Forensic science, computers and the internet. Academic Press.
[21] Adderley, R. (2019). Graph-based temporal analysis for use in digital forensics. Journal of Digital Forensics, Security and Law, 14(1), 47-66.
[22] Aiimi. (2022). Aiimi Labs on Named Entity Recognition. Aiimi.com. Available online at: https://www.aiimi.com/insights/aiimi-labs-on-named-entity-recognition, last accessed June 5th 2022.
[23] Bhavani, D. (2019). Understanding Named Entity Recognition Pre-Trained Models. V-Soft Consulting. Vsoftconsulting.com. Available Online: https://blog.vsoftconsulting.com/blog/understanding-named-entity-recognition-pre-trained-models, last accessed: June 9th 2022
[24] Dilmegani, C. (2022). Machine Learning Accuracy: True vs. False Positive/Negative. AIMultiple. AIMultiple.com. Available Online: https://research.aimultiple.com/machine-learning-accuracy/, last accessed June 9th 2022
[25] Tripathy, A. (2020). Named Entity Recognition NER using spaCy | NLP | Part 4. Medium.com & Towardsdatascience.com. Available Online: https://towardsdatascience.com/named-entity-recognition-ner-using-spacy-nlp-part-4-28da2ece57c6, last accessed June 9th 2022
[26] Patel, R. N. (2021). Named Entity Recognition Data. NER Data. Kaggle.com. Available Online: https://www.kaggle.com/datasets/rajnathpatel/ner-data?select=ner.csv, last accessed June 9th 2022
[27] Taparia, A. (2021). Working with Documents – Python .docx Module. GeeksforGeeks. Geeksforgeeks.org. Available Online: https://www.geeksforgeeks.org/working-with-documents-python-docx-module/, last accessed June 10th 2022

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)