Classification of Complaints for Criminal Intelligence Purposes

Pauline Rousseau, Dimitris Kotzinos,
CY Cergy Paris University, France

Daniel Camara
Center for Artificial Intelligence, Judiciary Pole of the Gendarmerie National, France

ABSTRACT

The increase in the volume of available data is changing how people perceive their own fields and how the people may interact with this surplus of information. Public security is not different; Law Enforcement Agencies (LEAs) now have available a large quantity of information to help them fight criminality. One challenging problem is to classify/predict criminal activities. The differentiation over two different complaints may only be clear through the careful analysis of complaints' open text fields, e.g., the modus operandi, where it is described the specificity of the perpetrated crime. Sometimes the intention behind a crime is not evident unless it is correlated to other crimes and patterns get extracted from them. This chapter shows that it is possible to classify criminal data using machine learning-based methods and that open text fields, such as the modus operandi, may play a fundamental role in the performance of the classification.

Keywords: Modus Operandi, Crime, Artificial Intelligence, Machine Learning, Random Forest, Classification Methods Comparison, Metal Theft, Governmental Data, Law Enforcement

INTRODUCTION

The increase in available information impacts all aspects of society, and criminality analysis is not an exception. The access to recent technologies and the systematic digitalization of all aspects of the investigative process has largely increased the amount of available information for investigations. This enables a whole new set of possibilities for analyzing and correlating the available data, in ways investigators have never thought of before. Law enforcement agencies (LEAs) have the possibility of performing much more precise and fine-grained analyses of criminality, that were impossible some years ago. One example is the classification of criminal complaints. A complaint is a declaration by a victim over a possible crime. One example of a complaint could be a car's exhaust pipe theft. In this case the vehicle's owner can go to the local police authority and file a complaint to report the vehicle degradation.

Typically, complaints are already classified into broad categories when filed. This classification is used for statistical purposes and for directing the complaints to specialized units for investigation. However, these large classifications may not be precise enough to be helpful in criminality understanding, i.e. intelligence purposes. They may also be too broad to be useful for the specialized units, which typically work with more fine-grained classifications inside their broad activity. When trying to understand criminality, agents need a more accurate classification to detect specific phenomena. For example, the objective of the degradation could be not the exhaustion pipe itself, but the precious metal that was within it. In this case units, other than vehicle-related ones, may also be interested and concerned by the complaint. This is even more true if the phenomena are new or not yet widely known from the terrain units, that collect the complaints. Moreover, the complaints' differentiation may only be perceptible over the open text fields, e.g., the description of the crime, called modus operandi. Sometimes specialists can evaluate a crime only through the analysis of the modus operandi. For example, the objective of the vehicle degradation was to collect precious metals, which is something new in that specific region. Even if for a specialist in a specific domain, the distinction between the different sub-classification is relatively simple, the total number of precise sub-classes for all units, and above all, the raw number of entries to classify is potentially huge. Given the volume of data, manual classification is not an option.

This article evaluates the use of artificial intelligence-based methods to perform a sub-classification of complaints. Complaints consist inherently of semi-structured data, i.e., some fields are structured (e.g.: date of the fact, initial classification, the value of the damage/stolen good, age of the victim), others are open free text (e.g. Modus Operandi and the Description of Stollen Goods). The method presented in this work considers that all the available information may be important for a precise classification of criminal activities. In general, classification methods either take structured or non-structured data into account, this chapter argues that both are needed. The experimentations will show that using both types of data, a significant performance improvement may be reached. For LEAs the explainability of the method is of paramount importance. This paper considers explainable and non-explainable methods for comparison.

This paper is organized as follows, Section I Background presents the state of the art in criminal data classification. Section II Dataset Description explains the type of data used for classification. Section III Data pipeline explains the full data pipeline treatment put in place for performing the classification. Section IV Classification Methods explains the methods used for the classification. Section V Experimentations presents the results of the classification experiments. Section VI Conclusion and Future works, presents the conclusions and possible future works on the domain of criminal data classification.

BACKGROUND

A simple way to understand criminality is given by the problem analysis triangle (Clarke, 2005), also called the crime triangle. Accordingly to this theory, three elements must exist for a crime to occur: an offender, a victim, and a location where the path of the first two crosses. In some sense, the interaction of these three factors creates a criminal opportunity (Cohen & Felson, 1979). Criminal data is constantly growing; thus, manually inspecting and investigating each crime is infeasible. Data mining methods exist to obtain a more detailed view and aim to reduce the rate of criminality and provide information to feed the crime triangle. This section reviews a series of works on the automatic classification of criminal data.

Different works used different types of data, some work directly with criminal data from LEAs (Abdulrahman & Abedalkhader, 2017; Chandrasekar et al., n.d.; Hossain et al., 2020; Khatun et al., 2021; Mahmud et al., 2021; Munasinghe et al., 2015; Nath, 2006; Shojaee et al., n.d.; Sundhara Kumar & Bhalaji, 2016; Waduge, n.d.; Yao et al., 2020; Yerpude & Gudur, 2017) others use data related to the e-commerce (Xuan et al., 2018) or social media. Two datasets are popular Communities and Crime dataset (UCI Machine Learning Repository, s.d.) from UCI machine learning repository and the open dataset from SFPD crime incident (San Francisco Police Department, s.d.). Even though other datasets are also used, for example in (Nath, 2006) and (Mahmud et al., 2021).

Regarding the employed classification techniques, one of the most used methods is Random Forest (Breiman, Random Forests, 2001). Different cases of study recognize the view that Random Forest shows better performance among different methods such as Decision Tree and KNN or Naïve Bayes classifiers (Hossain et al., 2020; Khatun et al., 2021; Sundhara Kumar & Bhalaji, 2016; Xuan et al., 2018; Yao et al., 2020; Yerpude & Gudur, 2017).

The work of Hossain et al. (2020) compares Decision Tree, Random Forest, Adaboost and KNN algorithms to determine the best performance for the classification of crime categories. Hossain et al employ the oversampling method and undersampling on the same dataset to emphasize the impact between imbalanced and balanced databases over the classification. The investigation, conducted by Khatun et al.(2021), proposes to compare likewise Decision Tree, Random Forest and additionally KNN algorithm. Because of the overfitting problem caused by Decision Trees, the authors conclude that the most adapted method for analyzing arrest record prediction and also crime types of prediction is Random Forest. Yerpude & Gudur (2017) also demonstrate the strength of Random Forests for per capita violent crimes prediction. Despite the comparison with the Decision Tree, Regression and Naïves Bayes methods, Random Forests provides the highest result (F1 Score 86.54%) with cleaned data. This work also confirms the importance of the data cleaining phase in criminal data analysis. A better performance is reached when data is treated, F1-score of 86.54%, than it is not, F1-Score of 84.80%.

Xuan et al. (2018) examine two kinds of random forests (Random Forests and CART) for credit card fraud prediction. Their work emphasise the problem imbalanced data may represent and concludes that CART has a better performance than random forest, even though random forest obtains relevant results on small datasets. In the same direction, Yao et al. (2020) propose a study based on Random Forest approach to show crime hotspots based on spatial factors. This study confirms the improvement in classification when using spacial information.

On the other hand, On the other hand, (Chandrasekar et al., n.d.) classify blue-collar and violent crimes, and the auhors obtain better results using Support Vector Machine (0.8239% precision for violent class and 0.9602589% precision for Blue Collar class) than with Naive Bayes, Random Forest, Gradient Boost, for crime prediction

The authors of (Mahmud et al., 2021) work with data from the last three years of crime to analyze the crime rate of Bangladesh and, more precisely, to determine the safety paths by avoiding safety problems such as hijack, kidnapping, and harassment based on three targets attributes (Perpetrator Ages, Perpetrator Genders and Victims relation). Among Linear regression, Naïve Bayes and KNN algorithm, KNN obtains the greatest precision when evaluating accuracy (76.9298%). (Nath, 2006) exploits semi-supervised learning-based K-Means.

Different studies compare manual and automatic feature selection. For instance, (Sundhara Kumar & Bhalaji, 2016) compare manual selection with Boruta, while (Shojaee et al., n.d.) compare it to the Chi-Square method. Their results indicate that automatic feature selection promotes a better classification quality (98.66% precision with random forest based on Boruta, and Decision Tree based on Chi-Square reaches 85.7% precision) than manually (97.20% precision with RF and 84.6% precision with Decision Tree). Finally, only one study (Nath, 2006), employs method based on the expert's knowledge feature selection. It is important to consider expert knewledge when selecting the features, but automatic features selection allow to enhance the results on the model, and increase the classification precision.

In (Abdulrahman & Abedalkhader, 2017) authors compare Naïve Bayes Classifier and KNN. For KNN approach two distributions are tested, Uniform and Inverse distributions. For Naïve Bayes approach Bernoulli and Multinomial distributions are tested. It is concluded that Bernoulli and Multinomial approaches present the best results among applied techniques, when evaluating the results based on the log loss function. Additionally, the studies (Abdulrahman & Abedalkhader, 2017; Khatun et al., 2021; Yerpude & Gudur, 2017) benefit from cross-validation to obtain a more reliable estimate. The studies (Khatun et al., 2021; Yerpude & Gudur, 2017) exploit cross-validation to eliminate the chance of overfitting.

Among all the mentioned work, some studies manipulate fundamental and intuitive features: time, date, location, and crime type. Then, there might be information about the suspect(s) (identified or unidentified) and victim(s). Additionally, the authors of (Chandrasekar et al., n.d.) add demographical data (e.g. the mean income level of a neighborhood, racial diversity, Owner-Occupied Units, among others). Complementary (Munasinghe et al., 2015) add attributes including entry orientation, exit orientation and state of the property. Finally, Munasinghe et al. (2015) are interested in analyzing criminal groups based on the modus operandi attributes. For Munasinghe et al. the modus operandi can be characterized by twelve well-chosen features. The twelve features are clasified into three different types, MO defining attributes, MO supportive attribute and identification attributes. The modus operandi, in this case, is not an open text, but a specific set of values for the chosen features. One of objectives was to verfy if this approach is enough to capture the essence of the crimes and differentiate them from other types. The results indicates that the method presented relevant results in detecting the modus operandi of different groups.

Table 1 compares some of the most relevant works in crime classification, the parameters that are taken into account are: objective, used dataset, features used, target and main contributions. For the row features the used codes stand for:

F-SAN: Features used in the dataset Open Dataset San Francisco from SFPD

F-UCI: Features employed from the UCI

F-ADB: Features associated to the database.

Table 1. Summary of characteristics of each paper used in the literature

TITLE	**(Hossain et al., 2020)**	**(Sundhara Kumar & Bhalaji, 2016)**	**(Khatun et al., 2021)**	**(Shojaee et al., n.d.)**	**(Yao et al., 2020)**	**(Yerpude & Gudur, 2017)**	**(Abdulrahman & Abedalkhader, 2017)**	**(Nath, 2006)**	**(Chandrasekar et al., n.d.)**	**(Xuan et al., 2018)**	**(Mahmud et al., 2021)**	**(Munasinghe et al., 2015)**
AIMS	predict criminal activity, frequent crimes and rare crimes	predict the nature of a crime i.e. violent crime or a non-violent crime	discover relevant precautionary measures from prediction rates.	predict crime status	predict hotspot	predict the features responsible for causing crime in a region or locality	predict the classification of a crime based on time and location.	detect the crimes patterns and speed up the process of solving crime	predict category of crime	identify a fraud in time	assure a safety way to the destination in Bangladesh	identify a machine learning mechanism to list suspects based on MO
DATASET	open dataset from SFPD	Communities and Crime data from UCI	Criminal data: San Francisco, Chicago, Los Angeles	Communities and Crime data from UCI	open dataset from SFPD	Communities and Crime data from UCI	open dataset from SFPD	Criminal data	Criminal data: San Francisco Open Data & data from United States Census	e-commerce company in China	Criminal data	Criminal data from Criminal Records Division
FEATURES	F-SAN	F-UCI	F-SAN & other features	F-UCI	F-SAN	F-UCI	F-SAN	F-ADB	F-SAN + demographic features	F-ADB	F-ADB	F-ADB
TARGET	crime status	hotspot	Per Capita Violent Crimes	category of crime	hotspot/crime patterns	category of crime	credit card fraud	Perpetrator Age, Perpetrator Genders and Victims relation	listing criminals based on modus	crime status	hotspot	Per Capita Violent Crimes
CONTRIBUTIONS	Show the influence of the feature selection method on the results in the classification	Improvement of the results classification with random forests based on historical crime data and covariates (POI data and demographic information)	Influence of cleaned data over the classification	Study two techniques on KNN and three distributions on Naïves Bayes classifier.	(1) formulate crime pattern detection as a machine learning task (2) Introduce the concept of weighing for attributes	Find an approach to collapse the crime categories into fewer, larger groups to find structure in the data.	(1) Comparaison between two random forests (Random-tree-based and CART-based one) (2) evaluate the effectiveness of these two methods with real data.	Demonstrate the strength of K-nearest neighbor	(1) Shortlisting criminals based on their Modus Operandi features. (2) An algorithm can be selected to create clusters to imply an identifiable Modus Operandi. (3) Characterize the modus operandi in real databases (4) intention is to create MO based on a combination of attributes	Show the influence of feature selection method on the results in the classification	Improvement of the results classification with random forests based on historical crime data and covariates (POI data and demographic information)	Influence of cleaned data over the classification

DATASET DESCRIPTION

The work presented here is based on an anonymized dataset formed from real complaints received by the French Gendarmerie over 2020. The anonymization was made coherently, any element that could be considered personal data, e.g., names, addresses, telephone numbers, e-mails, among others. The first time a personal data is found, it is changed by a random value that follows the same format. For example, for a telephone, a random ten digits number, for an e-mail a user name followed by an "@" and a two or three set of words, separed by ".". However, from this time onwards, every time the first telephone appears, it is changed by the same value that was at first chosed randomly. The conversion keys are forgotten at the end of the anonymization process.

French Gendarmerie receives around 5000 complaints per day. These are written in French and, in general, are declarations from victims regarding criminally reprehensible issues, but can also be independently filled by police officers when these agents verify the need for it. The impact and sensitivity of each one of these complaints vary significantly. It goes from noise late at night to murder attempts, passing through car theft, domestic violence, and damage to property. All these complaints are of interest and need to be treated, even if these elements receive a standard main classification when filled. The main classification is completed manually and refers to the supposedly broken law. Moreover, this classification is stated by the police officer when the complaint is filed, with the elements that were available at that time. It means that voluntary, or involuntary, omissions from the declarant may impact the complaint's correct classification.

In other words, at least two human sources of bias exist and, therefore, may impact the reported complaint's classirication quality. The first is, how the police officer will transcribe the complaint on the system, and the second is the victim's possible amplified feelings. Each complaint receives at least one classification from the existing standard ones. The number of possible classifications is considerable, around 15,000 for the French Gendarmerie. Even though the attribution rules are clear and uniformized over all the French territory, in practice, the classification has a subjective component. It depends significantly on the interpretation and experience the police officer has in filing that specific type of complaint. From all the possible classifications, some are close and differ regarding criminal activity details. For example, for housebreaking, if it involves degradation, e.g., the door was broken, should receive one number, if there was no visible signal of force used, e.g., a lost key was employed, the classification value is another. It could be yet another if the entrance were through an open window. These are just some of the housebreaking classification variants, but this kind of option sets exists for almost all possible complaint types. Another example could be linked to street robbery, it can be simple, if it is a pickpocket,. It can be under constraint if the thief menaces the victim verbally and demands the wallet. It is also different if the thief has a gun. Moreover, each complaint may hold more than one infraction, which increases the range of possibilities even more. If the theft was under menace, but more than just verbally assaulting the aggressor, also physically assaulted the victim. This aggression should also be added as a second infraction, with a classification that considers the violence and the severity of the aggression consequences.

Other factors are likewise considered when creating the standard classification, for example, the intention of the suspect. Taking all these factors into account, even though well-defined rules exist, in practice, the same complaint may be differently classified depending on the police officer who received it. On top of that, even if the number of variants is considerable, the classification remains a broad classification, which considers only the most common criminal actions. New criminal activities, i.e. modus operands, take a noticeable amount of time to identify, verify, and receive their unique identification number.

Even though important and relevant, this initial classification is not extremely useful for specialized units. For example, metal theft is a particular kind of infraction, sometimes cars, battery, cable and catalytic converters are stolen, not because of the object itself, but for the metal contained. These may be in fact metal thefts, even if those are rarely recognized as such.

DATA PIPELINE

For LEAs, quality control, the traces of data sources and treatments are of utmost importance. These controls and traces provide the legal guarantees that the stored information can be used in court. If a bias influences the analysis results, this bias needs to appear and be registered. One of the biggest fears of LEAs is that analytical methods will produce results that seem relevant and useful, when these are not fair, because based on false or biased information.

From the data collection to the data analysis, different kinds of bias exist, for example, cognitive, stereotype and prejudice bias. When the officer fills in the complaint, his/her humor and his/her experience influence the redaction. Another bias that may critically impact the results classification is related to prejudices. If an officer makes a judgment for or against the victim, even if unconsciously, this bias may be incorporated into the complaint. That is why, it is essential for LEAs to keep track of the decisions made during the data pipeline. This section points out different challenges of different stages of ML pipelines, such as the management data, the data cleaning, the data augmentation, and the monitoring ML algorithms results. These challenges can be raised in the form of questions:

- **Question 1:** How to find and manage data to be consumed by ML models?
- **Question 2:** How to transform and clean the data as well as extract features?
- **Question 3:** How to serve ML models fast?
- **Question 4:** How to monitor and debug ML models?

This section covers the dataset pipeline used for the studied case: data collection, data elements and data Exploration, data cleaning, features engineering, features selection text field treatment.

Data Collection

This study was conducted with anonymized real data provided by Gendarmerie Nationale and the Central Office for the Fight Against Itinerant Delinquency, (*Office central de lutte contre la délinquance itinérante* - OCLDI). OCLDI holds an important position in fighting against criminality in France. The unit has several roles, including the monitoring of metal thefts.

This classification work was performed over an anonymized dataset of filled complaints (CRPJ) deposited at French Gendarmerie. The Center for Data Science (CSD) in the Gendarmerie Nationale receives ~1.8 Million complaints per year. These original complaints are a continuous flow of information that arrives at any time, all year round (24/7/365). CRPJs vary significantly in the level of information present and severity of the complaints. Moreover, this data is real data, with all the advantages and disadvantages real data have. Raw data is often imperfect, inconsistent, and redundant. On top of that, different complaints have different fields, meaning that only a subset of the available fields is common for all the CRPJs.

Data elements and Data Exploration

The raw complaints dataset had 43 feature columns, of which many rows had missing values. Each entry is unique and represents a complaint that is created as soon as the complaint is registered. An entry is composed of different fields including fields related to the creation of the complaint and open text descriptive fields, i.e. describing the fact committed. Some fields are simple and some complex, as example of simple fields, we can highlight the crime id, action_taken, report_date, unite_of_register, modus operandi and type_investigation, all these are, at first, string values. Complex fields are fields that group pieces of information together i.e. victim(s), infraction(s), suspect(s), damage, vehicle. These have, by objective, create a context for the fact described in the complaint. For example, the infraction date and time, the location, and the type of the place the infraction happened. Here, is a complaint illustration with the type of the fields:

- **modus operandi**: text field - In a law enforcement context, a modus operandi refers to a criminal's typical mode of operation and ways of acting.
- **Victim**
    - **birth date**: text
    - **birthplace**: text

  - **sex**: text
  - **name**: text (anonymized)
  - **telephone**: text (anonymized)
  - **nationality**: text
  - **residence geocode**: (anonymized)
  - **residence city**: text
  - **residence postal_code**: text
  - **job**: text
- **Infraction**
  - **Type place**: text
  - country: text
  - **natinf**: text that represents the nature of the infraction. It designates a numerical code that classifies the crime. Each natinf has its own definition and numerical identity.
  - **Geocode**: text (anonymized)
  - **Kind of infraction**: text
  - **start time of the offence**: text
  - **end time of the offence**: text
  - **start date of the offence**: text
  - **End date of the offence**: text
- **Suspect**: the fields between those of the victim and the suspect are identical type

The full dataset consists of around 5 million entries. Among these 5 million complaints, 16% present more than two offenses, 6% have more than two victims, and 3% have more than two suspects registered. For the fundamental characteristics, authors denote that 0.13% of the start date of the offense are incorrectly filled, or the value is missing. For the end date of the offense, this value is 18.17%. Apart from the end date of the infraction, these values are relatively small.

On the other hand, the percentage of missing entries is 63%. However, this is understandable, as not always suspects are identified when the complaint is filled. Complaints with missing victims exist but are rare. However, some of the fields may be missing, 21% of victim sex, victim date of birth, and victim job are erroneous or missing values.

Data Cleaning

Working on real data presents benefits and drawbacks. Working with real data provides information on concrete experience, real significant elements may be pointed out. However, working with real data may be messy. Real data may contain entries with missing and inconsistent data due to filling and parsing errors. It is essential to reduce some noises, incomplete and inconsistent data with data cleaning. The work of Yerpude & Gudur, (2017) shows that data cleaning influences the classification results. This work also confirms a better performance with cleaned data, F1-score of 86.54%, while the classification on non-cleaned data, the F1-Score was 84.80%.

The first step of the data cleaning process is treating missing values and redundant registries. Regarding the missing values, different methods exist. Some autors suggest discarding the records that contain the missing values, others advise changing misplaced or missing values by median or mean (Khatun et al., 2021; Yerpude & Gudur, 2017), automatically correcting the data with default values. Regarding the redundancy, the data cleaning process may try to find the repetitions and remove the redundant elements.

The second data cleaning step involves handling records containing missing values, such as omitting the incorrect fields. In this study, the authors decide to treat the missing data by replacing it with a standard and neutral value.

For the last data cleaning step, a data type transformation on several fields is made. Date fields, time fields, text fields, geo-code fields are transformed to more specific types. Initially, the date fields (date of infraction, birth date victim/suspect) are considered to be strings. Dates are transformed in date types, time are transformed in time values, and numerical fields become numerical. Concerning the geocode fields, these fields are initially considered in string and transformed into tuple of float. Then, the authors analyze the geocode with two decimal places in order to expand the position. In other words, the aim is to distinguish the position of one large city from a neighboring large city and separate one village from the next.

Features engineering

Some data may not be extremely useful for classification since these are either too precise or will never repeat in the prediction time (date and time of infraction). Regarding precise information, a good example is the age of the victim. A larger interpretation may be reached if the victims' ages are grouped into categories, e.g., young, teenager, adult, and senior. For age, if the person is between 0-12 years old, the person is labeled as child, between 13-18 as young, 19-25 as young adult, 26-50 as adult, 51-60 as old adult, and superior to 61, as senior person. In the features engineering process, new columns are created to extend the explainability of the initial data.

Example for victim data:

- age: 50
- age categories: adult

The postal code of the place of the crime, of the residence of the victim/suspect are treated to create the following columns: department and region. The distances between the incident and the residence of the victim and suspect are classified into close, average and distant. If the distance is inferior to 10 km, it is labeled as close, between 11-40 as average, and above 40 km, it is labeled as distant.

Example for infraction data:

- code postal: 58110
- geocode: 47.03 , 3.57
- department : Nièvre
- region: Bourgogne
- id dept: 58
- id region: 26
- date start of infraction: 03/04/2020
- date end of infraction: 04/03/2020

Example for victim data:

- code postal: 58110
- geocode: 47.03 , 3.57
- department: Nièvre
- region: Bourgogne
- id dept: 58
- id region: 26
- date of birth: 05/05/1965
- place date of birth: Orléans, France

For the date fields, the authors extend the dimensions to month, day of the week, day of the month (1-31), day of the year (1-365), season, state of the day (day, night), day moment (morning, afternoon, evening, night) and week-related phase (week or weekend).

Text field treatment

For LEAs text fields are an essential source of information. It presents a considerable quantity of specific information about the crime and the specific characteristics of each one of the individual events. Exactly because these text fields are designed to convey specific information, these fields cannot be standardized, and thus their treatment is not taken into account by regular methods. Even if hard to treat, these fields express specific methods and patterns that may help LEAs to better understand criminality.

For this reason, this work considers that open text fields, such as the modus operandi, should be effective in improving classification results and must be considered as feature. For example, on the dataset used in the experimentation part of this article, for many entries, the correct classification can only be reached if the open text fields are considered. The integration of the modus operandi information is designed through a topic modeling technique. A new column called "modus topic" was created, where for each complaint, a topic is assigned. During the training, the modus operandi fields associated with a given class are collected and analyzed.

The transformations applied were lower case transformation, punctuation removal, tokenization, lemmatization, stop words removal and small words removal (words smaller than three characters are considered to have comparatively little value for the classification). After these treatments, the ten more frequent words in different documents are considered the keywords that better represent that class. Figure 1 and Figure 2 represent the words cloud of different classes in the experimentation dataset. For precious metal theft, one can observe that the words "*vehicule*" (French for car), "*stationné*" (stopped) and "*échappement*" (exhaustion pipe) are some of the most common words. That makes sense, as one of the most common ways to still precious metals today is to target the exhaustion pipes of cars. Inside these car parts, one can find palladium, a valuable metal. So, it is coherent that the result provides a vehicle in the top words for precious metal theft.

This work considers only the modus operandi as a feature. No other open text field on the experimentation dataset had enough variability to be considered of interest.

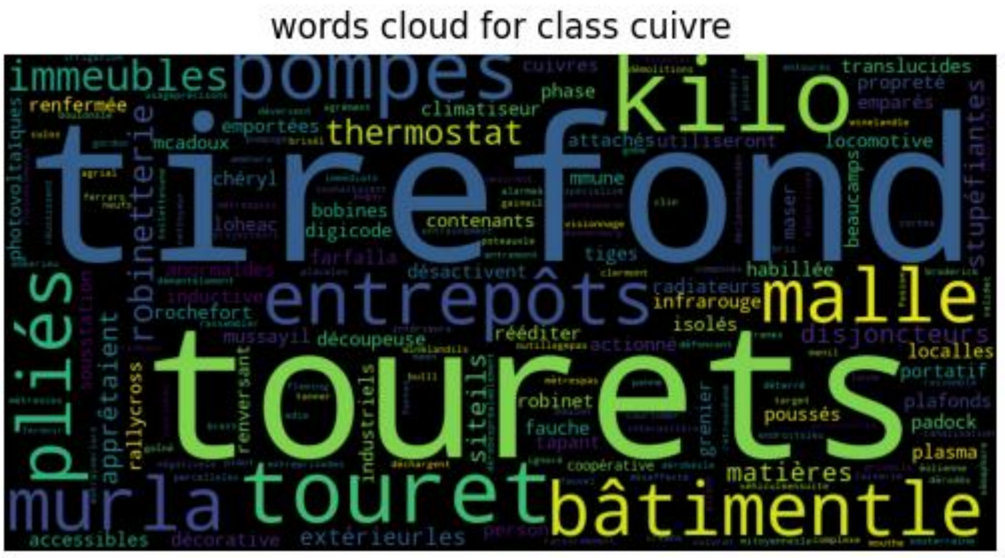

Figure 1. illustrating words cloud for class "cuivre" obtained through topic modelling technique

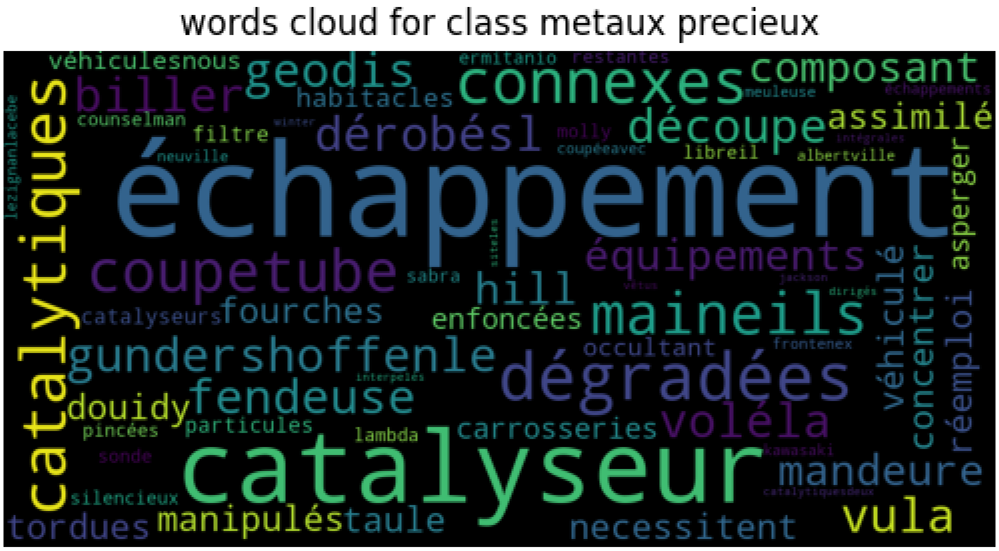

Figure 2. illustrating words cloud for class "métaux précieux" through topic modelling technique

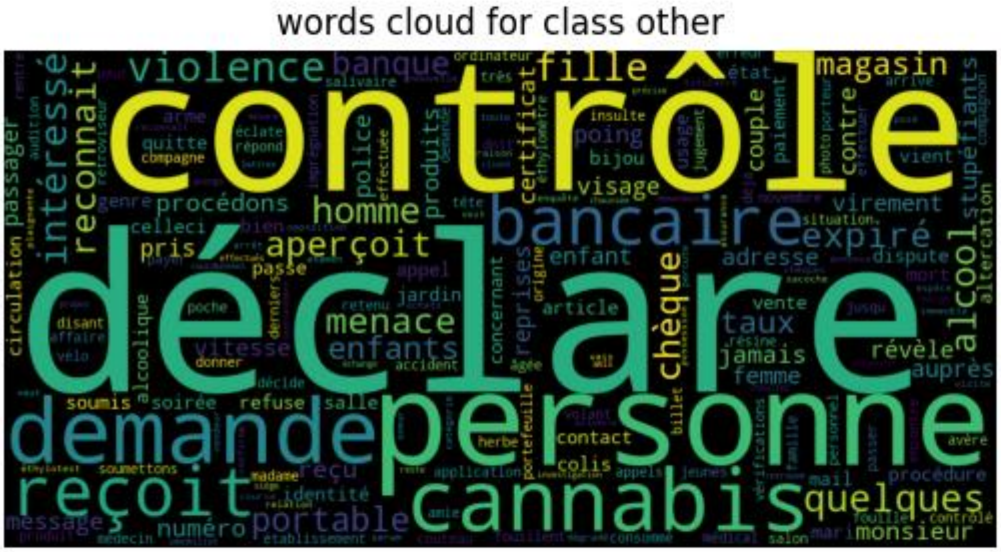

Figure 3. illustrating words cloud for the class "no-metal" through topic modelling technique

Classification methods

Now that the features are defined, the classification method more adapted to the LEA-specific characteristics needs to be chosen. Before choosing a given algorithm for classifying the LEA data, this work compares different classification algorithms regarding their performance and main characteristics. The compared methods are Random Forests (RF), Naive Bayes Classifier, Gradient Boosting Machine (GBM) and Deep Learning. The target function is a binary classification between metal and no-metal theft. Since explainability is a major point for LEAs, explainable algorithms such as Random Forests, Naive Bayes and Gradient Boosting are preferable to less explainable ones such as deep neural networks. However, if the classification performance of the latter is considerably better, LEAs may choose to use it regardless the difficulty of explaining the results. For some tasks the negative impact of wrongful classifications may surpass the benefits of explainability.

The comparisons were made using the Python wrapper for H2O algorithms implementations (H2O.ai, h2o: Python Interface for H2O, 2021) H2O is an efficient open-source coding tool for data mining and machine learning algorithms, that implements a series of different machine learning algorithms. The following sections will present explain, at a high level, the main principles of the compared classification algorithms

Random Forests

This method consists in creating set of independent decision trees and using several estimators. Each tree represents a fragmented view of the problem, but by aggregating the partial view of each one of the trees, we will obtain a more general view. The method relies on a "group wisdom" approach. It is expected for a group of individual opinions to show better accuracy than a single one. Random forests (Breiman, Random Forests, 2001) relies on the correlation between different trees achieved by tree bagging and feature sampling. Random forests is also robust to noise and outliers since it relies on various tree views, as it analyzes a problem from different angles and relies on the majority's point of view. In addition, this method may handle categorical data, and LEAs dataset may contain several categorical features. The main drawback of random forest is the memory required to store all the trees and the inference time that increases with the number of trees. The results presented here are from the H2O RandomForestEstimator implementation (H2O.ai, h2o: H2ORandomForestEstimator, 2021)
The main parameters used for the experimentations are :

- ntrees=128 that represents the number of created trees
- max_depth=35 that the maximum tree depth for each tree
- stopping_tolerance=0 gives the relative tolerance for the metric to stop the training if there is no improvement (when the improvement is inferior to this given value)
- stopping_rounds=300 allows stopping training when the option selected for *stopping_metric* does not improve
- score_each_iteration=True
- score_tree_interval=0
- learn_rate=0.01. specifies the learning rate between 0.0 and 1.0.

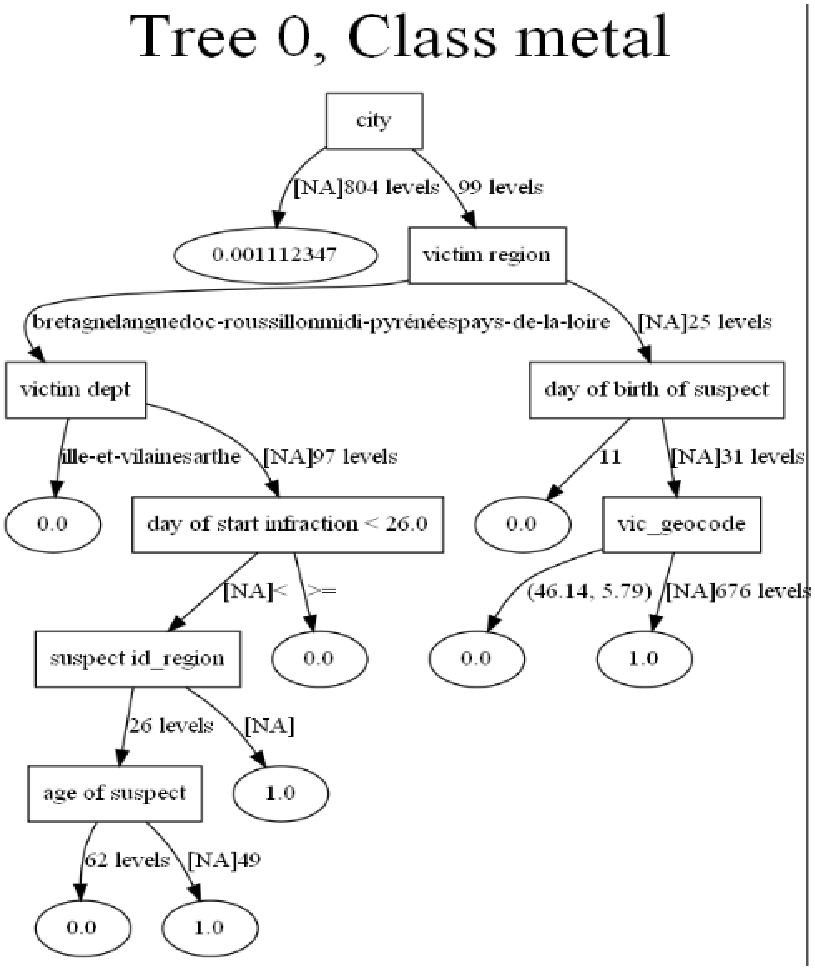

Figure 4. presents an example of a tree for classifying metal theft

Gradient Boosting

Gradient boosting (GBM), as originally introduced by (Friedman, 2001), is a method recognized for its accuracy and its speed. GBMs have demonstrated relevant success in various machine-learning and data-mining challenges (Bissacco et al., 2007; Hutchinson et al., 2011). It relies on the intuition that the best possible next model minimizes the overall prediction error when combined with previous models. The name gradient boosting comes from the way the target functions are organized in a gradient of the errors, regarding the expected prediction. Each new model should go a step further in the direction that minimizes prediction error. GBM algorithm uses the log loss function to measure how good the coefficients are for predicting in the given data. Its work is based on sequential work, this means the algorithm creates multiple weak models and each model learns from the mistakes linked to the previous model. The final classifier that becomes stronger as it combines the models built along the way. H2O's GBM sequentially builds regression trees on all the dataset features in a fully distributed way, i.e., each tree is built in parallel.

GBM is set with the same parameters used for Random Forests for the experimentation section. For, H2OGradientBoostingEstimator method, the main parameter values are: ntrees=128, max_depth=35, stopping_tolerance=0, stopping_rounds=300, score_each_iteration=True, score_tree_interval=0, learn_rate=0.01. By default, the nfolds parameter for the number of folds for cross-validation is set at 0, the min_rows considering the minimum number of observations for a leaf sets at 10, and the distribution is set at AUTO.

Naïve Bayes Classifier

Naïve Bayes (NB) Classification is based on Bayes theorem that describes the probability of an event, is given by:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where $P(A|B)$ is the probability of A given B is true, $P(B|A)$ is the probability of B given A is true $P(A)$ and $P(B)$= Probability of A and B respectively. A stands for the target/output and B for the remaining features.

This algorithm assumes that all predictors are independent, which means that the presence of one specific feature in a class does not influence the presence of another one. Naïve Bayes is a supervised learning algorithm and has several advantages, the first on ist that NB classifiers are fast. Moreover, this algorithm is easy to use, interpret and may express prior expert knowledge (Zhang & Li, 2007). However, as previously stated, this classifier assumes that all predictors are independent, which may not hold in real life, which leads to possibly biased estimators.

For the experiementation section the H2O implementation, H2OnaiveBayesEstimator, is used. It is based on Gaussian distribution of numeric predictors with mean and standard deviation computed from the training dataset. The nfolds parameter is set to 0, the distribution on AUTO, the balance_classes is not enabled, and the laplace parameter is 0.

Deep Neural Networks, or Deep Learning

This classifier is based on a multi-layer feedforward artificial neural network trained with stochastic gradient descent using back-propagation. Simple estimators, perceptrons, are organized in layers, and the output of one layer is directed as input of the next one. Each perceptron receives, apart from the stimuli from the previous layer, a bias and a weight that are evaluated by an activation function. The result of the activation function formula is then propagated as the result of this perception to the connected perceptrons of the next layer. Even if based on relatively simple structures, neural networks are able to model complex real-world relationships. Moreover, this network is robust to imperfectness.

The main drawbacks linked to deep neural network approaches are the quantity of data required for training and their lack of simple ways to explain their results, at least from the point of view of us, humans. It is relatively hard to explain the logic used to reach a conclusion in layman's terms.

By using the implementation on H2O, the H2ODeepLearningEstimator, the values used for main parameters are:

- epochs=1000, the number of times the train will pass over the available dataset
- train_samples_per_iteration=-1, the amount of data (all available data) for training for each step
- stopping_rounds=300 allows stopping the training when there is no improvement over that number of iterations

By default, the hidden layer sizes is set to (200,200), the adaptive_rate is enabled, the rho is set to 0.99, the epsilon to 1e-08, input_dropout_ratio at 0 allowing to improve the generalization with the input layer dropout, L1 regularization at 0 and L2 regularization at 0.

EXPERIMENTATIONS

Experiment setup

The presented experimentations results were performed in a machine with an iCore 5, 2.40 Ghz CPU, with 16.0 Go RAM. The data pipeline was implemented in Python 3.8 with the (H2O.ai, h2o: Python Interface for H2O, 2021). In order to deal with the text fields, nltk version 3.7 (Bird, Klein, & Edward, 2009) was used.

Dataset experiment generation

The full dataset consists of anonymized complaints, from all France (metropolitan France and overseas departments) between 2016 and 2020. The complaints are written in French and are composed of 55 Categorical fields, 33 numeric, 4 date, 8 text fields after data pipeline process. For this experiment, the data were divided into training, validation, and testing datasets. For each experiment, S different samples were gennerated, where S = {0, …, N} and N = 32. For each sample:

- **Train dataset** : which is built with X number of samples with two on the class of interest (i.e metal theft and non-metal theft).
- **Test, validation, and training** : are composed of distinct elements.

Evaluation metrics

The objective of this section is to compare the overall performance of the different methods. The chosen metric for the comparison is the F1-score, as it considers both precision and recall. Precision, recall and F1-Score metrics are defined as follows:

- **Precision**: refers to the number of individuals correctly assigned to a class i, true positives, compared to the total number of individuals predicted as belonging to class i (correctly predicted, and wrongly predicted, false positives).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

- **Recall**: The recall represents the number of individuals correctly assigned to class i compared to the total number of correct predicted individuals and the wrongly assigned as negative, false negatives.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- **F1 score** : provides a way to combine both, precision and recall, into a single measure that captures the influence of both properties. F1-score suggests that both precision and recall are important.

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Comparison of classifiers Classification methods

This section presents the performance of the classifiers that we described in section Classification Methods. The objective is to compare the performance of Gradient Boosting, Deep Learning, Naive Bayes and Random Forests classifiers in classifying complaints.

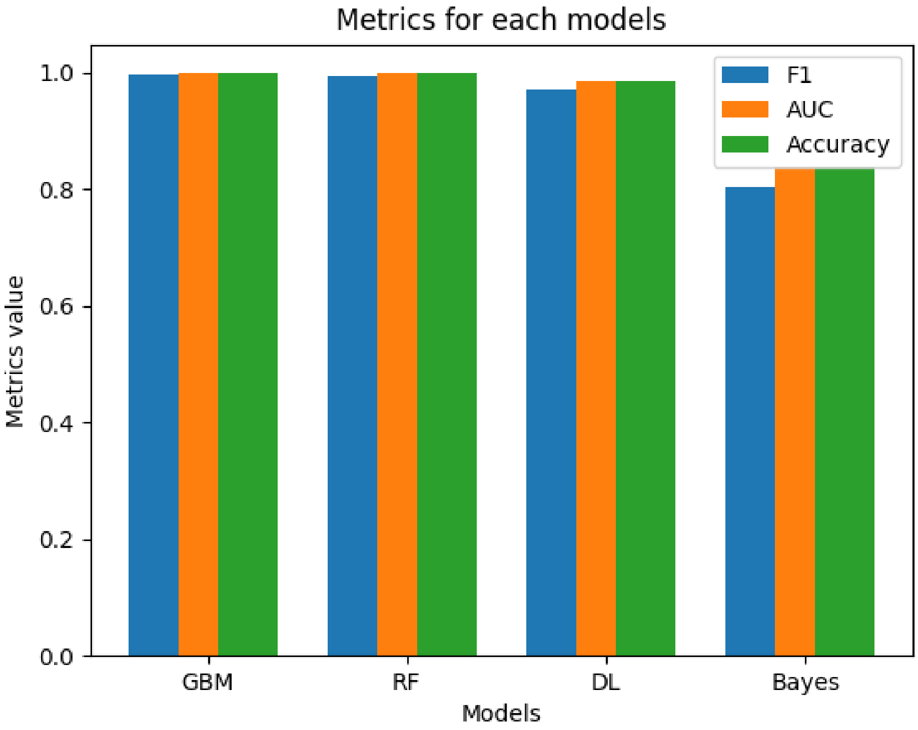

Figure 5. shows the F1-score for different machine learning methods

Figure 5 shows that, apart from Naïve Bayes with Gaussian distribution, the other three methods present similar results, above 90% performance, with Deep learning presenting a slightly worse performance. The relatively poor performance of Naïve Bayes suggests the data does not follow the expected distribution and data was discrete and not continuous. As expected, deep learning, Gradient Boosting and Random Forest models give satisfying F1 results. The presented results consider all the available fields, including the topic modeling Modus Operandi, described in Section Text field treatment. Three topics are considered for metal specific theft, copper, precious metals, lead-tin. All other complaints are classified as non-metal thefts. For each topic modeling nltk is used to convert text in lower case, remove punctuation, tokenize, lemmatize, remove stop words and filtering out smaller than 3-character words as explained in Section Text field treatment. The set of words for each set is unique and distinctive

Explainability is also an essential aspect of the method employed. The legal system requires the investigative process to be auditable and explainable. From the presented method Random Forests, Naïve Bayes and Gradient Boost, allows for results explainability. As, apart from Naïve Bayes, the results for this experimentation are equivalent for all the other methods, and Deep Neural networks are harder to explain. That is why, the next experiments will focus only over Random Forests.

Classification methods, in general, are not fit for using text fields. However, experts' knowledge suggests that the Modus Operandi may contain important pieces of information to be considered for classification. The intention of the next experiments is to verify that this expert knowledge, and if it is proven that the Modus Operandi may positively influence the results, which is the impact of this influence. Two experiments are conducted in order to investigate on the Modus Operandi impact, one using the Modus Operandi and the other not. In order to compare the difference between using or not using the modus operandi, both experiments were performed with the same dataset. The training set is designed with 1000 rows whose 500 targets and for the validation set, 500 data rows whose 100 are related to the target (metal theft). However, the number of features changes, Experiment 1 ignores the Modus Operandi, Experiment 2 considers it.

Experiment 1

The first experiment investigates two aspects linked to Random Forests as a viable classification method. First it evaluates the impact of the number and depth of trees on the classification performance. The second aspect observed is the limits of classification using only structured data. This baseline is crucial for understanding the impact of the analysis of open text fields on the criminality classification, as discussed in the second experiment. Regarding the optimal number of trees for random-forest-based methods, this experimentation validates the work of Oshiro et al. (2012), which targets medical datasets. Analyzing the performance of Random Forests in a series of datasets, Oshiro et al. found that the optimal value for the number of trees to be between 64 and 128. One of the objectives of this experiment was exactly to verify if these values also hold for criminality data classification.

Figure 6 presents an overview of F1 score evolution, taking the tree number and depth into account. The F1 score, presents small differences for the observed parameters. Broadly it varies between 0.810 and 0.835. The increase in performance is clearly linked to the number of trees, but the gain is not proportional. This confirms the conclusions reached by Oshiro et al., (2012) regarding a practical limitation in the performance reached by increasing the number of trees. Even if small, in terms of raw value, a visible difference exists in the F1-score between nb_trees 20 and 50. A loss of perrformance exists when forest has less than 50-64 trees. In the other sense, the gains in terms of F1-score, are not extremely significant when the trees number increase (from 100 up to 400 trees).
For the experiment with 128 trees, the system F1-score was 0.828, increasing the number of trees to 300, the value of F1 score raises to 0.832 a 0.4% improovement. It is apparent from both Table 2 and Table 3 that with high levels of tree numbers, the system does not gain significant performance. In contrast, regarding the depth, an optimal value, for the various sizes of trees is reached on depth 20. Table 2 and Table 3 present the evolution values for depth 20 and 35, for comparison. For an identical number of trees (nb_trees = 64), F1 score is at 0.820 with depth = 20, and F1 at 0.813 with depth = 35.

Table 2. illustrates the F1 evolution for depth = 20

F1 Evolution for depth = 20	
Nb Trees	**F1 Score**
20	0.8139759026387726
50	0.8258543201823993
64	0.8308103915014239
100	0.8293691531176794
128	0.8283594052749252
250	0.8329929761166028
300	0.8301615327671789

350	0.8320451566180628

Table 3. illustrates the F1 evolution for depth = 35

F1 Evolution for depth = 35	
Nb Trees	**F1 Score**
20	0.8307435635700084
50	0.823772784054474
64	0.8206252334898942
100	0.8296962774948098
128	0.8306640915022648
250	0.8337800809547501
300	0.8321154594465414
350	0.8119464350846022

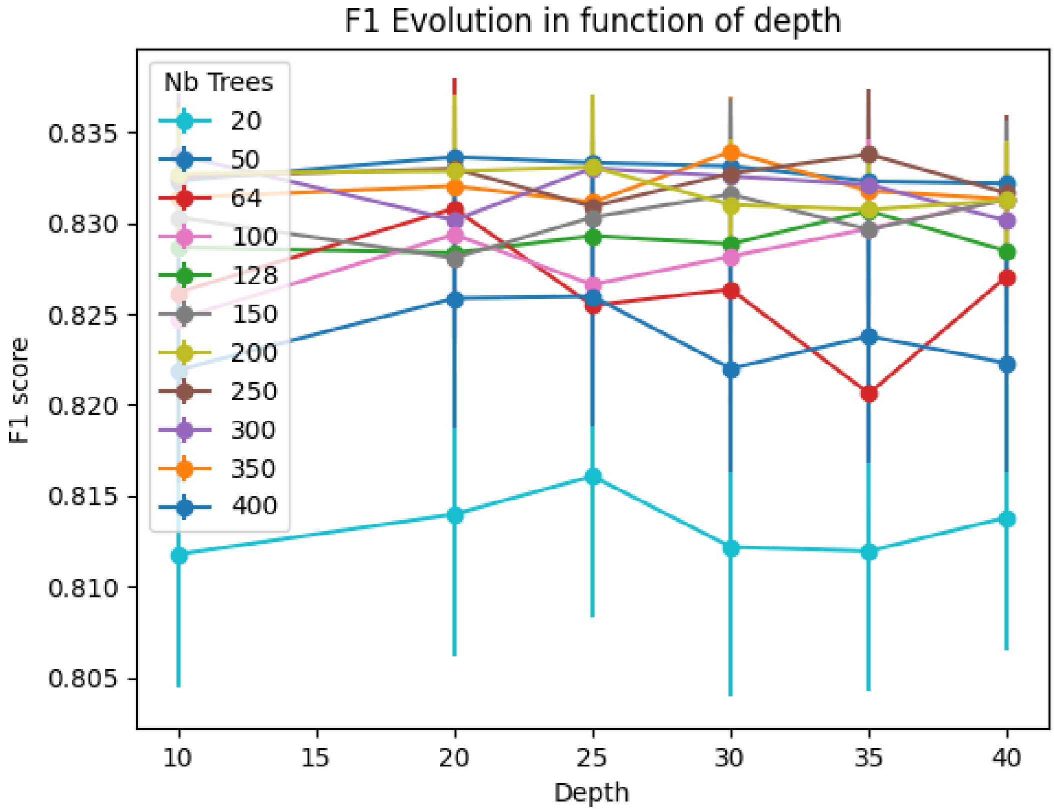

Figure 6. presents the F1 Evolution for Random Forest - Parameters for the validation : NbValid : 500 whose 100 interesting - without modus operandi topic

Overall, what emerges from the graphs is that the basic and elementary characteristics, namely, location of the crime incident, the date, hour and so on relay the performance benefits of the classification.

Experiment 2

The main focus of this experiment is to evaluate the impact of the modus operandi over the classification performance. This experiment considers the modus operandi topic obtained previously with the text processing method described in Section Text field treatment. The use of the modus topic column is the only difference between the results of Experiment 1 and Experiment 2. Table 4 shows the F1-score, when considering modus operandi, over depth 20 and varying the number of trees. Figure 7 presents significant increase in performance when using topic modeling to assess the modus operandi on the classification. The F1-Score performance that was around 0.83 increased to around 0.95, an increase of more than 10% in performance. What confirms the importance of the modus operandi for the classification of complaints.

Consistently with the results of Experiment 1, the performance increased with the number of trees, but the difference beyond 64 trees becomes even smaller, with 128 and 300 trees presenting equivalent best performance values. Considering the memory usage and the prediction time that increases with the number of trees, a model with 128 trees would be preferable over one with 300 trees. The experiment shows that a number of trees between 64 and 128 is sufficient to obtain an accurate performance, what is also coherent with the study presented in (Oshiro et al., 2012).

Nb Trees	F1 Score
64	0.9377127971535355
128	0.9504709478303515
200	0.949383276919891
300	0.9505756323065744
350	0.9496010639797149
400	0.9483078024986719

Table 4. F1 Evolution for different trees numbers with fixed depth = 20

The graph, presented in Figure 7 (Munasinghe et al., 2015), is coherent with the one presented in the previous experiment. It is possible to conclude that the depth does not present a significant impact, being 20 levels a good performance tradeoff. Based on the same parameters, for example depth = 20 and tree numbers = 128, an increase is noticed as the F1 score in the first experiment is 0.831 and 0.950 in the second one. This underlines how important the modus operandi is for the classification. Indeed, the intuition that Modus Operandi conveys relevant information, that is not captured by the other structured fields is confirmed.

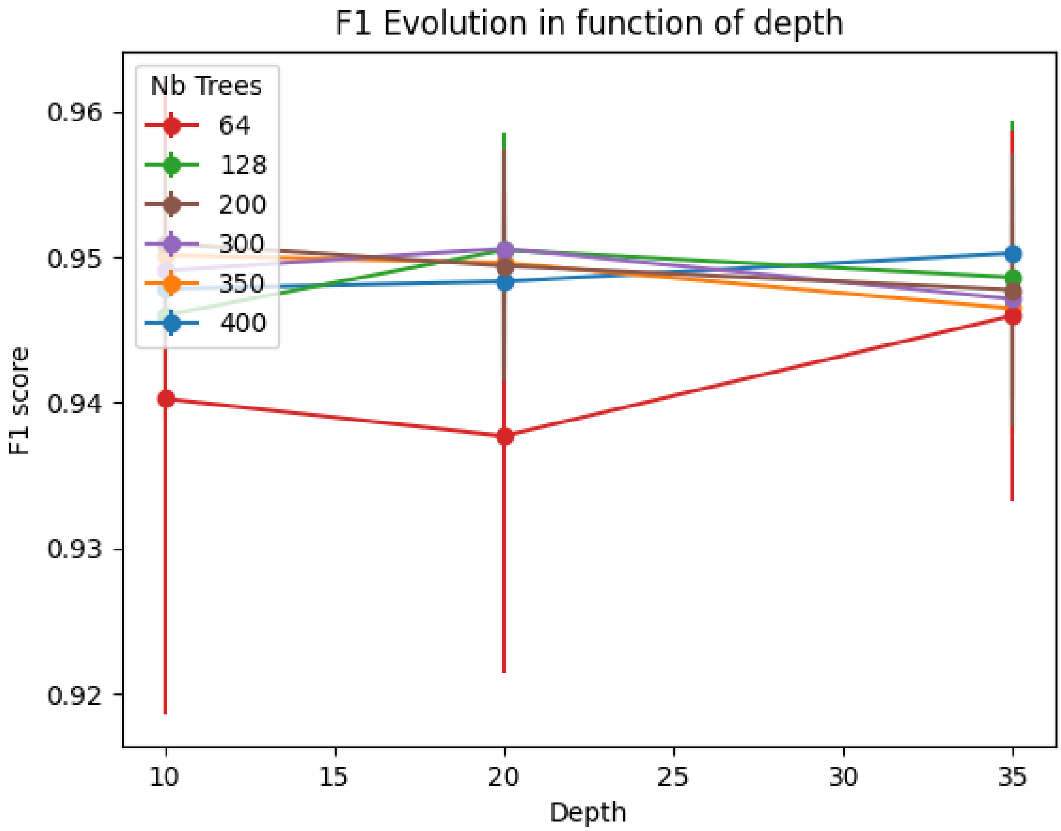

Figure 7 illustrating F1 Evolution for Random Forest - Parameters for the validation : NbValid : 500 whose 100 interesting wiith Modus Operandi topic

CONCLUSIONS AND FUTURE WORKS

The initial objective of this chapter was to evaluate how precise standard classification methods can be for the categorization of criminal-related data. Law enforcement agencies receive a large quantity of data and cannot manually organize everything. However, a characteristic of LEAs data, is that a considerable part of the knowledge is conveyed in open text fields. The second intention here is to understand if a simple topic modeling technique can be used to encode the knowledge on open text fields in a way that classification algorithms may treat. The experiments affirm that it is important to take into account the modus operandis in the classification. Thus the information in these types of fields is not fully represented by other structured ones. This result is also consistent with the way experts classify the received data. They consider standard fields, but the modus operandi is considered fundamental to organize the received complaints correctly. The experimentations, focus on metal theft data classification, but the results and the method are extensible to other types of crimes.

These results also confirm the study presented at (Munasinghe et al., 2015) that has also noted the importance of modus operandi in tracking the criminal. However, this MO form is not organized in a way that is well categorized. The results presented here highlight the importance of the modus operandi in the organization of criminal data. The modus operandi is one of the only fields where agents may express openly the way the crime has happened. The authors add a level of information to the basic ones (kind of crime, the location, anonymized information related to the victims/ suspects). Another important conclusion directly linked to that is that this kind of field needs to be correctly filled. Empty modus operandi, or filled with standard texts, may not improve the overall criminality understanding. More sophisticated methods may be used to extract even more knowledge from the open-ended fields, but these fields need to be correctly filled.

Another important finding is linked to the performance of the studied methods. Different methods present similarly good performances when classifying complaints. However, LEAs need explainable methods, and here it is shown that hat explainable methods perform similarly well to less explainable ones. Regarding performance issues, the study also puts in evidence the limits of the increasing in the number of trees and depth for random forest. The best tradeoff is between 64 and 128 trees, with a maximum depth of around 20 levels.

Even if this study shows the importance of using open text on the classification, by no means it define the boundaries of this interest. More detailed and detailed studies are required to understand the full impact of the treatment of these fields, regarding the classification. More sophisticated treatments, may improve even more the classification results. For example, the modus operandi could be modeled with sentence embeddings to convey its semantics. Sparse embedding vectors, may better capture subtitles semantic differences. These vectors can be used to improve the quality of the topic modeling, or be used directly in the classification, as these are in fact numerical values. Another possibility is to investigate how the text fields evolve over time to detect and understand the development of different topics.

ACKNOWLEDMENT

This research has been funded with support from the European Commission under the H2020-EU.3.7.1 and H2020-EU.3.7.8 project with Grant Agreement 833276. This publication reflects the views only of the authors, and the European Commission cannot be held responsible for any use which may be made of the information contained therein.

BIBLIOGRAPHY

Abdulrahman, N., & Abedalkhader, W. (2017). KNN Classifier and Naive Bayse Classifier for Crime Prediction in San Francisco Context. *International Journal of Database Management Systems*, *9*(4), 1–9. https://doi.org/10.5121/ijdms.2017.9401

Bird, S., Klein, E., & Edward, L. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* O'Reilly Media, Inc.: O'Reilly Media, Inc.

Breiman, L. (1984). Classification And Regression Trees. doi: https://doi.org/10.1002/cyto.990080516

Breiman, L. (2001). Random Forests. *Machine Learning, 45,* 5-32. doi:https://doi.org/10.1023/A:1010933404324

Chandrasekar, A., Raj, A. S., & Kumar, P. (n.d.). *Crime Prediction and Classification in San Francisco City*. 6.

Clarke, R. V. (2005). *Crime Analysis for Problem Solvers In 60 Small Steps.* Washington, DC: U.S Department of Justice, Office of Community Oriented Policing Services. Récupéré sur https://cops.usdoj.gov/RIC/Publications/cops-w0047-pub.pdf

Friedman, J. H. (2001). Greedy boosting approximation: a gradient boosting machine. *The Annals of Statistics, 29*(5), 1189-1232. doi:10.1214/aos/1013203451

Hossain, S., Abtahee, A., Kashem, I., Hoque, M. M., & Sarker, I. H. (2020). Crime Prediction Using Spatio-Temporal Data. In N. Chaubey, S. Parikh, & K. Amin (Eds.), *Computing Science, Communication and Security* (Vol. 1235, pp. 277–289). Springer Singapore. https://doi.org/10.1007/978-981-15-6648-6_22

H2O.ai. (2021). *h2o: H2ORandomForestEstimator.* h2o. Récupéré sur https://docs.h2o.ai/h2o/latest-stable/h2o-py/docs/modeling.html#h2orandomforestestimator

H2O.ai. (2021). *h2o: Python Interface for H2O.* h2o: Python package version 3.32.0.5. doi:https://github.com/h2oai/h2o-3

Khatun, Most. R., Ayon, S. I., Hossain, Md. R., & Alam, Md. J. (2021). Data mining technique to analyse and predict crime using crime categories and arrest records. *Indonesian Journal of Electrical Engineering and Computer Science*, *22*(2), 1052. https://doi.org/10.11591/ijeecs.v22.i2.pp1052-1060

Mahmud, S., Nuha, M., & Sattar, A. (2021). Crime Rate Prediction Using Machine Learning and Data Mining. In S. Borah, R. Pradhan, N. Dey, & P. Gupta (Eds.), *Soft Computing Techniques and Applications* (Vol. 1248, pp. 59–69). Springer Singapore. https://doi.org/10.1007/978-981-15-7394-1_5

Munasinghe, M., Perera, H., Udeshini, S., & Weerasinghe, R. (2015). Machine Learning based criminal short listing using Modus Operandi features. *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 69–76. https://doi.org/10.1109/ICTER.2015.7377669

Nath, S. V. (2006). Crime Pattern Detection Using Data Mining. *2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops*, 41–44. https://doi.org/10.1109/WI-IATW.2006.55

Oshiro, T. M., Perez, P. S., & Baranauskas, J. A. (2012). How Many Trees in a Random Forest? In P. Perner (Ed.), *Machine Learning and Data Mining in Pattern Recognition* (Vol. 7376, pp. 154–168). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31537-4_13

San Francisco Police Department. (s.d.). https://www.sanfranciscopolice.org/stay-safe/crime-data/crime-dashboard

Shojaee, S., Mustapha, A., Sidi, F., & Jabar, M. A. (n.d.). *A Study on Classification Learning Algorithms to Predict Crime Status*. 10.

Sundhara Kumar, K. B., & Bhalaji, N. (2016). A Study on Classification Algorithms for Crime Records. In A. Unal, M. Nayak, D. K. Mishra, D. Singh, & A. Joshi (Eds.), *Smart Trends in Information Technology and Computer Communications* (Vol. 628, pp. 873–880). Springer Singapore. https://doi.org/10.1007/978-981-10-3433-6_104

UCI Machine Learning Repository. (s.d.). https://archive.ics.uci.edu/ml/datasets/communities+and+crime

Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., & Jiang, C. (2018). Random forest for credit card fraud detection. *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, 1–6. https://doi.org/10.1109/ICNSC.2018.8361343

Yao, S., Wei, M., Yan, L., Wang, C., Dong, X., Liu, F., & Xiong, Y. (2020). Prediction of Crime Hotspots based on Spatial Factors of Random Forest. *2020 15th International Conference on Computer Science & Education (ICCSE)*, 811–815. https://doi.org/10.1109/ICCSE49874.2020.9201899

Yerpude, P., & Gudur, V. (2017). *PREDICTIVE MODELLING OF CRIME DATASET USING DATA MINING*. 16.

Zhang, H., & Li, D. (2007). Naïve Bayes Text Classifier. *2007 IEEE International Conference on Granular Computing (GRC 2007)*, (pp. 708-708). doi:10.1109/GrC.2007.40