IN SUPREMÆ DIGNITATIS

1343

UNIVERSITÀ DI PISA

Department of Information Engineering (DII)
M.Sc in Computer Engineering

Business and Project Management Project

Giuseppe Aniello, Edoardo Malaspina, Pietrangelo Manco

Academic Year 2021/2022

Contents

1 Project Goal and Research Questions

In this section we exposed the Goal of the Project, listing the topics of interest of our analysis and describing in details the major questions asked for each of them.

1.1 NER Question

The research was focused on the following question:

Who are the users of AI applications for Fraud Detection?

We analyzed a lot of patents, identifying the ones that are actually related to fraud detection using AI to discover who are the users of this technology. We exploited NER in order to find the companies related to this kind of applications.

1.2 Trend Question

The research was focused on the following question:

Can you identify a trend in AI application for Fraud Detection?

Analyzing the patents, we identified the ones that are actually related to fraud detection using AI to discover how this kind of technology evolved in the past years.

1.3 Main Topics Question

The research was focused on the following question:

Which are the main topics in documents related to AI applications for Fraud Detection?

In order to perform this task, we integrated the previous list of patents with 30 more topic-related documents found on Google Patents. The goal was to identify the most frequent topics discussed alongside the technology of interest.

1.4 State of the Art Question

The research was focused on the following question:

What are the State of the Art AI techniques for Fraud Detection?

We analyzed a lot of papers in which different research groups did surveys on the State of the Art methods for Fraud Detection, and using NLP techniques we identified the most common among those methods.

2 Process Description

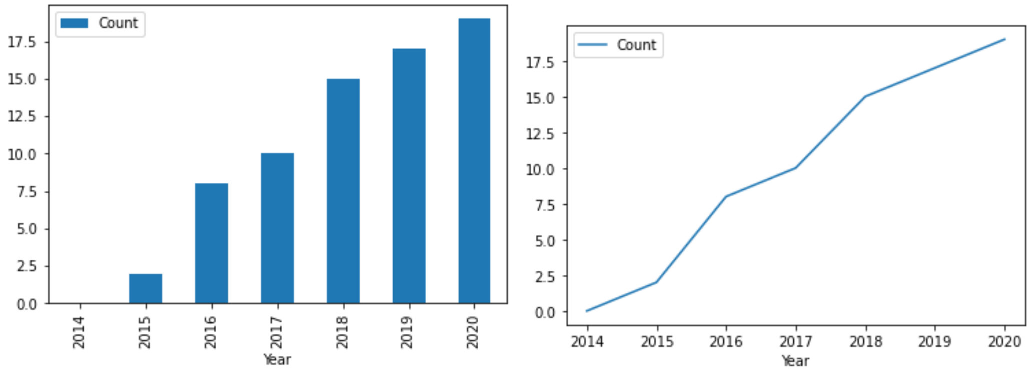
In this section we described the logic processes used in order to answer each question of interest for our analysis. All the technical details and the various steps will be shown in the "Technical Report" in a following section of this document.

2.1 NER Question

We started observing a dataset of 9818 patents and using a lot of dictionaries made of keywords, both fraud-related and AI-related, we identified the 74 that appeared the most useful to our analysis. Our dictionary was quite generic in order to be sure not to leave out patents of interest. Then we checked the titles using another dictionary to perform an automatic filtering of patents that were false positives. We also added 30 patents selected manually from Google Patents (so they didn't need the false positives check). After saving the text of each patent in a list we applied NER to identify the entities. We focused on 'ORG' entities (Organizations) and we filtered some stop-words that were not real companies. In the end we showed the most commonly recognised companies.

2.2 Trend Question

Exploiting the structure of the 74 patents discussed above, we identified the dates and divided them in groups according to the year of publication. We counted the number of patents for each year and then we exploited NLP techniques to find the most common words over the years. In this way we observed the differences of the main topic across the years and also the growth of this technology in the given time interval. See Figure 1.



(a) **(b)**

Figure 1: Plots that represent the growing trend of patents published over 5 years.

2.3 Main Topics Question

As previously discussed, in order to perform the Topic Analysis we used an extended list of patents. The patents' text was preprocessed via NLP techniques; after that, we computed a wordcloud, in order to be sure that the preprocessing of the text produced a reasonable set of meaningful words. In the end, we represented the results of the analysis in an interactive Intertopic Multidimensional Map, which shows the distribution of the patents' most common keywords in each topic, and the dependencies within them.

2.4 State of the Art Question

Natural Language Processing techniques were applied to 33 surveys selected from Google Scholar. Those surveys are all focused on the analysis of the State of the Art methods for Fraud Detection using AI but, given the fact that this field improved really fast during the years and the selected articles were published in different periods, we can find a lot of different methods in them. We found most common methods in those surveys in order to establish the most recognized State of the Art methods among the time interval of our articles.

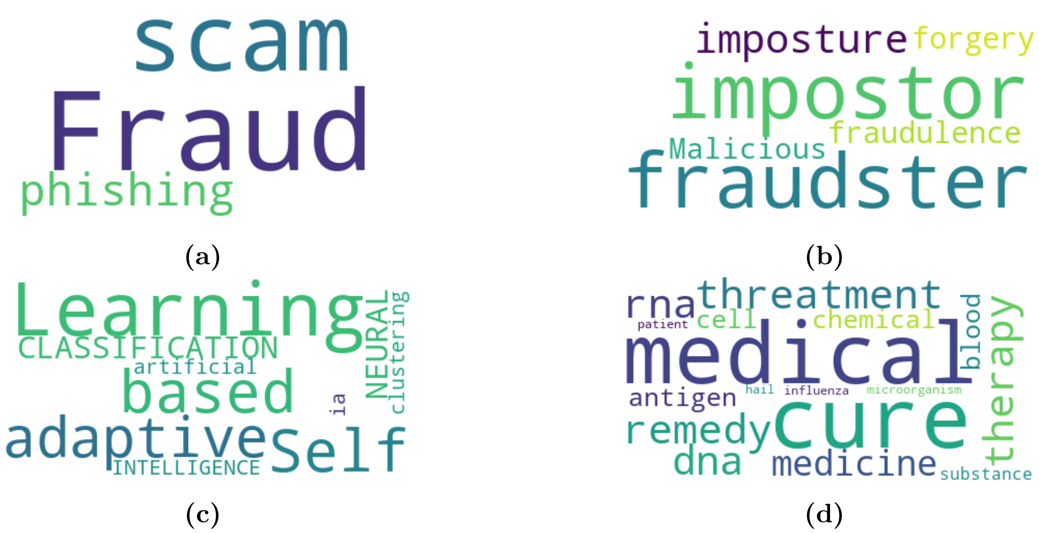
3 Technical Report

In this section we discuss in details the Python implementation for each of the questions of interest for our analysis.

3.1 Preparation

Before answering any of the mentioned questions, some preparation steps were taken:

- Dictionaries Library: in order to make the code look smoother, we defined a Python library containing seven dictionaries. In the first one we have words that are heavily related to Fraud Detection. In the second we have more words about fraud detection that could be interesting but that could also make the query include some false positive, this was necessary to be sure to not exclude false negatives. In the third we have words that are related to the AI field. In the fourth we have terms that are strictly from a medical context, this was useful because we noticed that in our data set there were lot of medical patents and a lot of false positives were from a medical context. These first 4 are represented as wordclouds in Figure 2. In the fifth, there are words that were wrongly labeled as Organizations by the NER algorithm we used, most of them are IT related terms (e.g. DNS, IP, HTTP); in the sixth, we have words that were meaningless in the production of the Wordcloud discussed above and, consequently, in the Topic Analysis. In the last dictionary, we put a large collection of fraud-related terms in order to perform a test on the False Positives Candidates' titles: if neither of these terms were contained in one of the titles, the relative patent was assumed to be a False Positive.



(a)

(b)

(c)

(d)

Figure 2: Wordclouds with the terms used in the dictionaries built to select the patents.

- Patents Selection: here we saved the patents that seemed to be related to our technology; the criteria was the presence of words from our dictionaries, only the patents which had bot words fraud-related and words AI-related were included. We did a core assumption: the first one was that our medical dictionary was strict enough to assume that if **only** words of the medical dictionary were present in the patent, it had to be discarded. Using large dictionaries guaranteed us to reduce as much as possible the number of false negatives (patents that were discarded though could have been useful).

- False Positive Control: here we made the other core assumption that if any patent contained at least one word form the strict fraud-related dictionary, it was considered a True Positive. If the patent contained less than three words from the large fraud-related dictionary and three from the AI-related dictionary, it was considered a False Positive Candidate. Using the really large fraud-related dictionary realized for this task, we performed an automatic check on the titles of the False Positive Candidates and the ones that didn't have words from this list were removed.

In addition to those steps, the text of the selected patents was preventively preprocessed, using various NLP techniques and Python Libraries; not all the preprocessing procedures were used in each of the questions of interest for this project, but we report them all here for readability's sake. First of, the text was read line by line and regrouped in a list of strings. Then, the following operations were performed on it:

- Lowering: we applied the lower() function to obtain lower-case text strings. This was necessary because some of the functions we used are case sensitive, and it could be potentially problematic to have the same word listed differently twice, or even more, because of that.
- White Spaces Removal: we defined a function with the goal of removing the extra white spaces from the text and we applied it to our list. This is a good practice that avoids a good number of meaningless results when computing the text analysis.
- Tokenization: we performed the tokenization of our text exploiting the nltk.word_tokenize() function, from the Python nltk library (Natural

Language Toolkit). This process generated a list of words starting from a list of sentences.

- Stop Words Filtering: we applied the stop words filtering using the standard ‘english’ dictionary of stopwords with some customizations to be sure to exclude a lot of words that were frequent but weren’t topics of interest. Those customizations were implemented in a dictionary included in our Python library for this project.
- POS Tagging: The part of speech tagging was performed using the ’averaged_perceptron_tagger’ given by the already mentioned nltk library: given our tokenized text as an input, we got each token labeled with its own part of speech. After that, we exploited to fact that the punctuation is the only part of speech that is labeled with a single letter to remove it from the text. In the end, we defined a function to simplify the tags, reducing all of them to just a single letter in order to optimize the memory occupation.
- Lemmatization: the lemmatization was performed using the WordNetLemmatizer given by the nltk library. With this step, the words coming from the same root, such as plural forms or different verb declensions, were all grouped under the same term in its basic form.
- Stemming: the last preprocessing step taken was the stemming, performed using the PorterStemmer module of the nltk library. This process consisted of the removal of the suffix from each lemmatized token, in order to keep just the root for each word.
- List to String: finally, we converted the list of tokens obtained after the preprocessing in a text string, which is a data format required by a module used further in the analysis.

3.2 NER Question

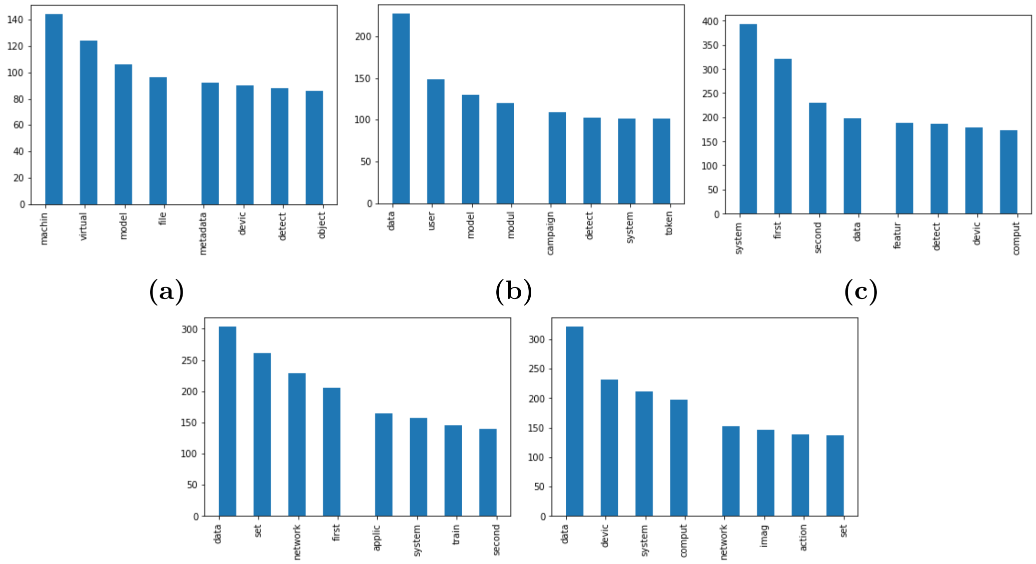
The NER analysis was performed using the Spacy library. In particular, we used the “en_core_web_trf” module, which is a name entity recognition model that uses a pre-trained english pipeline. We tried a large variety of pipelines, and we selected this one as it was the best by far, recognizing almost every organization with a minimum number of false positives, even if the NER procedure took quite some time. After the analysis was performed on the whole unprocessed text, we noticed that some words were still mislabeled: most of them were IT specific terms recognized as organizations. In order to solve this problem, we realized a dictionary of forbidden words, which was put into the relative library, as previously discussed. In the dictionary there are also terms related to the fact that our data set is composed of patents, such as "'U.S. Patent".

3.3 Trend Question

The Fraud Detection Patents' through time analysis was realized according to the following steps:

- Saving dates: the selected patents had a common structure, so we could exploit the tag <filling_date> in order to save the dates of all of those left after the selection process.
- Plotting trend over time: we counted the number of patents for each year and plotted it both in a line-plot and in an histogram to show the increasing trend of AI in Fraud Detection.
- Merging per year: now we wanted to focus on the different main topics over the years, so we merged together all the patents corresponding to the same year in one text.
- Text Preprocessing: the preprocessing steps were performed on each of the texts built in the previous point.
- Counting: we saved for each year the most occurring words and their occurrences and we built a dictionary (python dict) out of them.
- Plotting: we plotted the five histograms corresponding to the five years in which our patents were registered, showing the frequent words for each year. We also plotted an histogram made up of the histograms

of each year to observe if some words were frequent in more than one year. See Figure 3.



(a) **(b)** **(c)**

(d) **(e)**

Figure 3: The 5 histograms showing the most frequent words over time in the selected patents.

3.4 Main Topics Question

The Main Topic Question was addressed using mainly 2 Python Libraries: ’gensim’ and ’pyLDAvis’.

- Patents Selection: for this specific question, we used the extra patents manually selected from Google Patents, in order to have a clearer representation of the main topics of interest when the Fraud Detection issue is mentioned in the patents.
- Text Preprocessing: in this case, the whole preprocessing procedure was required. In addition to that, the required format for the analysis is a list of lists of tokens, where each of the lists is a preprocessed patent.

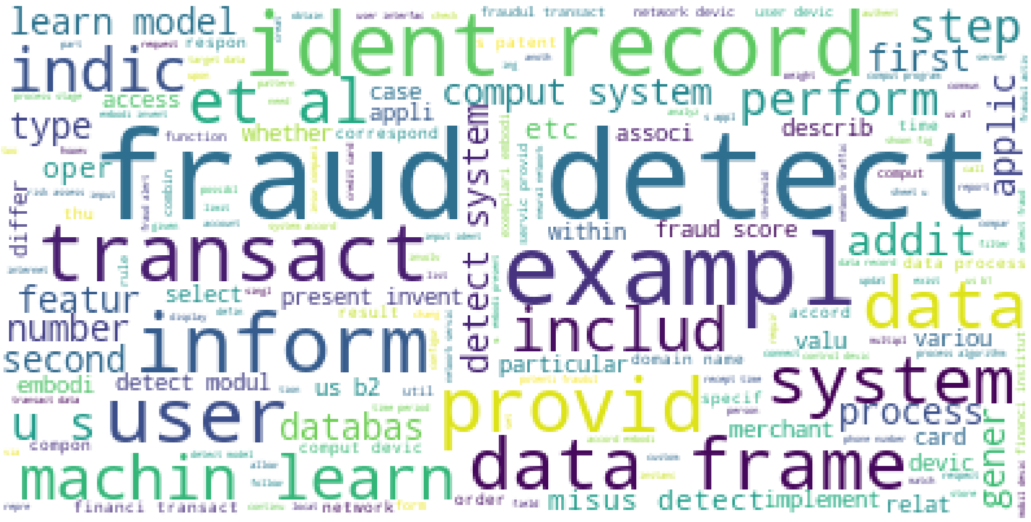
This was required because a relation between topics and patents is present in the model adopted.

- Prior Representation: in order to understand if the filtering done during the preprocess phase was enough, we used the Python library 'wordcloud' in order to visualize the most common terms within the text. The results showed some terms that survived the filtering, so we built a dictionary to improve the text cleaning. See Figure 4.
- Topic Analysis: the proper analysis was executed via the 'gensim' library, exploiting a Latent Dirichlet Allocation Model (LDA), which is a popular topic model. In order to establish the input number of expected topics within the selected patents, we made some tentatives and found out that the optimal number was . The gensim library allowed us to perform the analysis in an automatyzed manner, given the correct format for the input text.
- Results Visualization: The results were visualized with the help of the pyLDAvis library, that allowed us to generate an interactive multi dimensional intertopic distance map; the map highlights, for each topic, the 30 most common words and their distribution with respect to the total occurrences within the patents. The Topics are visualized as circles in a 2-Dimensional plane, and have different size and positions with respect to the axes and to the most frequent words within them. The relevance-metric for each term can be interactively changed via a parameter called λ.

3.5 State of the Art Question

The implementation of the last question goes as follows:

- Dataset: in this case, we used the 33 Surveys taken from Google Scholar. The common topic is the analysis of State of the Art methods for fraud detection applied to many fields.



(a)

Figure 4: The wordcloud realized in order to be sure that only meaningful terms were used in the Topics Analysis.

- Files Reading: the next step was to read the articles, downloaded in PDF format; we opened them as a document exploiting the 'fitz' library and saved the text of each page in a string. At the end of this step we obtained a list, with lenght equal to the number of surveys, in which each element is the entire text of each article. In the end, we merged all the elements into a single string.

- Text Preprocessing: other than the usual preprocessing procedures, for this question an additional layer of cleaning was required, since some useless words and single letters were isolated in the text; we did so by means of a regex. After that, a second layer of stop words filtering was applied as well: our task was to find methods, and the surveys were rich of scientific terms that didn't match our goal (e.g. 'et al').

- Counting: in order to find the methods, we needed to count the occurrences in our text. This was performed exploiting the Counter module.

- N-grams: until now we counted occurrences of single words. Obviously this isn't the right approach because we want to find AI techniques and usually their names are not made up of a single word. We found the common 2-grams and the common 3-grams and we observed that the right approach was to use the 3-grams. This step was performed exploiting the 'ngrams' module of the nltk library.

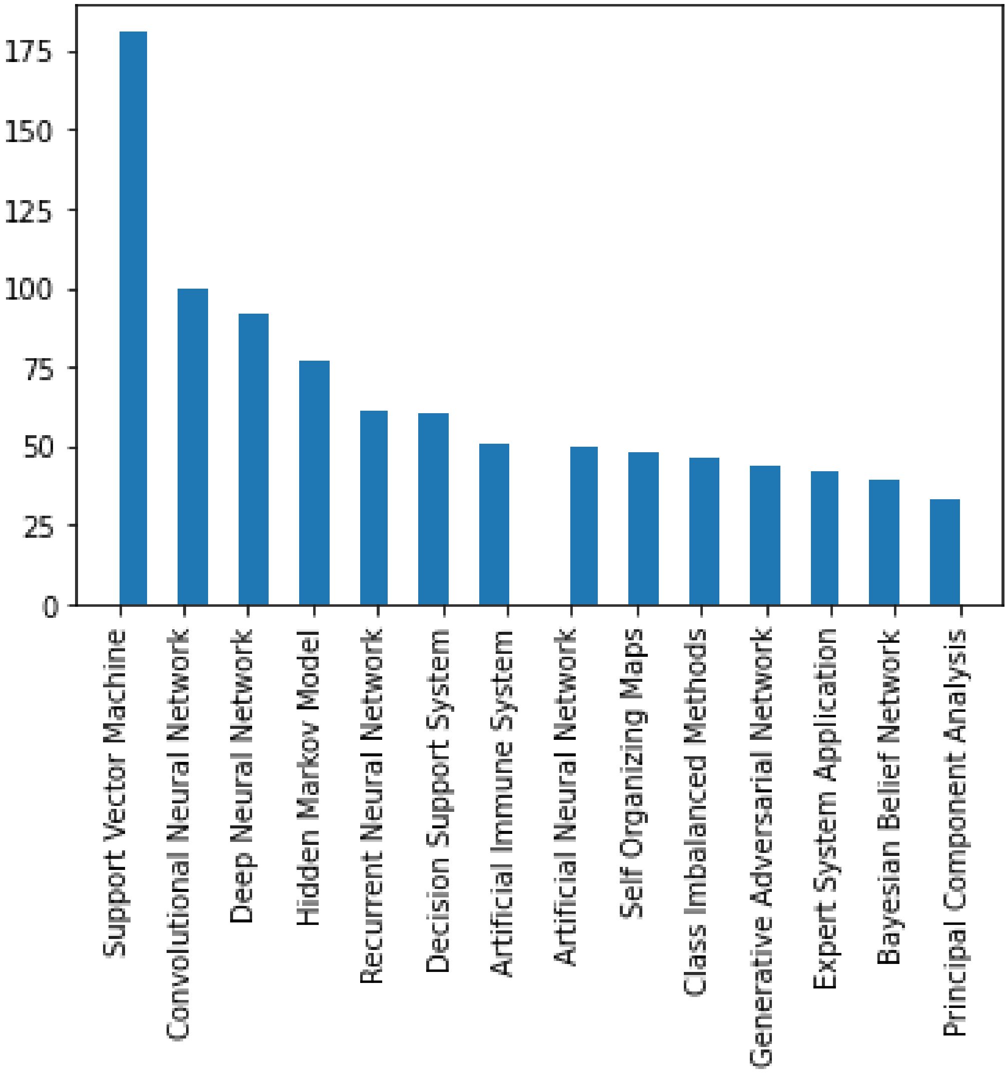
- Renaming: exploiting some knowledge of the field of interest, we renamed the preprocessed 3-grams using the correct name of the methods in order to show the real names instead of the stemmed and lemmatized version of them.
- Synonimy problem: during the previous step it was important to put attention to the fact that different authors can refer to the same methods in slightly different ways; for example, in our case we found that the Support Vector Machine method was called in two different ways. To solve this problem we called both common 3-grams in the same way, so we basically merged them togheter. Plotting: in the last step we plotted an histogram showing the 15 most common methods and their occurrences in the examined papers. See Figure 5.

4 Results Discussion

In the last section of this report, we discussed the results obtained for each question, highlighting some important use-cases and Business-related conclusions.

4.1 NER Question

We found that the most recurrent organizations were VISA, EVM (Europay Visa Mastercard), ATM, IBM, Mastercard and eBay. As expected, most of those names are banks or credit card services, which are the activities most in need for this kind of technology. It's interesting to observe that we have also IBM, which is an hardware producer that has oriented some of its new products to the cryptography, in order to make processors more suitable for the kind of application we analyzed. Another outlier is eBay, in



(a)

Figure 5: Occurrences per method found in the selected surveys.

which case is simple to imagine how such a model of Business could make use of the technology discussed. We could see that also companies with an industrial background are starting to take into consideration the problem of Fraud Detection, other then the natural users of this technology, like banks. This could be a sign that this field is getting more attention (fact that will be confirmed in one of the next Questions, in which will be highlighted the rising trend over time of AI applications for fraud detection). Finally, we exploited NER also to identify Geographical Entities(GPE), and we noticed that the main places related to our technology are: USA, Japan, China and UK. These results are reasonable considering that those countries are among the most advanced in the IT field.

4.2 Trend Question

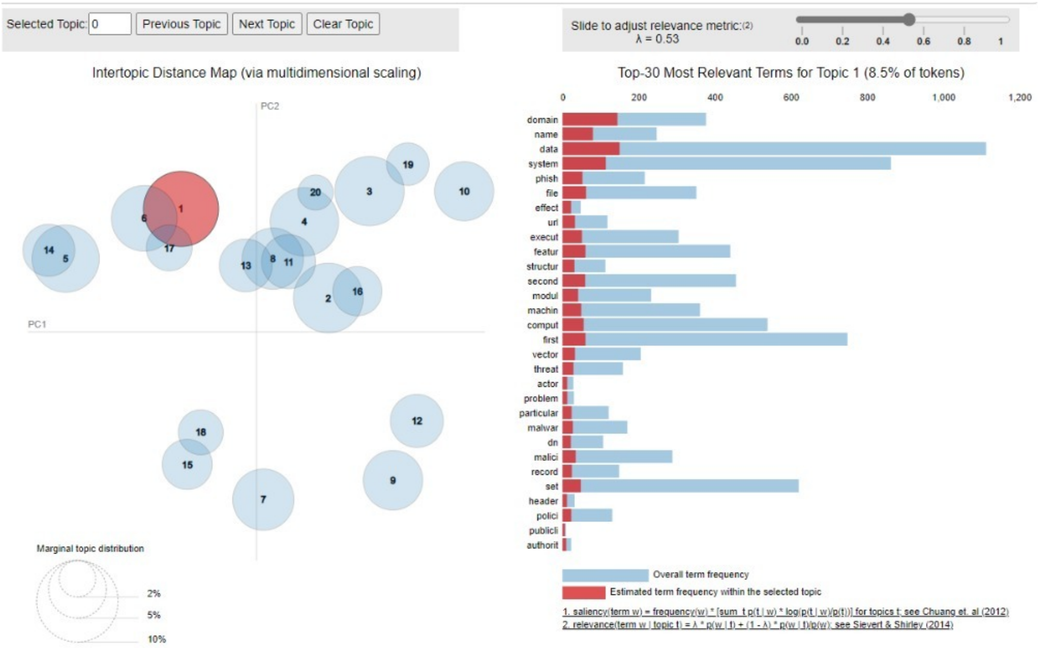
We noticed a clearly increasing trend of patents related to IA for Fraud Detection during the years from 2014 to 2020 : in 2014 we had 0 patents, just 2 in 2015 and then the technology started to explode, in 2017 we had 10 and in 2020 we had 19. We also noticed that we have some evolution over the years: for example, in 2020, 'image' as a frequent word started to appear; we looked into this matter and discovered that a new field of the Fraud Detection in fact is the "Image Forgery Detection". It's interesting to note that the data used to perform our analysis was extracted from patents and surveys: that means that we didn't only examine the research progress in the considered field, but also a concrete increase in the innovation process. In this case it's reasonable to think that this process was made necessary by an increase in digital frauds, which is a natural consequence of the digitalization that is happening in the last years.

4.3 State of the Art Question

We found that in the field of the Fraud Detection a lot of AI Techniques are used. The most common one is the Support Vector Machine, but this kind of result could be biased from the fact that this is an older method with respect to some of the others, so it's easy to understand why it appears in more papers; in addition to that, it's also a traditional Machine Learning Method, so it's often used as a baseline to compare different algorithms. An interesting observation is that different families of algorithms are used, both traditional Machine Learning and Neural Networks. This situation can be related to the fact that different families are more suitable to different quantities of data, so in fields in which we have less data older methods are still the State of the Art. The analysis performed can be useful both to people that approach this field from a less technical point of view, and to the ones more used to deal with the matter at hand. In the first case, it may help in the choice of the most suitable method for the problem one has to solve. On the other hand, one could also make use of the proposed analysis to choose the algorithms to compare his research results to.

4.4 Main Topics Question

The analysis performed on the selected data set to isolate the dominating topics among it showed us that such topics are, as expected, all IT related; it's possible to obtain a progressively more specific analysis for the topic distribution by varying the parameters of the model: the more topics we assume are present in the data, more specific they will be. The most common keywords within the topics show that most of them are based upon Network, Phishing, Machine Learning, Regression, SVM, Bank Transactions, Fraud Detection, Systems' vulnerabilities, Malwares, E-mails, Clusters, Cybersecurity, Online Games. It's also possible to observe an Image Forgery theme, as discussed in a previous section. Analyzing the topics that compose a dataset can be very useful for a lot of purposes: a company interested in expanding its production to the Fraud Detect Systems could quickly find out the main fields of interest of such technology and use that data to better schedule its future steps. Another use case is the learning potential behind the results shown: a person who wants to know more about the discussed technology, e.g. a student, could use them to establish which are the fields that are more likely to contain huge amounts of information about it. See Figure 6.



Selected Topic: 0

Slide to adjust relevance metric:(2) λ = 0.53

Previous Topic
Next Topic
Clear Topic
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 1 (8.5% of tokens)
0
200
400
600
800
1,000
1,200
domain
name
data
system
phish
file
effect
url
execut
featur
structur
second
modul
machin
comput
first
vector
threat
actor
problem
particular
malwar
dn
malici
record
set
header
polici
publicli
authorit
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

(a)

Figure 6: The interactive map realized to visualize the Topics Analysis. In the picture, the most prevalent topic is highlighted.