Analytics in Digital Forensics and eDiscovery Software - DevOps, Opportunities and Challenges

Sundar Krishnan *skrishnan@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

ABM Rezbaul Islam *ari014@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

Cihan Varol *cxv007@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

Narasimha Shashidhar *karpoor@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

Abstract

Digital forensic and eDiscovery software have embraced analytics such as machine learning and neural networks to speed up the investigation and thereby reduce costs. Since the integrity of forensic evidence is paramount to the investigation, care should be taken when working with evidence in an analytical experiment setting. Data mined from case evidence can provide different clues and together with automation, legal teams can better prepare legal arguments for the courtroom. In this paper, the authors develop a custom digital forensic software that leverages analytics and outline few development challenges and opportunities encountered along the way.

Keywords: Digital Forensic Analytics, Digital Forensics, Sexual Harassment, Supervised Learning, Hybrid Learning, Unsupervised Learning, Legal Analytics, eDiscovery, Electronic Stored Information, Case Investigation, Sentiment Analysis, Financial Fraud, Securities Fraud.

1. INTRODUCTION

Digital forensics is the process of identifying, recovering, preserving and documenting data found on digital devices (digital evidence) as data is often locked, deleted, or hidden (*Best Practices For Seizing Electronic Evidence, A Pocket Guide for First Responders*, 2018).Digital forensics technologies can be used by law enforcement to gather and preserve digital evidence as well as support or reject a case hypothesis in a court of law. The tools that aid in digital forensics are hardware and software that are carefully calibrated, audited, and maintained such that their functionality can be trusted in an investigation. Electronic evidence has become central to investigations in all types of crime, with data collected from a range of sources such as computers, smartphone, remote storage, wearable devices, unmanned aerial systems, ship borne equipment, and more (Interpol, n.d.).To cater to the wide range of evolving data sources, various open-source programs and software help digital forensic units to meet their needs.

Analytics is the process of discovering, interpreting, and communicating significant patterns in data giving us insights and meaning to data that we may not otherwise detect (Corporation,

n.d.).Insight into data through statistical processing can be applied to forecasting or outcome prediction, and this is known as Artificial Intelligence (AI) or advanced analytics. The concept of AI mimics the thought and reasoning processes of the human brain while applying it to help streamline and automate manual processes (Ron J, n.d.). Artificial intelligence (AI) has many facets, such as Machine Learning (ML), Deep Learning (DL) or Neural Networks (NN), data mining, automation, computer vision and Natural Language Processing (NLP). In this article, the author simply AI as encompassing ML, automation, and NLP. Machine learning can be described as algorithms that allow for accurate predictions of outcomes without any human interference or explicit programming. NLP allows computers to understand the text and spoken words in much the same way human beings can. With the growth of smartphone usage and low-cost Internet, people are now staying online for long duration thereby generating voluminous and complex amounts of data. This increases the difficulty to quickly analyze such volumes of data during an investigation without the support of AI. Various data-mining techniques together with ML and NLP have made forays into digital forensic software and can be used to quickly predict sentiments, profile suspects, fingerprint analysis, face recognition, file type classification, type of crime prediction, etc. It is to be noted that many eDiscovery software that mostly cater to civil litigation better leverage AI techniques on case data (evidence) than traditional digital forensic software used for criminal investigations. Reasons may be due to large scope of investigation in eDiscovery when compared to a criminal investigation where the scope is much narrower. The investigation scope dictates the volume and complexity of data collected and thus AI comes into plan more often in eDiscovery software to deal with vast amounts of data. Also, since implementation of AI in digital forensic software is in a nascent stage, investigators may want to tread carefully on its usage in criminal investigations as the penalties for a suspect many involve life or death penalties, unlike in civil litigation.

Development of a digital forensic software to cater to evidence mining, ML and NLP needs careful planning on design, data privacy and scalability. In this article, the authors design, develop and demonstrate a custom software that automates the handling of case evidence and leverages analytics (ML,NN, and NLP). The software predicts sentiments of case suspects, indicators of financial fraud and sexual harassment while pointing to their evidence sources in the case. The authors also touch upon opportunities and challenges of using analytics during development and operations of this software. Lastly, this custom forensic software can be made available for further academic research.

2. BACKGROUND

Digital forensic software and forensic techniques can be used in civil or criminal litigation to extract and analyze evidentiary data from various electronic sources. Artificial intelligence (AI)enabled digital forensic software can boost the analysis efficiency of a digital forensic investigation or in eDiscovery by quickly identifying trends, patterns, anomalies, commonalities, deep fakes, and other traits within the evidence pile. Jarrett et al.(Jarrett & Choo, 2021) conclude that AI-assisted investigations reveal a significant reduction in human mistakes, reducing inquiry time, costs, and wrong outcomes. Mitchell et al. (Mitchell, 2010) outline a few challenges that face digital forensics when applying AI and finds knowledge representation and ontology as the main challenges. The author finds that the lack of standards hinders the exchange of information between tasks in digital forensic software. Ruthann (Rughani, 2017)proposed an AI-based digital forensics framework that requires minimum user interaction and does the majority of routine operations by intelligence acquired from training. Digital forensic software leveraging AI can sometimes fail or provide incorrect results. Baggili et al. (Baggili & Behzadan, 2019) propose establishing a new discipline of AI Forensics under AI Safety to investigate cases of failure in AI systems. As the digital forensics software industry continues to embrace AI techniques in evidence analysis and presentation, groups of developers and security professionals have started to explore the application of AI reasoning in the Digital Forensics. One such group is the “DigForAsp” (Evidence Analysis via Intelligent Systems and Practices) who acknowledge that no established methodology exists today for digital evidence analysis during an investigation, and experts usually proceed by means of their experience and intuition. Bhatt et al. (Bhatt, 2017)train

an Artificial Neural Network (ANN) and analyze computer RAM along with disk images. They find forensic evidence of certain keywords that are part of the training data. While the use of analytics in analyzing forensic evidence is steadily increasing, there is little in the way of literature that outlines the development and operations of analytics driven custom forensic software along with its accompanying challenges and opportunities. In this paper, the authors propose a custom and functional digital forensic software "Digital Forensic Case Evidence Analytics" (DFCAE) that incorporates analytics and can be used by forensic investigators or eDiscovery professionals in analyzing case evidence for certain clues. The authors also touch upon the challenges and opportunities faced during the development of this software prior to discussing the highlights of this custom tool. The DFCAE software also caters to prior research undertaken by the authors in leveraging AI to mine textual case evidence ("Study Github Repository," n.d.).

3. ANALYTICS IN DIGITAL FORENSIC AND EDISCOVERY SOFTWARE

With the recent rave in analytics, such as Artificial Intelligence (AI),Machine Learning (ML), Neural Networks (NN), and Deep Learning, leveraging these techniques into custom digital forensics ore Discovery software to analyze case evidence has been highlybeneficial. Analytics, together with automation, has helpedreduce investigation time and thereby costs. With typical caseevidence data volumes every increasing due to affordable Cloud storage and cheap smartphone, mining of evidentiary data for clues and indicators to support legal arguments has become a mammoth task. Incorporating analytical approaches into digital forensic or eDiscovery software to speed-up case investigation with quality results is thus a focal area in software product development and academic research. In the below sections, the authors touch upon their challenges and opportunities faced when leveraging analytics in custom forensic software development and its use.

3.1 Analytics - Null Data

Evidentiary data can contain Null values in certain cells or for the entire row. A NULL value is a placeholder to denote values that are missing. Comparisons and arithmetic operations with a NULL produce NULL results and are thus meaningless to analytical techniques. NULL values in digital forensic data generally fall into one of two categories: values that are missing at random due to limitations of the forensic software collection mechanism and those values that are not missing at random due to design flaws of the electronic device. For example, digital forensic software extracting web browsing history may report a NULL value on a hyperlink as the browser allows for NULL hyperlinks under bookmarks, or a caller name is NULL in phone contacts of a smartphone as the device allows blank names on the contacts. If the field is allowed to be blank/NULL on the device by design, data extraction by forensic software can report it as blank/NULL. While there is a difference between blank data and null values, there a possibility that null values may exist in data collected from forensic devices. The challenge lies in how to now process null or blank data during analytics as we cannot ignore rows with null as it can result in the filtering of evidentiary data tantamount to accomplishing our goals by introducing bias. This is a problem in historical data used for analytical learning as well as during the analysis of evidentiary data at hand. As a digital forensic analyst, one should always check evidentiary data with a histogram for NULL values, blank rows or cells, null reported in a string format, and occurrences of "N/A". Ignoring all the rows containing a NULL value might not be a wise decision. Instead, prior to applying any analytical processes to the evidentiary data, it is advisable to document in detail the cells or rows (with NULL or blanks)that are 1. Missing completely at random (MCAR), 2. Missing at random (MAR) and 3. Missing not at random (MNAR). The implications of NULL values missing completely at random MCAR) can be catastrophic for the validity of the analysis techniques, investigation, and case arguments. Further attempts at forensic data extraction from the electronic device may address MAR and MNAR. If a decision is made to ignore rows or cells with NULLs, adequate documentation must be made to explain what the analysis result would be if such data was used and how in turn it would impact the result.

3.2 Repeatability, Randomness, and Sampling

According to the National Institute of Standards and Technology(NIST) NISTIR 8006 (Herman et al., n.d.) and Digital Forensic Research Workshop (DFRWS) (Pan & Batten, 2005), forensic test results must be repeatable and reproducible to be considered admissible in a legal setting. Digital forensics results are repeatable when the same results can be obtained repeatedly when using the same methods as in the same testing environment. In analytics, data preparation requires splitting a dataset (evidence data) into training and testing for supervised learning. We should avoid randomization before splitting train and test datasets as each run of the experiment will then yield different results due to the randomness involved in data selection. Seeding ensures that the Random Number Generators (RNG)output the same values in the same order each time we run it, recreating the dataset ("Randomization | Data Preparation and Feature Engineering for Machine Learning | Google Developers," n.d.). Hashing is a common way to split or sample data; however, the inputs to our hash function should not change each time we run the data generation program. Lastly, use of the current time or a random number as inputs to the hash should be avoided if we want to recreate our hashes on demand or replicate our experiment.

Class imbalance may affect our evidence datasets with more than two classes that may have multiple minority classes or multiple majority classes. Data sampling provides a collection of techniques that transforms a training dataset to balance the class distribution ("Tour of Data Sampling Methods for Imbalanced Classification," n.d.). Oversampling or under sampling should be avoided in an imbalanced class distribution. Oversampling methods duplicate examples in the minority class or synthesizes new examples from the examples in the minority class ("Randomization | Data Preparation and Feature Engineering for Machine Learning | Google Developers," n.d.). Duplication of evidence data for the sake of arriving at results in the investigation should be avoided as it interferes with the state and integrity of the case evidence.

3.3 Reporting, Logs, and Audits

Digital forensic software leveraging analytics must have plenty of visualization features like heat-maps, graphs, and charts. To the jury or at the court, statistical graphs such as for ROC, AUC, precision-recall, or accuracy may be of limited use, but rather the people/jury in the courtroom would like to see graphs and charts that they can easily infer from. Traditional reports are also encouraged, along with data exports and drill-down reports. Logs to support an audit trail are a must as part of the repeatability requirement of digital forensics. If needed, other investigators must be able to follow the logs and trigger actions on the software/software to reproduce the same results. Lastly, digital forensic software supporting analytics must allow for audits and offer an audit role type of user access with restricted access privileges.

3.4 Date & Time Format

Dates and time feature data are critical to a forensic investigation, and their formats can be detrimental to the success of the case. Care should be taken to first convert/encode all date and time data into a specific time zone and then apply proper conversion techniques. For example, pandas views date time data as strings. To convert these strings into date times(datetime64), we should use the pandas function to date time along with the format parameter and convert errors into not a datetime (NaT).

3.5 Data-warehouse or Database

A database is an organized collection of information stored in a way that makes logical sense facilitating easier searches, retrieval, manipulation, and analysis of data. Databases can be either SQL or NoSQL based. SQL based databases can scale vertically, while NoSQL can scale horizontally. SQL(relational) databases are less flexible and more rigid in terms of the data hierarchy but support queries that are easy to use and can be tuned for performance. A data-warehouse is a system that aggregates and stores information from a variety of disparate sources. Data-warehouses are designed from the ground up mainly for reporting and analysis purposes. True and verified copies of case evidence data can be imported into databases or a data-warehouse. The question arises as to which is the best data storage option for analytical experiments. As analytical experiments grow using the database, managing schema objects can

get complicated, requiring additional database administrator resources to manage the database. Similarly, a data-warehouse may seem to be a design overkill but can scale better when multiple analytical experiments are being conducted. However, data-warehouses do not support multiple concurrent connections as databases do. Storing case evidential data as a flat-file for analytical experiments is not advisable as flat-files do not support complex searches and read/write transactions as robustly as databases or data-warehouse.

3.6 Privacy PHI/PII in Evidence Data

Case evidence can contain Protected Health Information(PHI)/Personally Identifiable Information (PII)/Confidential Business Information (CBI) data causing data privacy and access concerns in handling of such data during analytical experiments. Also, to arrive a good quality of training data for supervised learning, sometimes historical case investigation data may be a good source to start with. However, using such historical data for analytical research can also raise legal concerns of privacy and ethics. While such historical legal cases may be closed and now archived, ownership of such data, and reuse of it to build a training dataset may itself need legal, privacy and client approvals. For example, to build a training dataset for Facebook posts containing financial fraud evidence, the analytical team may want to tap into forensic evidence data from historical cases. In such instances, the ownership of the historical data and related privacy concerns of its use will need to be clarified.

3.7 Encryption in Evidence Data

Sensitive evidence data that was stored in an encrypted way will need to be decrypted for use in analytical experiments. This leaves the data in an unsecured state, and care must be taken to re-encrypt it at the earliest. Likewise, results of analytical experiments utilizing this decrypted data may in-turn contain data that now needs to be secured. Allowing the analytical team to access the keys to decrypt and re-encrypt sensitive data can be a security risk.

3.8 Verification and Validation

Analytical methods and models used in digital forensics to analyze/mine case evidence can be called into question and opposed in courts. The challenge arises in the experiment/model/method verification and validation process. To better understand this challenge, we need to understand the types of data used in an analytical experiment.

1. Training data - This type of data helps build the machine learning algorithm within the analytical experiment. Data is input to the machine learning algorithm resulting in an expected output. The model repeatedly evaluates this data to learn more about the data's behavior and then adjusts itself to serve its intended purpose (Carty, 2021).

2. Validation data - During model training, new data can be infused into the model as part of validation. This new data is known as validation data or holdout set and is often 10% of the total data which was not used by the model as yet. Validation of data can be a tricky as it requires significant understanding of the data in order to select the correct approach such as k-fold cross validation or time-based splits. Validation data provides the first test against unseen data, allowing the forensic team to evaluate how well the model can make predictions based on the new validation data. The use of such validation data is uncommon but advised in a forensic analytical experiment as it can provide helpful information to optimize hyper parameters, which influences how the model assesses data (Carty, 2021).

3. Test data - After the model is built, trained, and validated, testing data once again validates that the analytical model can make accurate predictions. The testing data should be left unlabeled if the training and validation data included labels to evaluate the model's performance metrics. Test data is a last, real-world verification of an unknown dataset to ensure that the machine learning algorithm was properly trained (Carty, 2021).

Thus, utilizing validation data in the analytical experiment can provide an initial check that the model can return useful predictions in a real-world setting, which training data cannot do.

Validation data can be part of the training data but is advisable to be an entirely different dataset than the training dataset (Carty, 2021). The use of validation data can also reassure the jury or the court that the model's algorithm works as intended in predicting results as part of the analytical experiment.

3.9 Metrics and Graphs

Analytical experiment results are best represented in graphical formats along with key metrics such as model accuracy and loss. To be well understood and accepted in a court or by a jury, visualization of analytical experiment's decision-making process results such as evaluation metrics, learning curves, scatter plots, performance charts (like ROC, Lift Curve, Precision-Recall charts, confusion matrix, etc.) is critical. Further use of visualization techniques to summarize the investigation focus and analytical experiment results is advisable. For example, a bar chart on instances of sexual harassment indicators by the suspect over a period of time can be added on top of the analytical experiment's model prediction accuracy and precision-recall chart. Care must be taken to not over-burden the jury or court with statistical graphs, model architecture, and detailed metrics unless called for.

3.10 Domain Ontology Limitations

In the case of large volumes of data, automation coupled with data mining and AI can greatly speed up the forensics process and thereby allow for a quicker investigation. However, decisions made by and with the assistance of AI based forensic software need to be justifiable and explainable to a jury. Often, analytical experiments, AI algorithms, and accompanying automation tend to be too scientific for laypeople and thus EXplainable artificial intelligence (XAI) will need to be employed wherein lay explanations for outputs are provided when leveraging analytics (Hall, Sakzad, Kim-Kwang, & Choo, 2022). As AI technology and capabilities advance over time, it may become more difficult, or even impossible for AI systems to be explainable to a jury or in a courtroom. Thus, care must be taken during courtroom evidence presentation to limit results to simple graphs/ charts, metrics, graphical execution plan, drill-down reports, etc. from forensic software leveraging AI and from analytical experiments conducted on case evidence.

3.11 Multiple Analytical Approaches

Design of digital forensic software supporting analytics should involve multiple approaches and allow the user (investigator)to choose the most appropriate one. For example, if the custom forensic software addresses multi-class classification, multiple algorithms such as k-Nearest Neighbors, Decision Trees, Random Forest, Gradient Boosting and Naive Bayes may be offered by the software thereby allowing the user (investigator)to choose the most appropriate one based on classification results. This way, the software does not limit itself to one approach/algorithm but rather offers variety. Limiting to one algorithm may prove detrimental as a specific algorithm may not work best across multiple datasets(different case evidence data).

3.12 Security - Access Control, Evidence Destruction

While the case investigator may enjoy a certain degree of his/her access to the current case evidence on hand, their access to certain historical case data or labeled data will need to be considered. Case evidence may have PHI, PII, or CBI data making privacy and security key aspects of any analytical experiment. Disseminating results post analytical experiments may need such results to be circulated and stored with colleagues or shared with clients. This would call for triggering necessary data privacy and security access controls to both experiment results and other automation logs. All case evidence must have an end-of-life timeline defined. Analytical research experiments using historical or on-going case evidence must factor these timelines as results of these experiments themselves may contain copies of the original evidence. Uncontrolled sharing of these results can also lead to complications to evidence destruction.

4. SOFTWARE DEVELOPMENT METHODOLOGY

The custom software/software developed for this experiment was developed using an Intel(R) Core(TM) i5-3470 CPU @3.20GHz 16 GB RAM PC and a 64-bit Windows 10 operating system.

Software and programming language used was Python, PyCharm, SQL Server 2019, C#, and Visual Studio 2019. The custom software "Digital Forensic Case Evidence Analytics"(DFCAE) supports multiple modules such as suspect's sentiment analysis, financial fraud indicators of suspects, and sexual harassment indicators - all leveraging automation, data mining, and analytics. The software user interface was written using C#, calls necessary Python files and stores all data on a back-end SQL Server database. For software to be deployed and used by digital forensic and eDiscovery professionals, the authors decided to use WinForms and ultimately develop a client/server based Windows executable file with supporting DLL files. Each case evidence has its own database, and a common database serves as a master repository for labeled/unlabeled training data for analytics. Figure 1 shows the growing complexity of traditional database schema objects for storing forensic evidence when used for analytic experiments.

Text from case evidence (ESI) is mined using best practices (Krishnan, Shashidhar, Varol, & Rezbaul Islam, 2021), analytics and automation for results (indicators). For each upload, a new database in the SQL server instance is created and a few schema objects are automatically defined as part of SQL scripts. The software can handle evidence data from sources such as Facebook posts, Twitter data, SMS/WhatsApp messages, emails, and MS Word documents. Case investigators can switch between the three modules (Sentiments of suspects, Financial Fraud Detection of suspects and Sexual Harassment Detection of suspects) against the same case evidence. This way, the investigators have a choice to pursue different investigations against suspects of the case from the evidence collected. The investigators can store case metadata, upload evidence, review evidence statistics, trigger sentiment analysis, trigger for indicators of financial fraud and trigger for finding indicators for sexual harassment. Figure 2 and Figure 3 show the user interface screen for the investigation case metadata and ESI (case evidence) metadata. While each of these key features has been discussed in detail in previously published articles of this project (Krishnan, n.d.), we will briefly touch upon them once again.

The sentiment analysis of suspects found within the case evidence is carried out using multiple approaches and algorithms. Figure 4 shows the user interface module to detect sentiments of case suspects from case evidence. Thus, the investigators can trigger the module for multiple analytical approaches. The results are then displayed on the user screen. Currently the sentiments are either positive or negative but can be scaled depending on the training data that is uploaded via the software. Investigators can access various reporting functionality like charts, heat-maps that point to evidence source and the sentiments of the suspect. Results can be exported to a flat-file. Prior research and associated software code on the detection of sentiments of case suspects (Krishnan, Shashidhar, Varol, & Islam, 2022c)using this custom software is available on GitHub (Krishnan, n.d.).

A financial fraud detection module detects fraudulent behavior in pump and dump schemes and insider trading using multiple analytical approaches. Figure 5 shows the user interface module for detecting financial fraud indicators from case evidence. The investigators can choose stocks to target and the suspect of interest. The module predicts from evidence the sources that have strong indicators of such financial fraud. The module correlates to historical stock data from Yahoo Finance. Investigators can access various reporting functionality like charts, heat-maps that point to evidence-source and the fraudulent behavior of the suspect. Results can be exported to a flat-file. Prior research work and associated software code on the detection of financial fraud of case suspects (Krishnan, Shashidhar, Varol, & Islam, 2022a) using this custom software is available on GitHub (Krishnan, n.d.).

The sexual harassment detection module detects possible sexual harassment of a suspect using multiple analytical approaches. Figure 6 shows the user interface module for detection of sexual harassment from case evidence. The investigators can choose a suspect and trigger the module to predict indicators of possible sexual harassment. Investigators can access various reporting functionality like charts, heat-maps that point to evidence-source and the harassment behavior of the suspect. Results can be exported to a flat-file. Prior research work and associated software

code on the detection of sexual harassment indicators of case suspects (Krishnan, Shashidhar, Varol, & Islam, 2022b) using this custom software is available on GitHub (Krishnan, n.d.).

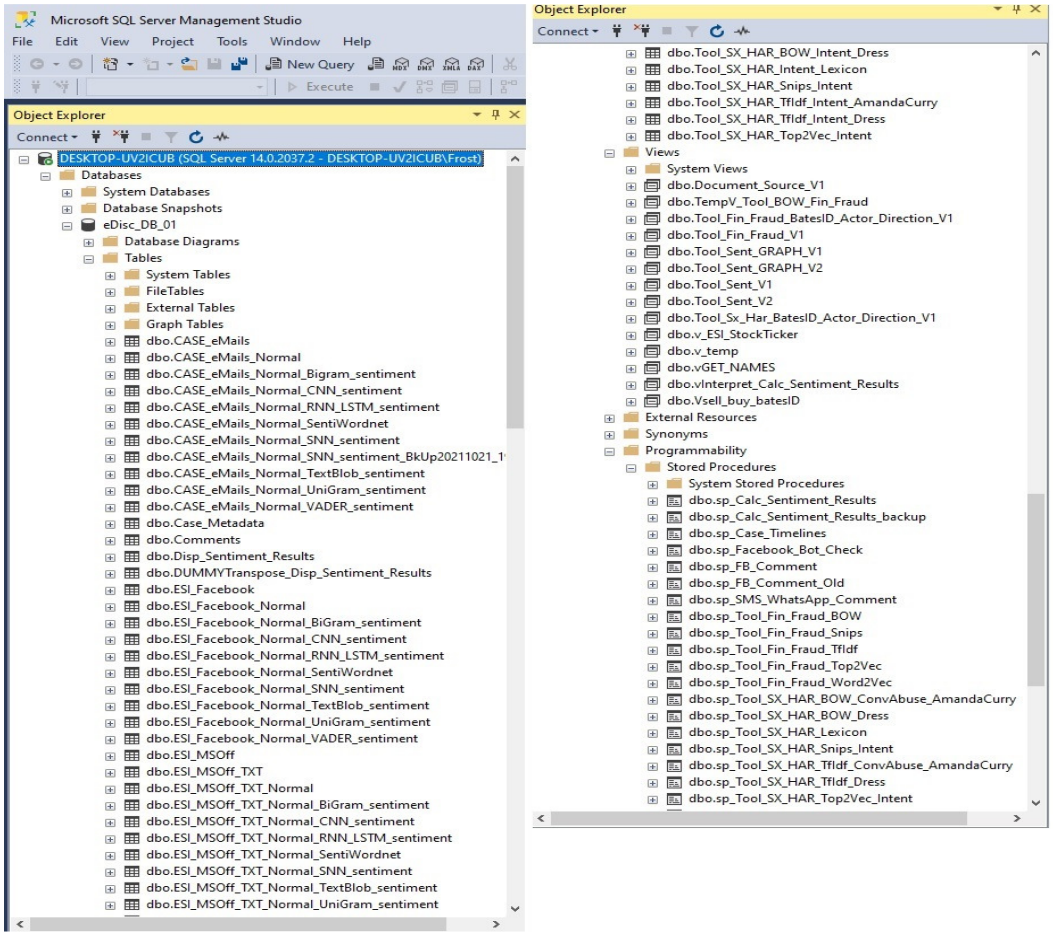

FIGURE 1: Database Schema view of custom forensic analysis software showing complexity of database schema and design when using a traditional database.

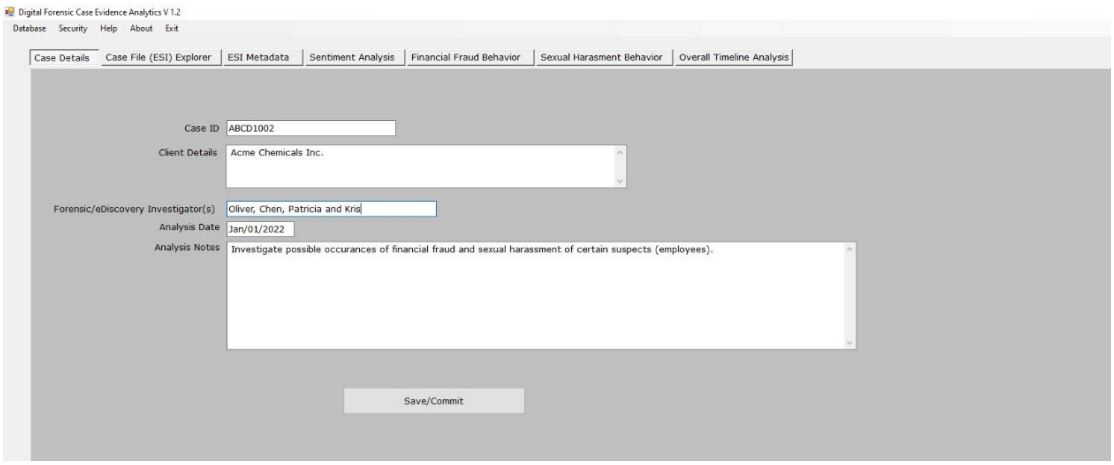

FIGURE 2: Screen capture of case metadata on our custom software.

This software allows for logging all user activity thus allowing for future audits. These activity logs are stored in the database and can be exported when needed for software usage audits or for training. Each analytical program triggered also contributes its run-time debugging information to a run-log flat-file, which can be accessed from this software. The software allows for further insight into case evidence by displaying HTML based time-series graph using Google's API for charts. Figure 7 showcases the communication timelines(from case evidence) of case suspects in our custom forensic analysis software. A help module was also created for the software, along with a security module was created for role-based access for users of this software. The source code, along with this software project files and repository, can be accessed online on GitHub (Krishnan, n.d.).

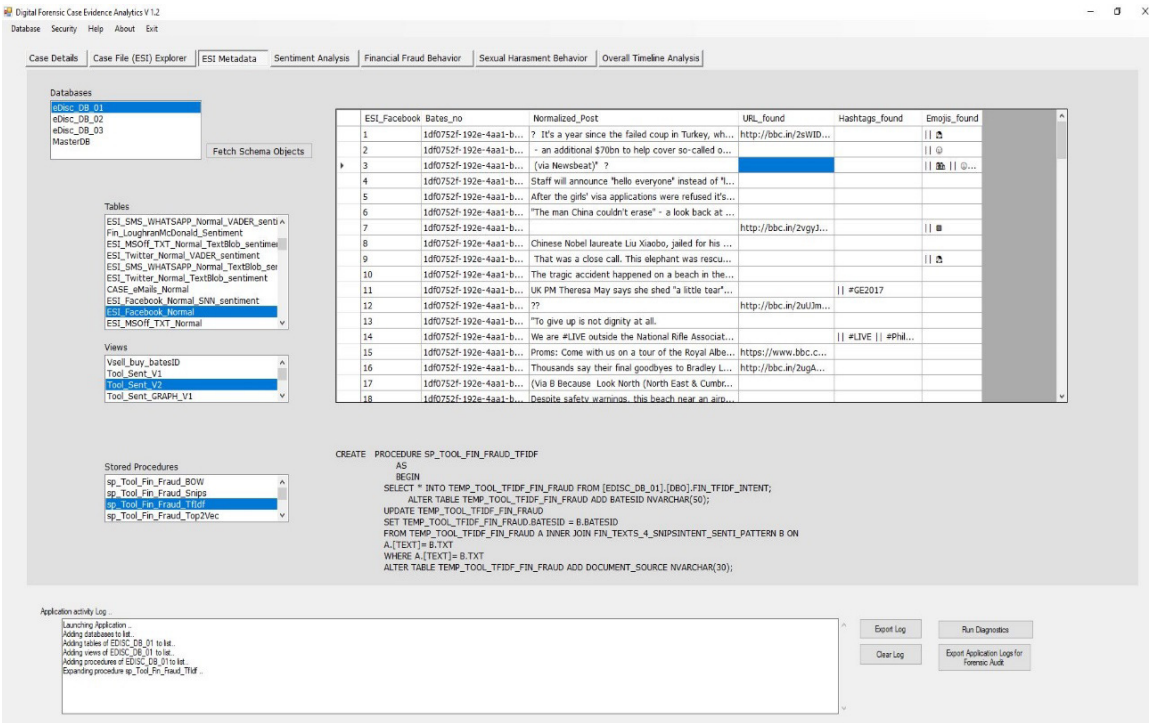

FIGURE 3: Database schema view from our custom forensic software for each case evidence (ESI).

5. CONCLUSION

Designing and developing a digital forensic software that leverages analytical techniques needs careful planning of design and back-end. The design of such software should factor in logging, security, and privacy requirements. Investigators would often need multiple analytical approaches from the custom forensic software to choose the best model. In this paper, the authors discuss a custom forensic software developed for multiple use-cases during a case investigation. The authors also discuss best practices in developing and operating such custom forensic software platform that supports analytics. As part of future work, the authors plan on adding additional modules such as steganography detection and signature detection while expanding its support for ingesting web-browser data and Portable Document Format (.pdf) files from the case evidence.

6. REFERENCES

Baggili, I., & Behzadan, V. (2019). Founding The Domain of AI Forensics. *CEUR Workshop Proceedings*, *2560*, 31–35. https://doi.org/10.48550/arxiv.1912.06497.

BestPracticesFor Seizing Electronic Evidence, A Pocket Guide for First Responders. (2018). Retrieved from https://www.cwagweb.org/wp-content/uploads/2018/05/BestPracticesforSeizingElectronicEvidence.pdf.

Bhatt, P. (2017). Machine Learning Forensics: A New Branch Of Digital Forensics. *International Journal of Advanced Research in Computer Science*, *8*(8), 217–222. https://doi.org/10.26483/IJARCS.V8I8.4613.

Carty, D. (2021). Training, Validation and Testing Data Explained. Retrieved May 4, 2022, from https://www.applause.com/blog/training-data-validation-data-vs-test-data.

Corporation, O. (n.d.). What is Big Data?

Hall, S. W., Sakzad, | Amin, Kim-Kwang, |, & Choo, R. (2022). Explainable artificial intelligence for digital forensics. *Wiley Interdisciplinary Reviews: Forensic Science*, *4*(2), e1434. https://doi.org/10.1002/WFS2.1434.

Herman, M., Ahsen, M. I., Salim, M., Jackson, R. H., Hurst, M. R., Leo, R., … Sardinas, R. (n.d.). NIST Cloud Computing Forensic Science Challenges. https://doi.org/10.6028/NIST.IR.8006.

Interpol. (n.d.). Catalogue of Digital Forensic Tools. Retrieved from https://www.interpol.int/content/download/16480/file/Catalogue of Digital Forensic Tools.pdf.

Jarrett, A., & Choo, K.-K. R. (2021). The impact of automation and artificial intelligence on digital forensics. *Wiley Interdisciplinary Reviews: Forensic Science*, *3*(6), e1418. https://doi.org/10.1002/WFS2.1418.

Krishnan, S. (n.d.). Project · GitHub. Retrieved May 6, 2022, from https://github.com/kshsus.

Krishnan, S., Shashidhar, N., Varol, C., & Islam, A. R. (2022a). A Novel Text Mining Approach to Securities and Financial Fraud Detection of Case Suspects. *International Journal of Artificial Intelligence and Expert Systems*, *10*(3). Retrieved from https://www.cscjournals.org/journals/IJAE/issues-archive.php.

Krishnan, S., Shashidhar, N., Varol, C., & Islam, A. R. (2022b). A Novel Text Mining Approach to Sexual Harassment Detection of Case Suspects. *International Journal of Artificial Intelligence and Expert Systems*, *10*(3). Retrieved from https://www.cscjournals.org/journals/IJAE/issues-archive.php.

Krishnan, S., Shashidhar, N., Varol, C., & Islam, A. R. (2022c). Sentiment Analysis of Case Suspects in Digital Forensics and Legal Analytics. *International Journal of Security*, *13*(1). Retrieved from https://www.cscjournals.org/journals/IJS/issues-archive.php.

Krishnan, S., Shashidhar, N., Varol, C., & Rezbaul Islam, A. (2021). Evidence Data Preprocessing for Forensic and Legal Analytics. *International Journal of Computational Linguistics (IJCL)*, *12*(2), 24–34. Retrieved from https://www.cscjournals.org/library/manuscriptinfo.php?mc=IJCL-122.

Mitchell, F. (2010). The use of Artificial Intelligence in digital forensics: An introduction - SAS-Space. *Digital Evidence and Electronic Signature Law Review*, *7*. Retrieved from https://sas-space.sas.ac.uk/5533/.

Pan, L., & Batten, L. (2005). *DIGITAL FORENSIC RESEARCH CONFERENCE Reproducibility of Digital Evidence in Forensic Investigations*.

Randomization | Data Preparation and Feature Engineering for Machine Learning | Google Developers. (n.d.). Retrieved May 2, 2022, from https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/randomization.

Ron J, R. J. (n.d.). The Use Of Artificial Intelligence In Digital Forensics. Retrieved April 27, 2022, from https://www.exterro.com/blog/the-use-of-artificial-intelligence-in-digital-forensics.

Rughani, P. H. (2017). ARTIFICIAL INTELLIGENCE BASED DIGITAL FORENSICS FRAMEWORK. *International Journal of Advanced Research in Computer Science*, *8*(8).

Study Github Repository. (n.d.).

Tour of Data Sampling Methods for Imbalanced Classification. (n.d.). Retrieved May 2, 2022, from https://machinelearningmastery.com/data-sampling-methods-for-imbalanced-classification/.

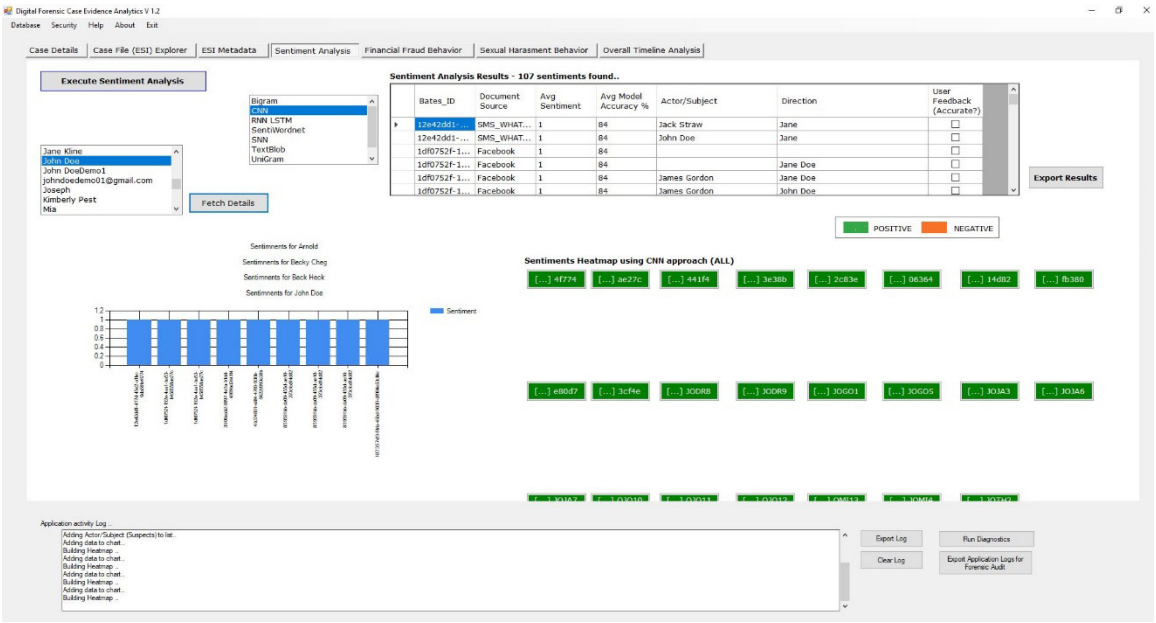

FIGURE 4: Sentiments of case suspects using in our custom forensic analysis software.

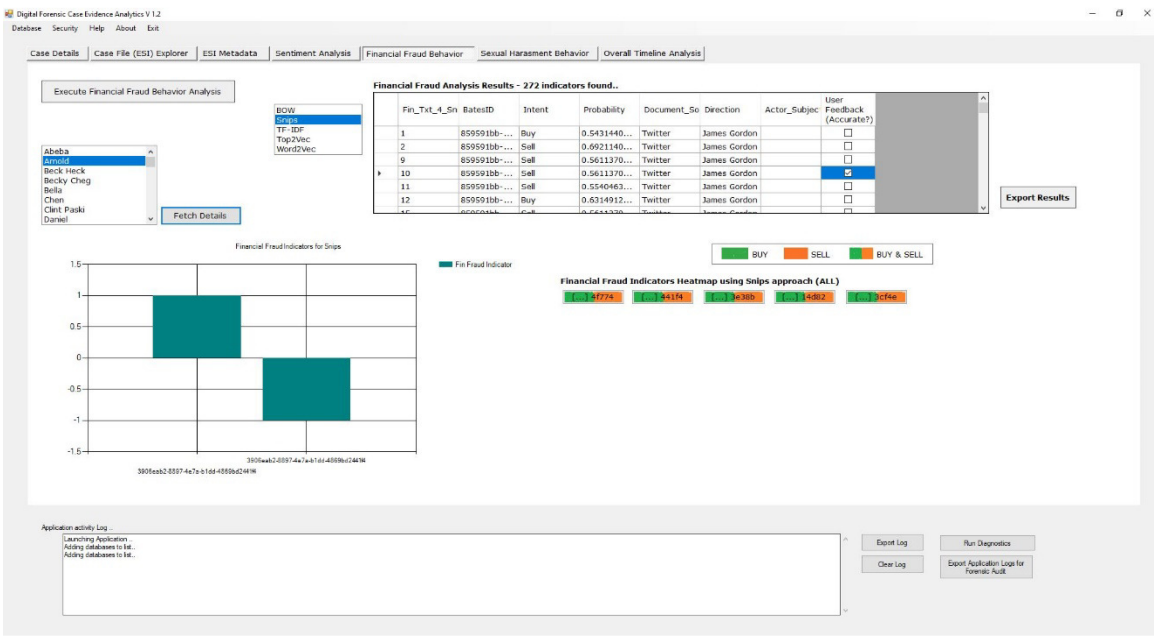

FIGURE 5: Detecting financial fraud indicators using custom forensic analysis software.

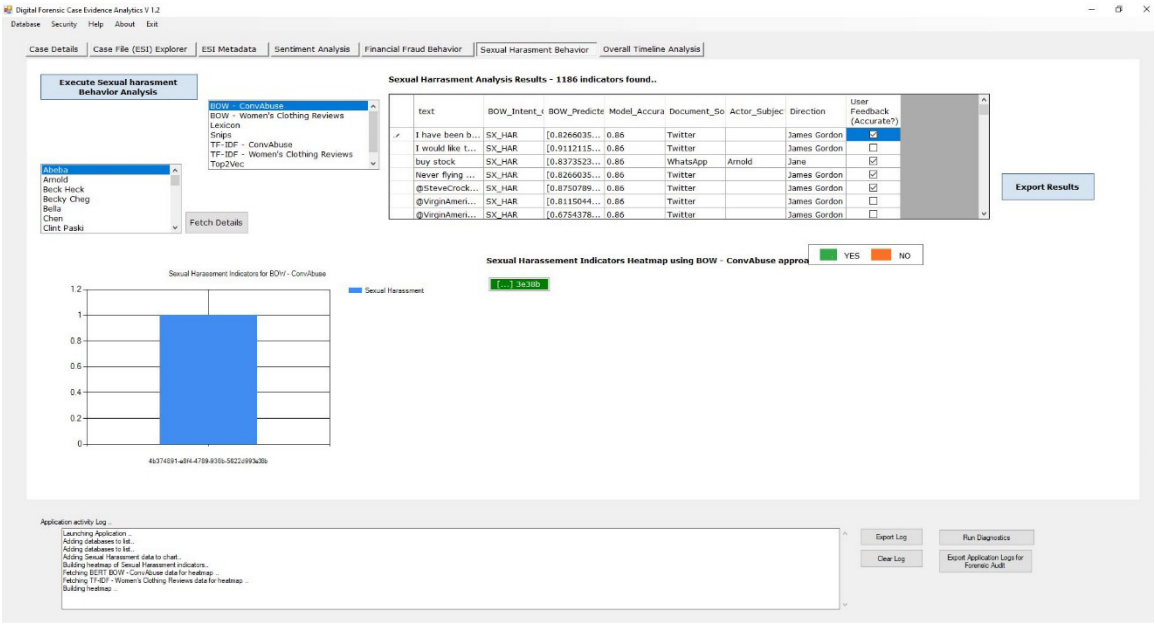

FIGURE 6: Detection of Sexual Harassment evidence in our custom forensic analysis software.

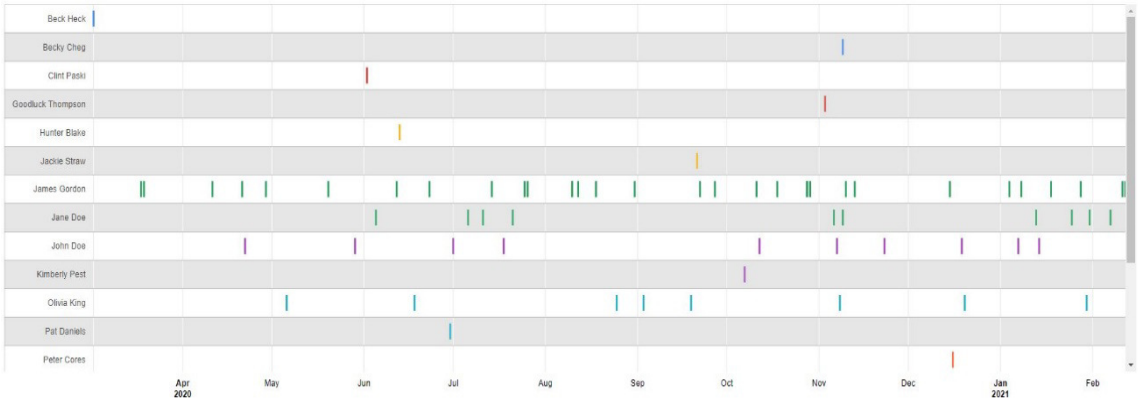

FIGURE 7: Communication timelines of case suspects using Google API in our custom forensic analysis software.