GDPR Personal Files Scanner - Automatic classification of files in accordance with the GDPR

Pedro José Soles Moura
Instituto Superior Técnico
Lisbon, Portugal
Email: pedrojmoura@tecnico.ulisboa.pt

***Abstract*—The General Data Protection Regulation [1] (GDPR) is the European regulation on the protection of natural persons with respect to the processing and free movement of personal files and was fully enforced in May 2018.**

According to chapter 2, article 5, paragraph 2 of the GDPR, controllers are accountable for all the life cycle of the personal data collected so far.

The auditing and further accountability of the data transfer requires the development of tools that track document and data exchanges done by the various actors of the process.

This thesis presents the first system that automatically processes documents and determines if such documents contain information that can be considered personal in the light of the GDPR. It uses Decision Trees, complemented by a series of heuristics for the creation of feature vectors. To train the system a new data-set of documents was developed. These documents are either synthetic of real and represent various classes of documents that can contain personal information. The system was implemented using Weka, validated against real documents and integrated into a mail server and keeping track of file's transference to USB drives. The algorithm achieves an accuracy of 83.3% and 87.4% on different sets of documents.

The use this system, integrated into companies' electronic communication infrastructures (mail server, document repositories) will help companies fulfill parts of the GDPR requirements, with respect to the control and traceability of the data transfers.

I. Introduction

The General Data Protection Regulation[1] was published in May 2016 and entered into enforcement on 25th May of 2018, giving each natural person the right to the protection of personal data. This regulation also require controllers (entities that process personal data) to enforce practices that guarantee the natural persons rights.

Part of the rights are related with the knowledge of what data is collected and processed, for what purposes, and with the decision on what to do with such data. Controllers are also required do guaranteed that the data is only used and transferred for the approved purposes and that there is no breaches.

Currently the implementation of the GDPR is still being made by controllers. The principal concern of companies is this stage is in the collection of consent for the processing of personal data, data portability or the right to be forgotten. These three requirements can be easily enforced with developments in the information systems by implementing access control and logging on the accesses to personal information along with posterior audits.

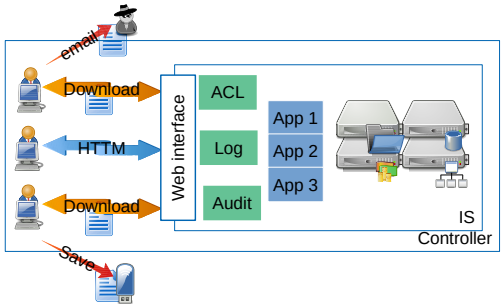

Fig. 1: Typical usage of an organization information systems

Figure 1 represent a typical usage of an organization information systems. The user on the right represent three different accesses and usages of personal data. The middle user accesses information using the web browser, filling forms to do actions and reading results on the screen. The other two users not only download files from the central information systems, keeping the file on the user's computer (for instance pdf documents or spreadsheet listings) and perform two actions that are out of control: mail of the document or copy to a external storage device, thus the system stops tracking such file.

As long as the middle user does not save, write down the information presented on the screen, it is assured that no data breach is occurring. The information systems will guarantee that the access to the personal information is in conformity with the authorizations.

Although organizations may use workflow and document management systems, the mail is still widely used to exchange information (both inside an organization and to external entities). In 2018, 235.6 billions of emails were sent [2], and according Ponemon Institute LLC [3], emails were a big source of data leaks, when users mistype the recipient address.

When a document with personal information is exchanged by e-mail the GDPR compliance can be impacted in various ways:

- Art. 13 Paragraphs 3 – The data subject has to be informed that the personal data will be further processed for a purpose other than that for which the personal data were collected.
- Art. 15 Paragraphs 1.(c) – the data subject should be informed of the list of recipient to whom the personal data will/was disclosed.

- Art. 32 Paragraphs 4 - The controller should guarantee that any personnel that has access to the personal data only processes it with the explicit instructions;
- Art. 33 - The controls should detect data breaches in order to document them and notify required entities.

Currently, the moment a document with personal information sent by e-mail none of the previous requirements can be fulfilled, since the controller stops having any information on such document. This problem occurs the moment a document with personal information is downloaded and saved to and hard disk.

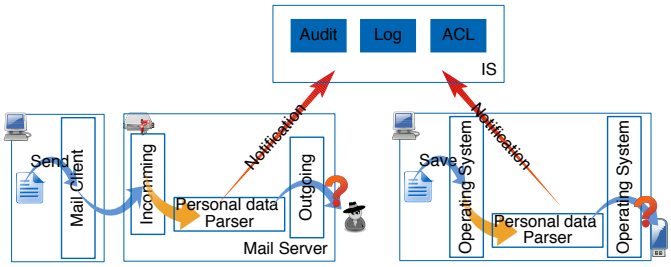

Fig. 2: Data flow control mechanisms' layout

One possible way to give controllers information about the flow of documents with personal data, is to implement mechanisms to detect if files contain personal data are being processed, stored or exchanges from desktop computers, implementing an architecture similar to the enterprise anti-virus, as presented in Figure 2.

The proposed system runs as a daemon on the desktop computer or mail server (represented as Personal Data Parser on Figure 2) and detects if a certain file contains personal information. Depending on the implemented enterprise policies the file operations can be logged, restricted and audited (modules Audit, Log, ACL).

Two important control points in a organization, where files can be verified, are the saving of files to external storage units (e.g. USB pen drives), and the transfer of files by mail. Both of those operations could pose a threat to the GDPR compliance if the data flow isn't controlled, verified or logged.

The Personal Files Scanner verifies the content of files and determines if any of the present information can be considered personal according to GDPR: any information that can lead to an individual is considered personal.

After the evaluation of various alternatives for the classification of files, the chosen machine learning algorithm was Decision Trees. in order to train the system a new dataset was constructed and a set of heuristics were defined for the creation of the feature vectors.

Our system was evaluated against the dataset and against real files obtained from two personal computers. With the defined training dataset and developed heuristics, the accuracy of the algorithm is about 83% using the files from one computer and about 87% using the other files. The Personal Files Scanner was also integrated into a service that verifies all file saves into external drives and into a mail server. The quantitative results and demonstrated uses for Personal Files Scanner , gives us the confidence that this tool can contribute to the enforcement of GDPR in most organizations.

The next chapter presents a literature review related to research and work here presented.

The third chapter describes what are the tool's requirements, its architecture as well as its implementation. It is detailed what is the training set used for the Machine Learning model training, as well as how that model was chosen and how it was implemented on the tool.

The fourth chapter describes two testing sets and how there were used to evaluate the tool, it is also described the procedures used for this evaluation. The final section on this chapter explains the results obtained.

The fifth chapter shows a overview of the two software applications that we implemented using the Personal Files Scanner

In the sixth chapter, there is a discussion about the tool's components, discussing some of their problems and what could have been different. it is also discussed the Personal Files Scanner and its applications.

The last chapter brings closure to the work highlighting the work done, what was achieved and discusses what work could be done next with this work as a base.

II. Related Work

Considering the requirements GDPR imposes and the mentioned problems, we need to research studies relating to GDPR, Machine Learning and any techniques used to solve similar problems.

A. General Data Protection Regulation

The General Data Protection Regulation [1] (GDPR) is the new regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, it repealed Directive 95/46/EC[4] (General Data Protection Regulation).

Two fundamental definitions for GDPR are personal data, i.e. *information relating to an identified or identifiable natural person (data subject);* and identifiable natural person, which is someone *that can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person*[1].

In line with these two definitions, and according to chapter 2, article 5 of the GDPR controllers are accountable for all the life cycle of the collected personal data. This cycle includes the local storage and processing of data, but also its transfer to processors or other recipients.

GDPR forces companies to take some general procedures [5] regarding the data they have from their clients:

- Companies that have some kind of personal information about anyone from the European Union need to be in compliance with GDPR.

- Companies need to investigate their system and find if there are some weak spots where data could be leaked.
- Companies need the consent of the one who shared his/hers personal data.
- Companies have to explicitly specify how they will use the information and which information is kept.
- Companies have to control and possibly deny the data flow, to ensure that no data has been leaked.
- Public companies needs to hire a data protection officer, responsible for reviewing the GDPR, and checking its compliance.

Individuals/data subjects have now some rights that must be fulfilled by companies:

- Individuals can revoke his consent about a company having his data at any time. After a revocation of consent companies are required to delete requested data.
- Individuals can request a list of what information is being processed and referring him/her. Companies must comply with this request and provide give a copy of all the information being stored that refers such individual.

The previous procedures and requirements on the handling personal data, GDPR are explicitly expresses in the regulation:

- The data subject has to be informed that his personal data will processed for a purpose other than that for which the personal data was initially collected (Art. 13 Paragraphs 3);
- The data subject should be informed of the list of recipient to whom the personal data will/was disclosed (Art. 15 Paragraphs 1.(c)).
- The controller should guarantee that any personnel that has access to the personal data only processes it with the explicit instructions (Art. 32 Paragraphs 4)
- The controllers should detect data breaches in order to document them and notify required entities (Art. 33).

These requirement are in line with one of the principles related to the processing of personal data (Article 5) that define that personal data shall be processed in a manner that protects it from unauthorized use and ensures other security characteristics.

According to the chapter VIII of the GDPR, companies that don't comply with it or fail to protect personal data can incur compensations, fines or other penalties.

1) Current implementation: The regulation entered into full enforcement in May 2018, but its implementation by companies has being slower.

Another survey[6] also finds that at the same date the implementation was low: only 19% of the respondents confirmed that they were GDPR ready. The survey also reveals that most of the companies are spending more than 250.000 euros in the implementation of necessary procedures, with 16% spending up to 5 million euros.

Identity and access management (IAM) helps companies to be GDPR compliant by ensuring that information is on a need to know basis, where only the person that needs some information has access to it. IAM helps to know where information is stored when accessed by someone and what it is being used for. The ITProPortal website [7] gives some insights on the importance of the IAM, while CSOOnline [8] gives some insights on how IAM protects personal data, suggesting that it ensures for example authorization, auditability, among others. NNIT [9] explains that companies have been reluctant to implement a proper IAM since, until the GDPR's implementation date, there was less risk in not doing so.

Until January 2019 (8 months after the GDPR's enforcing date), there has seen an increase on the number of complaints, as presented by the European Data Protection Board [10] data. The number of reported breaches has also increased. The reason for this increase may not the related to the number of breaches, or the use of new tools to find them, but is due to the awareness about the new requirements [11].

2) IT tools support: Besides organizational and policy changes, companies are also implementing and changing information systems in order to comply with the GDPR. Part of these changes are done internally to current information systems, but there are some generic tools that can be used and adapted to different companies.

One of such tools is offered by IT Governance[12]. It provide provides a set of Cloud-based tools [13] to help with the GDPR compliance.

DRM techniques can also be applied to the protection and control of information access. Locklizard [14] protects by including DRM mechanisms into pdf files. When a application generates a pdf file, the Locklizard library encrypts, guaranteeing that it can only be accessed using the secure pdf reader. This reader can enforce access controls to the files, easing the compliance to some GDPR requirements.

B. Machine Learning

Machine Learning is a field of computer science used in various applications, Computer Vision, Medicine, Data Mining, and more related to the work here presented, text classification. It allows computer system to make predictions or classifications of data in a automatic fashion and with minimal user interaction.

Machine Learning algorithms can be split into two Major groups: Supervised algorithms [15] that uses a training set to train some model to learn to predict outcomes, and unsupervised algorithms that don't need any training set. Hence, the name unsupervised, since no one supervises the learning procedure.

With respect to the relation of machine learning and the GDPR most of the work is related to implications of the regulation on this type of data processing. Part of the concerns related to the use of machine learning Algorithms on personal data comes from the article 22 of the GDPR states that no decision which produces legal effects should be based solely on automated processing including profiling[16]. In line with this, some studies about this article presented the impact on Machine Learning Algorithms on the automatic processing of personal data[17].

Since the GDPR requires that every decision on the data processing must have a good and demonstrative explanation it is necessary to translate to human readable information what the machine learning systems do. Rulex [18] is a tool that produces understandable rules from several Machine Learning algorithms that the user can use to filter the results. These ruling sets are created to ensure that any company can easily explain the automated decisions. Other authors [19] claim that any machine learning based system should require the intervention and explanation by humans.

This work is more related to the work done in automatic tools that use machine learning algorithms to do auditing and spam/virus detection, as presented in the following subsections.

C. Email Audit and Spam Detection

A lot of information is being interchanged through emails. Which also leads to an increase of emails, reason why email auditing is becoming more important. Email auditing needs to classify emails into different categorizations with minimum human interaction, which is similar to our case, where we need to classify files into containing personal information or not.

Glen L. Gray et al [20] describes two main type of email analysis: content analysis and log analysis. In the content analysis, the system searches emails for certain key words. This approach has a problem, searching for many keywords would lead to a lot of false positives, therefore they suggest that more parameters could be added to the query, which coincides with the our approach. It is also detailed some difficulties on Email Data Mining, like misspelled words, which affects the Key Word Search approach.

Adrian Gepp et al [21] studied a lot of articles on auditing and chose the most important ones. They found multiple studies on different techniques that can be applied to different fields. Some techniques for financial distress and others for financial fraud. They concluded that there isn't a algorithm that is widely knows as better one. Most of these approaches can be used for our problem.

This increase in the usage of emails as a communication tool leads inevitably to an increase in unwanted emails, with for example advertising. These emails are called spam. People usually prefer to avoid these emails altogether. Thus, there is the need to create a way of detecting the type of email.

According to the study [22] by R. Kishore Kumar et al, no algorithm achieves 100% accuracy on spam detection, which leads to the existence of multiple studies researching the best algorithm to perform this classification. R. Kishore Kumar et al's study uses a data mining tool to explore various classifiers. It studies many algorithms, concluding that C4.5 is one of the algorithms that achieves better results in the classification.

Neetu Sharma et al [23] gives an explanation on what is spam as well as describes some algorithms that can be used for spam detection. The algorithms require a training set to learn how to classify mails. Algorithms like Support Machine Vectors (SVM), K-Nearest Neighbour Classifier (kNN) and the Naive Bayes Classifier.

Omar Saad et al [24] describes in more detail what is spam, also giving insight on the structure of an email. The study states that there are two levels where the spam filtering can operate, individual and enterprise levels. The same can be said of our problem, classifying files as personal or not can also be implemented on both levels. Omar Saad et al also approaches some algorithms that can be applied to spam filtering. Like the Naive Bayes Classifier, SVMs, K-Nearest Neighbour Classifier, Artificial Neural Networks and Artificial Immune System classifier. In our process of choosing the best algorithm, we also tested an implementation of the Artificial Neural Network.

D. Malware Detection

With all data being stored in digital form in the past years, there have been a lot of studies and books concerning cybersecurity, since a breach in security leads to the loss/exposure of data, which can mean a violation to the GDPR. According to the study [25] in 2008 there were well over 100000 known computer viruses. Antiviruses are a good example of a data mining method for cybersecurity, since one of the ways they operate is by scanning the potentially dangerous file and checks if they contain any virus code, comparing it with known viruses. That is one of the reasons why their database is always being updated to have a wider range of known viruses (bigger and more rich training set).

Shubair Abdulla1 at el [26] researched methods to detect unknown worms and analysed how those Malware Detection algorithms could be implemented on networks. For the classification, they take into account two algorithms, comparing their performance. Naive Bayes (NB) and K-Nearest Neighbour (KNN). The chosen set of features is based on worms' observation. KNN classifier achieved better results than the NB algorithm. When using 2000+ instances KNN achieved an average of 2.8% more than NB.

Jhonattan J. Barriga A. et al [27] gives a brief overview of what is an anti-virus and how it detects malware. It specifies three main categories of methods: Signature Based Method, which compares sequences of bytes with a database, Behaviour Based, which detects malicious code by analysing its behaviour, Heuristic Based, which analyses keystrokes and system behaviours to detect something abnormal, comparing those with previous vulnerabilities, and there is yet another category: Artificial Intelligence-based, which are methods implementing algorithms that can learn by themselves. They study the two main categories of Machine Learning algorithms: Supervised Learning and Unsupervised Learning algorithms. In this case, Neural Networks, both supervised and unsupervised.

III. Personal Files Scanner

A. Requirements

This system analyses text and spreadsheet documents to verify if they contain or not personal data, according to GDPR. It needs to do so with minimal human interaction.

It needs to be:

- Accurate, to ensure a correct files classification.

- Efficient, to ensure a quick classification, without too much wait time.
- Interoperable, so that it can be deployed in conjunction with other systems.
- Extensible, so that it can be improved or adapted to any environment with minimum effort.

B. Architecture

In the figure 3, can be seen the architecture of the tool. The testing phase depends on the completion of the training phase. There are components in charge of the data processing and components relative to Machine Learning that classifies the file.

The File Parser contains two modules: the File Reader and the Feature Extractor. The File Reader receives a file or a folder to analyse (procedure 1 and 4 in the figure 3). This module transforms the files into text lines (procedure 2 in the same figure), to be latter processed by the Feature Extractor component. The Feature Extractor receives those lines and according to a set of rules defined by the heuristics returns a set of feature vectors for each file (procedure 3 and 5 in the figure 3). The supported files are text files (txt, docx, doc and pdf) and Microsoft Excel files (xlsx and xls).

In the training phase (procedure 4 and 5 in the figure 3), the results of the File Parser Module are combined with a manual files' classification to create a set of feature vectors that is used to create the Machine Learning model. This process produces a model that has learned how to classify files.

In the testing phase (procedure 1, 2 and 3 in the figure 3), a feature vector for each file, obtained in the File Parser, is provided to the Machine Learning model, that will use them to classify such files.

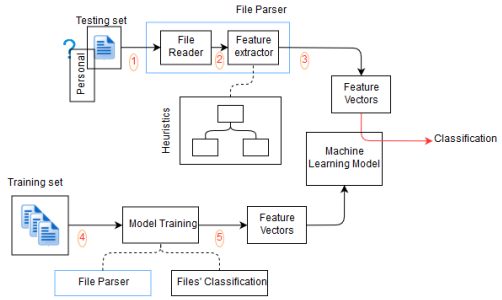

Fig. 3: Personal Files Scanner Architecture

C. Implementation

1) File Reader: This module receives a path where the files are located. Every path, contained on it, is recursively visited and the detected files are scanned if its format is supported.

The supported file formats are the following:

- Text files (txt).
- Microsoft Word files (doc/docx).
- Microsoft Excel files (xls/xlsx).
- Portable Document Format (pdf).

For the .txt files, it is used java.io to open and read the file, in Microsoft Office Word files' case(docx,doc), and Microsoft Office Excel (xlsx/xls) files it is used a library "org.apache.poi" from an API called Apache POI created by the Apache Software Foundation, for Portable Document Format (.pdf) files it is used the library Apache PDFBox.

The files are separated by lines and given to the next component.

2) Feature Extraction: When heuristics are used on a system, they are used to solve problems efficiently but with no guarantees of finding the best solution [28]. A heuristic based algorithm is one that searches for sets of keywords or keyphrases to differentiate files.

In order to program these heuristics a abstract class is provided. That class implements the interface ClassInterface, which has 3 methods:

- find method - Which receives 6 strings.
- setStringFound method - Which receives 1 boolean.
- getStringFound method.

The method find checks if the heuristic is there, considering the constraints each heuristic has. The method setStringFound changes the value of StringFound, which is a boolean that acts as a flag representing if that heuristic was found or not. The method getStringFound returns the value of StringFound.

D. Feature Vector

A feature vector represents the presence of each heuristic, as well as the class to classify. each feature vector is represented by several yes and several no, one for each heuristic. Which means that the heuristic is on the file or not. Ending in a question mark which represents that the file has not been yet classified.

E. Training set

We want the tool to be easily suited to any environment, therefore it is better to have a training set that can be adapted to the needs of the system implementing it. Thus, leading us to Supervised Machine Learning algorithms.

The files from the training set were chosen from a wide variety of files on numerous computers (a student's, a professor's and a doctor's). This dataset has two types of files, text like txt, doc/docx and pdf, and spreadsheet (xls/xlsx) files. It has synthetic and real files. The purpose of the synthetic files is just to add some kind of pattern for the Machine Learning algorithm to learn better.

The dataset characteristics can be summarized in table I, with respect to the class of information stored.

	Personal	Non-personal	Total
Real	15	18	33
Synthetic	33	4	37
Total	48	22	70

TABLE I: Number of files in the training set

F. Machine Learning model choice and training set validation

With the file flagged as potentially personal (containing at least one heuristic) there is a need to choose which machine learning algorithm to use to do the classification. The best

type of algorithm for this type of classification are supervised algorithms because they use a training set that the user can modify at will. The algorithms considered are C4.5 (Decision Trees), Support Vector Machine, Neural Networks and Naive Bayes. It was used the open-source software created by The University of Waikato that allows the testing of all algorithms. It was tested different configurations for the algorithms in the software Waikato Environment for Knowledge Analysis [29].

1) Dataset validation: To check if the dataset is large and rich enough to be useful, it was used as training set and validation with several Machine Learning algorithms, listed in the section III-F. This evaluation was performed using the WEKA [29] software, which by giving it a training set and a testing set it is possible to test the algorithms' performance. For each of the selected algorithms a subset of the available documents was used to train, and another subset was used for validation. In order to create each document's feature vector, a set of heuristics, that correspond to the information on each document, was used.

Once the training set reaches 80% of the size of the original training set is reached almost all algorithms have 100% accuracy (The accuracy of Neural Networks is 90.9%) which indicates that the dataset used as training set has enough patterns to classify correctly the rest of the files. Therefore, the entire training set is valid for the heuristics searched for, as well as the Portuguese language.

2) Results: The training set was tested using two approaches, testing directly on the same files directly and using cross validation (further explained in the section IV-B1).

The algorithm that learns worst using the training set as testing set is the Naive Bayes. However, the Support Vector Machine and Neural Networks algorithms have less accuracy than the Naive Bayes one when performing 13-fold cross-validation, which leads us to believe that those algorithms need bigger training sets. When performing 10-fold cross-validation, for this training set, the algorithms with better accuracy to use are the Decision Trees or Neural Networks, but since overall Decision Trees had the best results that was the chosen algorithm.

G. Model Implementation

To classify the files it is needed the Machine Learning model. The WEKA package contains the method to create the decision tree. This model was created in a average time of 0.3 seconds.

IV. Personal Files Scanner Evaluation

A. Testing Setup

It was created two testing sets to test the Personal Files Scanner. One testing set is a set of files taken from a professor's downloads folder and the other is a students' computer (more similar to the training set), both filtered to ensure that they only had Portuguese files (since the tool was trained for this language) and no files containing only images.

For these tests, it was used a computer with a 8 cores AMD ryzen 7 2700x and 16 Gb of RAM.

Since the Machine Learning model was already decided using the heuristics and training set, these are the same as in subsection III-C2 and section III-E, respectively. Thus, the model is the same one as the one chosen in section III-F.

B. Performance Evaluation

1) Evaluation Procedures: Since there are many Machine Learning algorithms with different characteristics, there is the need to use some method to compare them. A method like k-Fold Cross-Validation, that estimates how well each algorithm classifies unseen data. The k-Fold Cross-Validation procedure divides the training set in k blocks. Then one block will be the testing set while the rest is the training set storing the evaluation score, next the second block is the testing set while the other blocks are the training set and so on until there is no block that hasn't been the testing set. At the end, it is averaged the scores of the k tests.

	Predicted as personal	Predicted as non-personal
Actual personal	True Positive (TP)	False Negative (FN)
Actual non-personal	False Positive (FP)	True Negative (TN)

TABLE II: Confusion Matrix

Aditya Mishra [30] details some metric to evaluate how well a algorithm performed, like the confusion matrix (table II), that works great to visualize how the algorithm tackled a binary problem like ours. Which can be used to compute another metrics like the following:

- Classification Accuracy (ACC) - Which tells how close the measured values and the true values are.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- True Positive Rate (TPR) - Which is the percentage of the correctly classified positives.

$$TPR = \frac{TP}{TP + FN} \tag{2}$$

- Positive Predictive Value (PPV) - That it is the percentage of positives that are true positives.

$$PPV = \frac{TP}{TP + FP} \tag{3}$$

Using both testing sets described in the section IV-A, it is possible to test the effects that different training sets and different number of heuristics have on the tool's performance. The following tests were used:

- Test 1 - Small training set (48 files) and few heuristics (29 heuristics, two of which were never detected).
- Test 2 - The same training set and more heuristics (the same 34 that were detailed in the subsection III-C2) than in test 1.
- Test 3 - Bigger training set (the one with 70 files detailed in III-E) and same heuristics than in test 2.

2) Results: Using the batch of tests detailed in the previous section, the tables III and IV contain the classification results of the professor's downloads folder.

	Predicted as personal	Predicted as non-personal
Actual personal	93	32
Actual non-personal	18	165

TABLE III: Professor's download folder analysis confusion matrix for test 3

	Accuracy	TPR	PPV
Test 1	64.5%	47.7%	53.5%
Test 2	70.3%	50.5%	64.4%
Test 3	83.8%	74.4%	83.8%

TABLE IV: Professor's download folder classification results

As can be seen, from the table IV, increasing the training set's size and the number of heuristics, leads to the best results. Thus, it can be assumed that besides choosing a good algorithm, there is the need of having a good training set and smart choosing of heuristics.

The most dangerous wrong classification are false negatives (files classified as non-personal, when they are in fact personal), the false negatives obtained were mostly files that had no way of being discovered since they didn't have any important heuristic, for example one false negative was a report from a project, that the only thing it had was the name of the student, therefore the only way for the report to be detected is to add a heuristic for each possible name.

The algorithm performed better with the student's computer, using the same training set and heuristics as in test 3. It had 87.4% accuracy, which was an increase of 3.6% when compared with the professor's folder, which helps us conclude that analysing files more similar to the training set leads to better results. Thus, the training set and the heuristics should always be adapted to the tool's environment.

Since the paper [15] compares some algorithms on a binary classification, which is the same type of classification as this work's case. Thus, it is possible to compare this work's results with the ones from the paper. The best results present in the study, using decision trees, has the accuracy of 84.6%, which isn't much higher than 83.8% and is less than 87.4%. Therefore, it can be considered, that the Personal Files Scanner's results were in line with other works.

C. Response Time

1) Results: Using the files from the student's computer, it was possible to test how much time the Personal Files Scanner takes to analyse files with different sizes.

Almost every file smaller than 4000 Kb takes less than 0.5 seconds to be classified. This happens because these files neither have a high number of heuristics nor many pages. The files, smaller than 3500 kb, taking more than 1.5 seconds, although not large in size, they contain numerous pages, which incurs in taking longer to read them line by line. As expected, when we go to higher file's size, thus increasing the number of pages, which also leads to an increase in the number of heuristics found, the file processing takes longer.

V. Applications

A. Media Drive Scanner

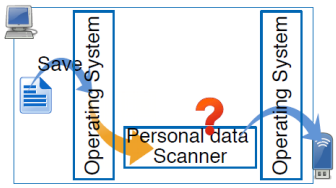

Fig. 4: Layout of a system that uses this tool to detect personal files being copied

After a file is copied to an USB drive, its controller loses the control over that data, from the location to what processing is being done on it. That's the reason that keeping logs of who made the copy, when and if it was an authorized copy is so important.

This application (which is the Personal Data Scanner present in the figure 4) tries to solve this problem by keeping track over all the drives like a detection system using the java.nio.file library, similar to anti-virus that waits for a new file to be created. Whenever a new file is created, the application uses the same library to detect this event, then such file is moved to a directory outside the USB drive, so that the Personal Files Scanner can be run on it. If it doesn't contain personal information the file is returned to the USB drive. If the file does indeed contain personal information, then a action should be taken. This action depends on what are the requirements of this program. The file can be returned to the USB drive just keeping the access log or the copy action can be denied, thus not returning the file to the USB drive, effectively ensuring that no control over the data is lost.

B. Electronic Email Scanner

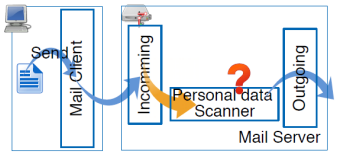

Fig. 5: Flow of a system that uses this tool to check emails being sent/received

When sending an email to another person there is no guarantee that the other person won't mishandle the data or that this email won't be sent by mistake to the wrong person. Which leads to the fact that emails have been a major cause of leaks. This application intends to help solve this problem by analysing the emails before they are sent.

The mail server seen in the figure 5 was Postfix. Postfix supports a plugin called FuGlu [31], which is a daemon for email scanning. This plugin intercepts suspect emails that pass

through postfix, scans them and it may or may not return them to postfix allowing its designated path. FuGlu supports many filters and gives the users the possibility to program, in Python, additional plugins. Therefore, the Electronic Email Scanner was be implemented as a plugin for FuGlu.

The Electronic Email Scanner receives the suspicious email, delivers the attachments to the Personal Files Scanner tool, thus discovering if any attachments is a personal file. Upon this discovery, it is kept a log of who sent the email and to whom. The plugin can also be used to deny sending those emails to prevent data leaks, or to request authorization.

C. Overhead Evaluation

1) Media Drive Scanner:

a) Setup: The model, heuristics, training set and computer are the same as the ones used in the section IV-A. The testing set are the files from the students' computer, from the section IV-A.

For these tests, it was inserted a USB drive into the computer, and all said files were copied to the USB drive, with the application running and measuring the time each copy took.

b) Results: As expected, when the Media Drive Scanner is active it takes more time to copy a file. That happens because with the scanner running, there is the copy time and the time to move this file at least once (in this case twice) plus the processing performed by the Personal Files Scanner.

The response time still has the same behaviour, which happens because the larger processing is done by the Personal Files Scanner. The copying and moving actions just apply an extra delay.

The difference in processing time between the Media Drive Scanner and the Personal Files scanner ranges from 0,41 seconds (with a file size of 37 Kb) to 2,00 seconds (with a file size of 7442,75 Kb).

2) Electronic Email Scanner:

a) Setup: The model, training set, heuristics and computer are the same as the ones used for the tests in the subsection V-C1, whereas the testing set is composed by 15 files (9 non-personal and 6 personal). For these tests, it was configured a local only email server, that on the way out would flag each email sent has suspect, to ensure that FuGlu would capture it.

To be able to analyse the application's impact on the system, it was visualized the CPU and RAM load with three tests, where it was sent an email per second during 60 seconds, right after 20 seconds of idle. The tests were the following:

- The first test was to send emails with an attachment file of 29.8 Kb (which was a Curriculum Vitae of a student).
- Test number two was to send emails with an attachment file of 2.6 Mb (which was a publicly available contract with 12 pages).
- The third was to send the entire testing set, four times.

b) Results: From the figure 6 to figure 11 contains the load that the email system takes on a system, with (blue line) and without the scanner (orange line) running, while sending one email per second throughout 1 minute, which is represented by the black line(which is a higher send rate than the average email send rate on the Técnico mail server). As expected the Electronic Email Scanner incurs in additional system load, since there is a significant amount of extra processing being done.

As expected, the test 2 takes longer that the test 1. Due to the fact, that the small file (29.8 Kb) takes less than a second to be analysed, whereas the big file (2.6 Mb) takes 3.22 seconds, thus delay the subsequent emails. Therefore, the impact that the system takes on the computer also is bigger on test 2, since it has to process more files at the same time. The third test time is in the middle and both CPU and RAM load keep increasing and decreasing, depending on the file's size.

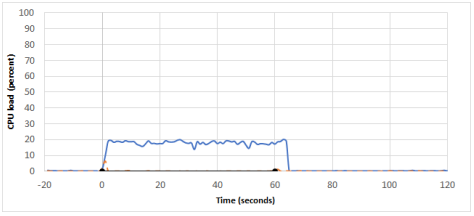

Fig. 6: CPU load for test 1

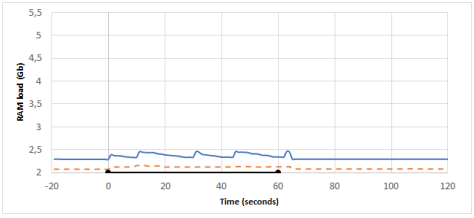

Fig. 7: RAM load for test 1

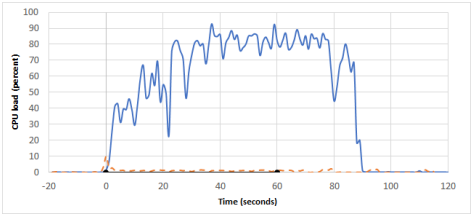

Fig. 8: CPU load for test 2

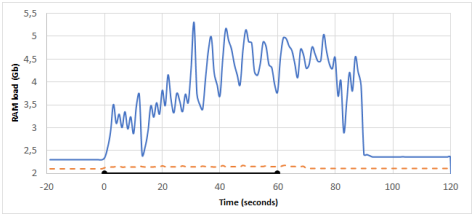

Fig. 9: RAM load for test 2

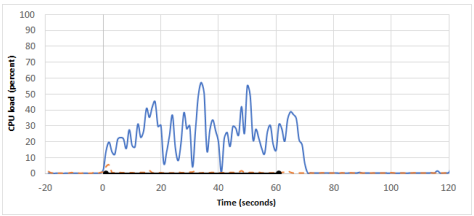

Fig. 10: CPU load for test 3

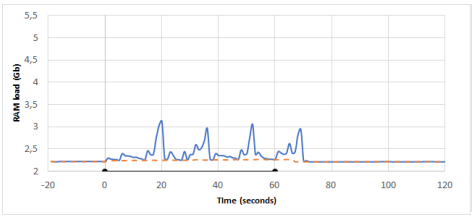

Fig. 11: RAM load for test 3

As can be seen from the figure 6 to figure 11, the scanner doesn't add much delay, being the worst case scenario a difference of 16 seconds when sending the larger file, this extra time comes from the files processing which delays the emails.

VI. Discussion

A. Dataset

The training set contains some files that have been created synthetically. However, creating these file only with this pattern means that it has less noise than a real file.

For all the files that were used as a testing set the training set leads to a good accuracy value. With a bigger training set, with different patterns it could lead to even better results.

The dataset can have some missing patterns since there is a wide variety of files, depending on each user. So this dataset should always be completed with more and more files.

B. Heuristics

The heuristic based algorithms have the problem that a literal keyword is the one being detected, similar keywords aren't detected and for them to be detected a new heuristic would have to be added. An example of this being a problem is that if some keyword is badly written it isn't detected, which could lead to a wrong classification.

Some heuristics also appear too many times, appearing sometimes in places they don't mean anything, and also in different languages. To solve the heuristics' appearance problem one could add additional constraints.

The heuristics classes used in this work, were created considering the Portuguese language, thus requiring additional classes to be implemented when switching to another language.

C. Machine Learning

The Decision Trees have some problems, since they can overfit and they have problems on the classification of some pattern that doesn't appear on the tree. Problems that can be solved increasing the training set with different patterns.

The algorithm that reads the file in search of the heuristics only reads text, which means that a file that contains images and no text would be analysed as being empty.

Since the Decision Tree model created from the training set is quickly created (less than a second), one procedure that could be implemented is to have so form of feedback from the results, where those results could be, automatically, added to the training set, ensuring a bigger and richer training set with less effort. However, that idea has a major flaw, since there is no guarantee that every file is 100% correctly classified. Which would mean that the automatic feedback would have to be supervised file by file to ensure that every classification was indeed correct.

D. Personal Files Scanner

As demonstrated by the creation of the applications in the chapter V the Personal Files Scanner has multiple applications. The problem introduced in the section I has been solved by the introduction of these applications, giving organizations the option to deny the email sending and the file copying or at least log them. This way, the data flow control is never lost, thus ensuring the correct manipulation of personal data.

E. Applications

Both the Electronic Mail Scanner and the Media Drive Scanner are dependent on the creation of a log file by the Personal Files Scanner, because they both use the existence of this file to determine if the file analysed contains personal information. They both incur in extra time because of the extra processing, but as the Electronic Mail Scanner is deployed in a mail server, one doesn't even know when the email has been sent, thus not noticing this application's presence.

The Electronic Mail Scanner was created as a plugin for a Postfix plugin (fuGlu), thus being dependent on FuGlu updates.

VII. Conclusions

This work shows what is the General Data Protection Regulation and what is the difference it makes. Specially what the Regulation means to companies that needs personal information from people in a regular basis. It showed that some companies still are not fully prepared for the new regulation, which shows how important the tool created can be, since it can facilitate in detecting personal files, in different occasions.

It was possible to understand from the tests performed on the tool, that different algorithms and different occasions have different training set needs. Some may require a larger training set with more noise. Therefore, it is important that the training set is the correct one, to ensure that the tool is accurate on the files' classification.

Overall, the impact that the tool takes on a system needs to be compared with its importance. With the batch of tests and testing the tool in action, in real applications, it can be concluded that the impact isn't big, when compared with the benefits it can bring. It was possible to see that a efficient tool like this one can also be accurate when classifying files.

This tool can be adapted to any environment and can be used to create a multitude of applications that deal with personal data. Thus, being considered an extensible tool.

When considering languages besides the Portuguese one, the tool needs new heuristics, which can be implemented by providing a new classe, for each heuristic, to the tool. That's the reason it is so important that the tool is flexible and easily adaptable.

A. Future Work

It would be interesting to add to this tool the ability to read images. That way both text and images could be classified, making it possible to discover personal files in a wider range of types of documents. This tool can also be used as a basis to another automate tool, to check GDPR-compliance. This work's tool discovers what personal information are contained in a file, then it could exist a tool that would compare those results with what authorizations a company's client has given.

It would also be interesting to test the Personal Files Scanner in computers belonging to organizations, since the training set and heuristics used on this work would detect most of the personal data kept by Portuguese organizations. This testing would also inform the organization where the personal files are kept, since some of them could be in hidden folders.

The tool created in this thesis could be integrated in Auditing and Access Control tools.

ACKNOWLEDGMENT

The authors would like to thank Prof. João Nuno de Oliveira e Silva, for all the support, guidance and for always being available.

REFERENCES

[1] "Regulation (eu) 2016/679 of the european parliament and of the council." https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679. Accessed: March 3rd 2018.

[2] R. Team, "Email statistics report 2015-2019," *The Radicati Group*, 2015.

[3] P. I. LLC, "The escalating importance of email encryption, confirmar author," 2011.

[4] "Directive 95/46/ec of the european parliament." https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A31995L0046. Accessed: January 19th 2019.

[5] C. Tikkinen-Piri, A. Rohunen, and J. Markkula, "Eu general data protection regulation: Changes and implications for personal data collecting companies," *Computer Law & Security Review*, 2018.

[6] P. Fisher, T. Grosser, and L. Iffert, "Managing personal data beyond the gdpr," tech. rep., BARC - Business Application Research Center, 2018.

[7] M. Kapetanakis, "Gdpr - changing the rules of identity and access management." https://www.itproportal.com/features/gdpr-changing-the-rules-of-identity-and-access-management/. Accessed: April 13rd 2018.

[8] A. Biger-Levin, "The role of identity in gdpr compliance." https://www.csoonline.com/article/3269589/the-role-of-identity-in-gdpr-compliance.html. Accessed: April 13rd 2018.

[9] S. Peacock, "Identity and access management becomes a top priority due to the eu gdpr." https://www.nnit.com/OfferingsAndArticles/Pages/Identity-and-Access-Management-and-EUGDPR.aspx. Accessed: April 13rd 2018.

[10] E. Commission, "Gdpr in numbers." https://ec.europa.eu/commission/sites/beta-political/files/190125_gdpr_infographics_v4.pdf, 2019. Accessed: March 15th 2019.

[11] GDPRToday, "Gdpr in numbers." https://www.gdprtoday.org/gdpr-in-numbers-3/, 2019. Accessed: March 15th 2019.

[12] IT Governance Ltd, "It governance website." https://www.itgovernance.co.uk/. Accessed: March 25th 2018.

[13] IT Governance Ltd, "Gdpr compliance software." https://www.itgovernance.co.uk/gdpr-compliance-software. Accessed: March 25th 2018.

[14] Locklizard, "Gdpr & document protection." https://www.locklizard.com/document-security-blog/gdpr-sending-documents-securely/. Accessed: November 27th 2018.

[15] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*, pp. 161–168, ACM, 2006.

[16] Privacy International, "Data is power: Profiling and automated decision-making in gdpr." https://privacyinternational.org/report/1718/data-power-profiling-and-automated-decision-making-gdpr. Accessed: June 24th 2018.

[17] N. Wallace and D. Castro, "The impact of the eu's new data protection regulation on ai," tech. rep., Centre for Data Innovation: Washington, DC, USA, 2018.

[18] Rulex, Inc., "New compliance risks for machine learning – the eu general data protection regulation (gdpr)." http://www.rulex.ai/wp-content/uploads/2017/04/GDPR-WhitePaper-20170331-EP.pdf, 2017. Accessed: June 24th 2018.

[19] D. Kamarinou, C. Millard, and J. Singh, "Machine learning with personal data," *Queen Mary School of Law Legal Studies*, no. 247/2016, 2016. available at SSRN: https://ssrn.com/abstract=2865811.

[20] G. L. Gray and R. Debreceny, "Data mining of emails to support periodic and continuous assurance," *College of Business and Economics, California State University at Northridge, Working Paper*, 2007.

[21] A. Gepp, M. K. Linnenluecke, T. J. O'Neill, and T. Smith, "Big data techniques in auditing research and practice: Current trends and future opportunities," *Journal of Accounting Literature*, vol. 40, pp. 102–115, 2018.

[22] R. K. Kumar, G. Poonkuzhali, and P. Sudhakar, "Comparative study on email spam classifier using data mining techniques," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, pp. 14–16, 2012.

[23] N. Sharma and A. Verma, "Survey on text classification (spam) using machine learning," *(IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp. 5098–5102, 2014.

[24] O. Saad, A. Darwish, and R. Faraj, "A survey of machine learning techniques for spam filtering," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 12, no. 2, p. 66, 2012.

[25] E. Al Daoud, I. H. Jebril, and B. Zaqaibeh, "Computer virus strategies and detection methods," *Int. J. Open Problems Compt. Math*, vol. 1, no. 2, pp. 12–20, 2008.

[26] S. Abdulla, S. Ramadass, A. A. Altyeb, and A. Al-Nassiri, "Employing machine learning algorithms to detect unknown scanning and email worms.," *Int. Arab J. Inf. Technol.*, vol. 11, no. 2, pp. 140–148, 2014.

[27] J. J. B. A. and S. G. Yoo, "Malware detection and evasion with machine learning techniques: A survey," *International Journal of Applied Engineering Research*, vol. 12, no. 18, pp. 7207–7214, 2017.

[28] M. H. Romanycia and F. J. Pelletier, "What is a heuristic?," *Computational Intelligence*, vol. 1, no. 1, pp. 47–58, 1985.

[29] "Weka documentation." https://www.cs.waikato.ac.nz/ ml/index.html. Accessed: June 16th 2018.

[30] A. Mishra, "Metrics to evaluate your machine learning algorithm." https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234, 2018. Accessed: May 24th 2019.

[31] "Fuglu - fuglu documentation." https://fuglu.org/. Accessed: May 22nd 2018.