Detection of Fraud, Abuse and Cost Inefficiencies in the National Healthcare System

Francisco Guerreiro Gomes Pedreira
Instituto Superior Tcnico
University of Lisbon
Lisbon, Portugal
francisco.gomes.pedreira@tecnico.ulisboa.pt

***Abstract*—In todays world healthcare fraud and abuse has taken enormous proportions. Every year millions of dollars are lost in the healthcare system due to fraud. It is therefore important to develop systems that can efficiently combat and prevent this behaviour. This paper proposes a solution for the problem of detecting fraud and abuse in the healthcare system. The chosen approach can be divided into three main phases. In the first phase, a data generator is used to create a dataset composed by prescriptions and geographical distributions. In the second phase, two different algorithms are used to detect fraudulent behaviour, namely a geo-location based and a rule based algorithm. These algorithms use a supervised learning approach and are evaluated using a variety of metrics, specifically accuracy, con- fusion matrix, precision and recall, and K-fold cross validation. In the last phase, the output of both algorithms is analysed and the fraudulent claims are sent for further investigation. Using the prescriptions dataset the obtained value for the recall, when using 10-fold cross validation, is of 0.33 which is not a very high value. However using the geographical distributions dataset a recall of 0.69, again using 10-fold, is obtained which is a good result for this measure. A novel approach is taken in this work by designing and implementing a data generator that can create data for a plethora of algorithms. Using this approach allows for cost reduction since only a few cases, the ones identified by the algorithms as fraudulent, will be sent for further analysis, resulting in a cheaper and more efficient service.**

Keywords*—*Healthcare Fraud, Fraud, Abuse, National Healthcare System, Detection of Fraud

I. Introduction

In the modern world, fraud is present in many fields such as banking, finances, business practice and healthcare. One example is identity theft [1], in which criminals use stolen identities to steal money from bank accounts, claim eligibility for services or hack into networks without previous authorization. Another example is financial fraud. Millions were lost to this kind of fraud as shown in [2]. A specific example is when criminals hold money or other property from the company, causing loss and damage [3]. Credit card is also a very popular target for criminals as shown in [4]. The credit card industry is estimated to lose about $2 billion a year globally due to fraud [5]. Here, credit card information is stolen and used to pay for services or goods in behalf of the original owner. In 2004 alone, in the U.S.A, 800 million dollars were lost in fraud.

Inside companies there are also other types of fraud committed which involve business practices. In [6] it was reported that in two years, 8.3 million dollars were overcharged by a large number of taxi drivers who deceived their costumers by arbitrarily modifying their taximeter.

In the healthcare system, fraud generates a loss of over $30 billion annually to healthcare insurance frauds [7].

As has been seen, there are several kinds of fraud, mainly credit card, financial, business practice and, last but not least, healthcare fraud.

The last will be the matter of this work.

A. General Goals

The main goal of this work is to implement an application that can detect fraud in the national healthcare system. In order to accomplish this goal, the following objectives must be fulfilled:

- Design and implement a data generator.
- Define two algorithms to be used in the detection of fraud.
- Analyse the output of the algorithms and decide which claims to send to further investigation.

The rest of this work is divided as follows. In section II the existing literature will be discussed. Section III discusses the designed architecture. Section IV presents the metrics used to test the proposed solution and the results obtained from applying the chosen metrics and finally Section V presents the conclusion of this work along with some improvements that can be added in a future work.

II. Related Work

Health care produces massive amounts of information in todays world. And in order to process this data automated algorithms must be used, since humans cant process it timely. Some examples of developed algorithms can be found in [8], [9] and [10].

Having this in mind, and knowing that the problem at hands is detecting fraud, statistical methods were developed to help tackling the problem. They are divided in two main methods: supervised and unsupervised.

The supervised methods are used to give the system the ability to recognize which cases are legitimate and which are fraudulent. In order to achieve this, these methods require prior knowledge, that is, they require all the cases in a training

dataset to be labelled, as either legitimate or fraudulent, by experts. This way, the system is trained to understand how to classify a new case and so it will know to which class this new case belongs when it is introduced in the system.

The unsupervised methods are used to cluster similar cases into groups. This allows the system to find outliers in the data since they will be aggregated in the same group. Unlike supervised methods, unsupervised do not require previous data, but because of this they cannot detect known cases of fraud. Hence it is important to use this type of methods along with the supervised ones to ensure that the largest number of fraudulent cases are detected.

An important aspect in fraud detection is understanding who might be involved in the criminal activities. According to the authors of [11] there are three main participants: service providers, insurance subscribers and insurance carriers. Service providers can be doctors, hospitals, ambulance companies or laboratories. Insurance subscribers are patients and patients employers. Insurance carriers include governmental health departments and private insurance companies.

Among all the mentioned participants, service providers account for the majority of fraud committed.

Before introducing the main types of fraud and their perpetrators, a brief explanation of machine learning will be presented.

[11] was used as a reference to write a big part of this section since it was one of the most cited papers in the literature, containing very good information on the topic of this paper.

A. *Types of Fraud*

In this section the main types of fraud will be introduced. There are three major entities involved in the criminal activities, namely, service providers, insurance subscribers and insurance carriers. However, service providers account for the vast majority of the fraud committed. Bellow, a list of the various types of fraud, as described in [11], will be presented.

- Service providers' committed types of fraud:

1) **Billing unperformed services**: billing services, which were not performed, to the insurance company, in order to obtain profit;
2) **Performing unnecessary medical services**: performing medical services which are not required by the patient, in order to generate insurance payments for those services;
3) **Falsifying non-covered treatment**: falsifying a non-covered treatment as a medically necessary covered treatment in the interest of obtaining insurance payments;
4) **Unbundling**: billing each stage of a procedure as if it were a separate treatment;
5) **Upcoding**: billing more costly services than the ones actually performed (for example, classifying a patients illness into the highest possible treatment category in order to claim a higher reimbursement);
6) **Tamper with the diagnosis**: tampering with the patients diagnosis and/or treatment histories in the interest of justifying tests, surgeries, or other procedures that are not medically necessary;

- Insurance subscribers' committed types of fraud:

1) **Forging records of eligibility**: forging records of employment and/or eligibility in order to obtain a lower cost insurance;
2) **Filing claims for unreceived services**: filing claims for medical services which are not actually received;
3) **Illegal use of cards**: using other person's coverage or insurance card to illegally claim the insurance benefits;

- Insurance carriers' committed types of fraud:

1) **Falsification**: falsifying reimbursements;
2) **Forgery**: forging benefit/service statements;

In addition to the types described above, there is another newly emerging type called conspiracy [11], which involves more than one party. An example is a patient and a physician fabricating medical service and transition records in order to defraud the insurance company to whom the patient subscribes.

Bellow a graphical representation of some types of fraud will be presented to allow for a better understanding.

In Fig. 1 we have an example of unperformed services being billed to the insurance company. This happens when a service provider introduces new services in the bill, which were not actually performed on the patient, in order to obtain extra income from the insurance company. In Fig. 2 we have a scenario in which tests that are not necessary to the patient are conducted and billed with the purpose of, once again, obtain extra revenue.

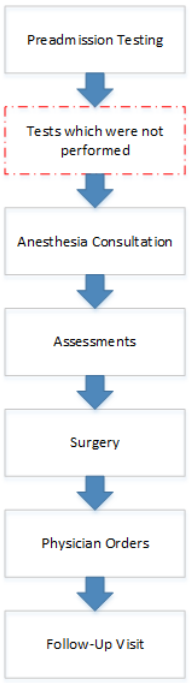

Fig. 1: Billing unperformed services

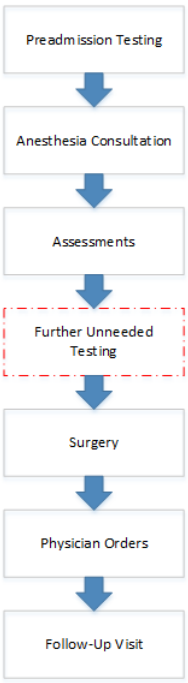

Fig. 2: Performing unnecessary medical services

B. Supervised Methods

As was shown before, supervised methods are used to effectively detect fraudulent behaviour in the healthcare system. The authors of [11] mentioned neural networks, decision trees, fuzzy logic and Bayesian Networks as the most common supervised methods. Similarity learning is mentioned in [12], clinical pathway is seen in [13] and Support Vector Machine is presented by the authors of [14]. The authors of [15] implemented a **system based on rules** that checks for missing or invalid data and also the medical validity of the prescription. It used two sets of rules, namely administrative and medical rules. Administrative rules were used to detect missing or invalid data in the initial phase, that is, when the data was introduced into the system, while medical rules were used to check for the medical validity of the prescriptions.

C. Unsupervised Methods

Unsupervised methods, on the other hand, are used to cluster data into groups. Different groups represent different characteristics. These methods are a crucial step to identify new groups that have never been found (which in the case of this paper represents a new type of fraud). Several methods have been applied in the literature as seen in [7], [16] and [17]. A geo-location based algorithm was implemented in [18]. It consisted in three main phases: preparation, data preprocessing and analysis. The preparation consisted in collecting longitude and latitude from beneficiaries and service providers and mapping these locations to an SSA code, i.e., State Code from Claim. In the data processing phase the Euclidean distance between beneficiaries and service providers was calculated, then using the diagnosis as a control variable, three datasets with the three more common diagnoses were created and finally the payment amount and the distance were used in a clustering model in order to find cases where either the payment amount was high, the distance was high or both.

In the following section, the architecture of the solution will be presented and discussed.

III. Architecture

The main goal of this section is to define how the architecture is structured and organised. In order to achieve this objective, an overview of the architecture will be presented, followed by a comprehensive description of each module. In this architecture a data generator will generate a dataset, that will then be analysed by two algorithms, present in the Data Analysis module, as to detect fraudulent behaviour and finally the output of those same algorithms is shown to the user so that it can then be sent to experts for further analysis.

The architecture consists of three main modules: Data Generator, Data Analysis and Output Analysis. The Data Generator is used to generate all the necessary data to use in the Data Analysis module. The Data Analysis module is responsible for the detection of fraudulent patterns and behaviours. And finally the Output Analysis module gathers all the information regarding the fraudulent cases, groups them by type of fraud and presents the result to the user. This result would then be sent to experts for further investigation. A simple graphical representation of the architecture can be seen in Fig. 3.

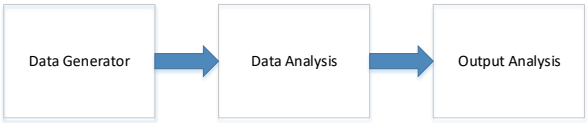

Fig. 3: System Architecture

A. Data Generator

A data generator (DG) was designed and implemented in order to generate a plethora of information. This information is then used by the detection algorithms to discover fraudulent cases. This approach is interesting, since it allows different algorithms to be tested using a single dataset that can be changed simply by executing the Data Generator program again.

The DG module was designed to generate all the necessary information to be used by the Data Analysis module. This includes a database containing all the possible values for each field and several methods to manipulate this data in order to create a new dataset every time the algorithm is executed. It is composed by several smaller components. These components are represented in the solution as classes.

A brief description of each of the composing classes is shown below.

1) **DataGenerator** - uses all the other classes to create the dataset;
2) **CANRangeBetween** - generates a Claim Account Number (CAN), one of the two numbers that constitute the HICN (Health Insurance Claim Number);

3) **BICGenerator** - generates a Beneficiary Identification Code, used in conjunction with the CAN to create the HICN;
4) **Diagnosis** - generates a random diagnosis;
5) **Doctor** - generates a random doctor with an associated NPI (National Provider Identifier);
6) **Drug** - generates a random drug which can have a normal price or a dubious price depending on the odds;
7) **GenerateMapDrugDiagnosis** - generates a HashMap were each drug corresponds to one specific diagnosis, where there can be a valid or invalid diagnosis depending on the odds;
8) **Location** - generates a random location in the Pennsylvania state;
9) **LocationGenerator** - used to generate locations within a certain distance from an origin point;
10) **NPIGenerator** - used to generate an NPI;
11) **OfficialDrugPriceListGenerator** - used to generate the official drug price list to which every drug price will be checked against;
12) **Patient** - generates a patient;
13) **RandomDate** - used to generate random dates for the prescription;
14) **TextFileReader** - used to read information from text files that will be used to create the final dataset;

In the following lines the more critical classes will be presented and their functionality discussed.

1) Diagnosis: This class is used to collect all the possible diagnoses, aggregate them in an ArrayList and send them to the GenerateMapDrugDiagnosis class, where they will be matched with their respective drug. To do this, a method is used: getIllnessList(). This method reads a text file which contains a list of the top 200 drugs in 2016 [19] and it's corresponding purpose, which are used as a diagnostic in the solution.

2) Drug: This class is, similarly to the Diagnois class, used to collect all the possible drugs in order to send them to the GenerateMapDrugDiagnosis. Using the data in [19] a list of all the drugs is collected. All the drugs are paired with the corresponding diagnosis.

3) GenerateMapDrugDiagnosis: In this class the HashMap that contains all the drugs and their correspondent diagnoses is created. There are three possibilities when using this class. One is to generate a normal case, where a random drug is selected along with it's corresponding diagnosis. These values are used as the drug and the diagnosis in the prescription being generated at the moment. The second case scenario is when a drug is selected randomly but, instead of selecting the corresponding diagnosis, another diagnosis is randomly chosen. Finally the last method creates the HashMap where all the drugs are correctly associated with their respective diagnostic.

4) Location: This class is responsible for generating all the locations in the dataset. The first decision to be made was to choose from which country to collect information. The United States of America was chosen since it was simpler to get geographical information related to that country. Following the decision of the country, a state had to be chosen to collect the locations from. At first the New York state was considered, but soon one problem arose: the state has many geographical positions where there are watercourses. This is a problem because, as shown in the next class, there is a need to generate locations to a certain distance from an origin point. And it is important that they are not watercourses because they represent important geographical points, such as a residence or a pharmacy.

Therefore it was decided that the Pennsylvania state would be a better fit. Besides having a greater area it has close to no watercourses. It was decided that the locations would be randomly selected from the state of Pennsylvania.

5) Data Generator: This class is the one where all the other classes are used to create all the necessary pieces that form the dataset, i.e., where all the prescriptions and geographical distributions are generated. In order to achieve this, several methods were incorporated in this class.

The main method is called generateData(). It is important to point out that in all cases mentioned from now on there is a 2% chance of a fraudulent case being generated. This percentage is used based on [22] where the authors obtained values between 1% and 3% of fraud detection, therefore making 2% the average of these values and a realistic value. Being that most results are between 1% and 3%, 2% seems to be a good value for the topic of this work.

It starts by generating the HashMap with all the drugs and correspondent correct diagnoses and storing it in the database.

The next step consists in the creation of the dataset itself. A for loop is initiated, where the number of iterations is dictated by the numberOfRecords variable.

The first piece of data created are the issue and execution dates. These can be generated in two ways, or better, using two intervals: 0 to 5 or 6 to 7. The first interval will randomly generate dates which have 0 to 5 days of interval between them. Whereas if the interval is between 6 and 7, the generated dates will have those intervals, one or the other.

Next the drug and the diagnosis are created. There are two possible instances: either the drug and the diagnosis match or not. The first case is the legitimate one since the rule in the rules based algorithm states that there can not be a greater interval than 5 days in between dates. The second is the fraudulent one since it violates the rule.

The drug price follows but in this case there are three possible outcomes: normal case, fraudulent case and dubious case. The fraudulent case occurs when an incorrect price is obtained. The incorrect price is calculated by adding 5 plus a random number between 5 and 10 to the original price. This method of calculation was used to try and emulate a real case scenario since in the real world the difference can not be big to avoid detection. The dubious price was a special case generated by the reasons aforementioned and a decision was made to add a value of 2 to the original price since it was lower than the minimum value created for the fraudulent price and it is was still a very low difference to the original price. Since the detection algorithm can not simply detect if the price is the same as the original price, an assumption had to be made that, in order for a price to be considered fraudulent, it had to be in between an interval which was the

original price plus minAdd and the original price plus maxAdd, where minAdd and maxAdd are the inferior and superior limits that delineate the range of a fraudulent price. The following formula, Formula 1, demonstrates the assumption:

$$\begin{aligned} &originalPrice + minAdd \\ <= &\ fraudulentPrice <= \\ &originalPrice + maxAdd \end{aligned} \tag{1}$$

The minAdd variable may contain values between 3 and 10 inclusive, since 2 or less will affect the dubious case detection rate. The maxAdd variable can range between 11 and 15 inclusive.

Finally the normal case occurs when the drug price matches the official price, i.e., the one present in the official price list.

After the price is obtained, the patient's contribution is calculated as being 30% of the total cost of the drug, meaning the remaining 70% is contributed by the SSB.

The SSB has a normal participation percentage of 70%. However this is not always the case. There are four possible cases.

The first one is when the total cost of the drug is zero for the patient, in which case, since there is no real dataset with the actual total value of a specific drug, a value must be assumed for this type of scenarios. The assumed value is represented by the variable assumedPrice, which can take different values and is very easily changeable within the solution. The assumedPrice variable can range between 5 and 15. The second case is the fraudulent one and here the SSB's participation percentage is reduced from 70% to 50% giving the SSB 20% of the money. This is done by multiplying the total drug cost by 50%. The third case is the dubious one and here the participation percentage is reduced from 70% to 60% giving 10% to the SSB. As the previous one, this value is represented as 60% of the total drug cost. The fourth and final case is the normal case, where the percentage is 70%.

In the following lines three variables are created: personHICN, pharmacyNPI and doctorNPI. Here there are also four different outcomes. A fraudulent case is represented as one of the three variables missing in the prescription. So, for every fraudulent outcome, one of the variables is missing. The fourth case is when all the variables are present, therefore the legitimate one.

Next the doctor is created. When creating the doctor, an NPI is associated with it and a variable with his SSB's status is set with the value true. Every doctor is created belonging to the SSB initially although there are cases, as will be seen later, where the status changes.

The next cases are related to the issue and execution date. In this case, a fraudulent set of dates is one where the execution date precedes the issue date.

After the dates are created, the patient is generated. Upon the creation, one HICN is attributed to it. After this, the patient's residence is generated along with it's workplace and both are attributed to the patient.

The next set of attributes are the latitude and longitude of the starting point to the pharmacy, i.e., they represent the starting point of the geographical distribution that is to be generated. As mentioned before the starting point can be one of three locations: patient's residence, patient's workplace or a hospital. These were chosen since they are believed to represent the three most common situations a patient encounters himself in. Having three distinct places to choose from, it is necessary to decide which places will be more likely to be chosen, since it does not make sense that the workplace has the same probability as the hospital to be selected. Therefore it was decided that the workplace had a very low chance of getting selected, represented by the chanceWorkplaceStartPoint variable, the residence had a bigger chance but still low of being selected, represented by the chanceResidenceStartPoint variabe and finally the hospital had the remaining chance, represented by the chanceHospitalStartPoint. The chanceWorkplaceStartPoint can range between 1% and 19%, the chanceResidenceStartPoint can range between 20% and 69% and finally the chanceHospitalStartPoint can range between 12% and 79%.

If desired it is easy to set the presented variables to different values within the solution. The locations are generated as seen in III-A4.

Following the selection of the starting point come the cases where the type of geographical distribution is decided. There are three outcomes: fraudulent, dubious and normal case. The normal distribution occurs when most patients go to the pharmacy which is closer to the starting point, some go the second closer and very few go to the furthest. The fraudulent case however inverts this tendency, by having a greater number of patients going to the furthest pharmacy. The third scenario is a dubious one where 25% go to the second closer and furthest pharmacies and the remaining 50% go to the closest one. This case is dubious since there is an even amount of patients going to the more far away pharmacies and a majority of them goes to the, expected, closer pharmacy. Once again this case was created to test the accuracy of the algorithms.

The distributions are generated by using the starting point and creating 100 different locations to a certain distance from it. The distance can be of 3, 5 or 10 Kilometres. Depending on the type of distribution, as mentioned, there will be different scenarios. In the legitimate case, the following distribution is generated:

- 3 Km normal3KmPerc
- 5 Km normal5KmPerc
- 10 Km normal10KmPerc

In this distribution the normal3KmPerc variable can range between 60% and 70%, normal5KmPerc can range from 20% to 30% and finally normal10KmPerc can range from 0% and 10%. The normal10KmPerc can not take a 20% value since that case is considered to be a dubious case, like will be seen in the following lines.

This distribution was chosen since it is an approximation to the normal case scenario where most of the patients would choose the closest pharmacy, whereas only a small percentage would go to a far away pharmacy.

The fraudulent distribution, however, generates a different pattern:

- 3 Km fraudulent3KmPerc
- 5 Km fraudulent5KmPerc
- 10 Km fraudulent10KmPerc

In this distribution the fraudulent3KmPerc variable can range between 40% and 50%, fraudulent5KmPerc can range from 20% to 30% and finally fraudulent10KmPerc can range from 30% to 40%.

In this case the distribution is clearly unbalanced as there is a greater percentage of patients going to the furthest pharmacy, clearly indicating a suspicious behaviour that could involve fraudulent activity.

At last there is the dubious distribution. The distribution is as follows:

- 3 Km dubious3KmPerc
- 5 Km dubious5KmPerc
- 10 Km dubious10KmPerc

Finally in this distribution the dubious3KmPerc variable can range between 50% and 70% and both dubious5KmPerc and dubious10KmPerc can range from 25% to 15%.

As mentioned above, this case is difficult to classify into either legitimate or fraudulent since there is an even distribution of the patients for the two more far away pharmacies, making it harder to assess whether there exists fraudulent activity. In Section IV the effectiveness of the fraud detection algorithms will be tested using this and the two dubious cases mentioned above.

The final data to be generated is the SSB's status of the entities represented in the dataset, which are the patient, the doctor and the pharmacy. As in previous data generation, there is a legitimate and a fraudulent case. The fraudulent case occurs when one of the three mentioned entities does not belong to the SSB. In the solution the doctor, the patient and the pharmacy have a boolean attribute that represents this status. In the end all the data is stored in the database.

B. Data Analysis

Analysing data for fraudulent behaviour is not a simple process since there may be cases where the data indicates a suspicious pattern but in reality it was just an abnormal case where one patient used a far away pharmacy simply because he was on vacations and bought his prescribed drugs at a pharmacy near his holiday location. In other words, there may be a potential scenario where a patient goes on holidays in a place that is faraway from his residence or local area and needs to buy a drug. In this scenario he will go to the nearest pharmacy in his holiday area and make the purchase. According to a geo-location algorithm that detects distances from a residence, or other relevant starting point in the local area of residence, to a pharmacy this would be considered a fraudulent behaviour since the measured distance would be considerably larger than the normal case scenario where a patient goes to the nearest pharmacy available. This case would then be reviewed by an expert and the conclusion would be that, in fact, it was an anomalous case however a legitimate one. An example of this behaviour can be seen in Fig. 4.

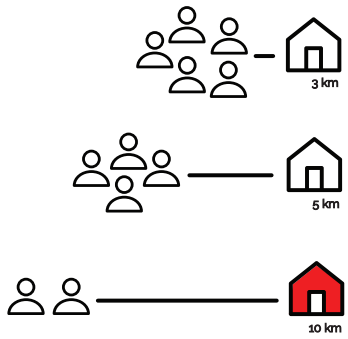

Fig. 4: Example of a possible fraudulent behaviour

In order to solve this problem, algorithms were developed to more accurately detect such patterns and behaviours. Of the many existing algorithms in the literature two were chosen: one based on medical and pharmaceutical rules and another based on geo-location. The first one analyses prescriptions for missing or incorrect data. The second one was adapted from the original algorithm present on the paper and instead of checking only the distance, it checks for patients' distributions along the various nearby pharmacies.

In the following subsections both algorithms will be analysed and explained in greater detail.

1) Medical and Pharmaceutical Rules Based Algorithm: The algorithm starts by retrieving the official drug price list and the list of drugs and corresponding diagnosis and stores them in two ArrayLists to facilitate future access.

The first method, checkExecDatePrecedesIssueDate(), corresponds to the rule that states the execution date may not precede the issue date present in the prescription. There are three case scenarios: the days are on the same month and year, the days are on different months and same year and the dates are in different months and years. In the first case the only necessary step is to subtract the execution date day from the issue date day and check the result. If the result is negative then it means there is a fraudulent case. For example 01/08/2016 is the execution date and 04/08/2016 is the issue date. If we subtract 1 to 4 we get negative 3 which is below 0 and thus a fraudulent case. In the second case the same principle applies except this time to the month. Lastly there is the scenario where the month and the year is different. The same logic can be applied to this case.

The next method, checkFiveDayExecutionPeriod(), as the name suggests, coincides with the rule that states there can not be a bigger interval than 5 days in between the issue date and the execution date of the prescription. In order to validate this rule, a similar approach to the first method was taken with

the difference being that here it is checked if the difference between the days is greater than 5, that is, 6 or more.

The third method is checkMissingCode(). It corresponds to the rule that states a prescription must include the insured person's, pharmacy's and doctor's codes. In this work the person's code is represented by the HICN while the pharmacy's and doctor's codes are represented by an NPI. As was seen above, there are cases where one of these will not be present. What this method does is check whether a prescription has one of them missing.

The checkSSN() method corresponds to the rule that states the doctors, patients and pharmacies must be members of the SSB. As such it checks each of these elements' SSB status. If they do not belong they are suspicious of fraud, otherwise they are not.

The checkFalseParticipationPercentage() method matches the rule where each diagnosis has a predefined participation percentage for the SSB in the price of drugs in which the normal case is 70%, the fraudulent is 50% and the dubious one is 60%. In order to verify the percentage, in the implementation, the SSB's participation value is obtained and then this value is divided by the total price of the drug and if the value is 0.5 then a suspicious fraudulent case exists.

The checkDrugsOverPrice() method verifies that a drug's price is the official one. To do so, a rule is verified:

$$\begin{aligned} &originalPrice + minAdd \\ <= &\ fraudulentPrice <= \\ &originalPrice + maxAdd \end{aligned} \tag{2}$$

This rule was created, as is mentioned above, to make possible the creation of the dubious case which consists in adding 2 to the original price. As such, if the price is in between this interval of values (the original value plus a number between 5 and 10) then it is considered a fraudulent case.

The checkCorrespondenceBetweenDrugDiagnosis() method, as the name implies, corresponds to the rule stating that the correspondence of each prescribed drug must be checked against the prescriptions diagnosis. To do this, the Hash Map generated in the Data Generator is used to check every drug and their respective diagnosis. In this method the only verification made is whether the diagnosis in the prescription is identical to the one in the Hash Map. If not, there is a fraudulent case.

These methods comprise the medical and pharmaceutical rules based algorithm. In the next subsection the geo-location based algorithm will be described.

2) Geo-location Based Algorithm: This algorithm is composed of one method: checkDistribution(). The objective is to calculate all the distances and for every 100 records check the distribution. As mentioned above there are three distributions: normal, dubious and fraudulent. This method checks whether the number of patients that go to a pharmacy located 5 Kilometres from the origin point, is lower than the number of patients that go to a pharmacy located 10 Kilometres away. This way the fraudulent case is always detected whereas the dubious case is never identified.

C. Output Analysis

This module is used to collect and present to the user all the detected fraudulent cases. After the Data Analysis phase is completed and all the cases are detected they are presented to the user so that they can be analysed by experts and a more extensive verification can be made.

In this module the output of all the algorithms is collected and presented to the user. There are two sets of outputs and those are the ones from the medical rules based algorithm and the geo-location based algorithm. The medical rules based algorithm are categorized by type of rule, i.e., each medical or pharmaceutical rule has a subset of fraudulent cases. The geo-location algorithm is different since there is only one case scenario which consists in having a fraudulent distribution and as such there is only one subset for the geo-location fraudulent cases.

IV. Evaluation Results

In this chapter the main goal is to test the developed algorithms and check their effectiveness. In order to accomplish this objective, performance measures will be used. They are confusion matrix, accuracy, precision and recall, and K-fold cross-validation. By using all of these metrics it is possible to have a good general idea of how well the system performs since five different measures are applied thus making the evaluation method fairly comprehensive.

The following table is presented with all the variables used in Section III, Table I, where all of them are given values which are discussed below.

TABLE I: Variables presented in the architecture with the respective values.

		Selected Values
Variables Presented in the Architecture	minAdd	5
	maxAdd	10
	assumedPrice	10
	chanceWorkplaceStartPoint	10%
	chanceResidenceStartPoint	20%
	chanceHospitalStartPoint	70%
	normal3KmPerc	60%
	normal5KmPerc	30%
	normal10KmPerc	10%
	fraudulent3KmPerc	50%
	fraudulent5KmPerc	20%
	fraudulent10KmPerc	30%
	dubious3KmPerc	50%
	dubious5KmPerc	25%
	dubious10KmPerc	25%

These values were assumed since no real data was found to support them. In the following paragraphs all the values are discussed and explained.

The minAdd and maxAdd variables are used to represent the range in which the algorithm detects overpriced drugs.

These values were chosen since they represent a slight increase but nonetheless they produce values with a reasonable difference. It was important to use values that would be both plausible and different enough to be considered a fraudulent case in the real world.

The assumedPrice variable is used when a drug has a cost of 0 to the patient, in which case the SSB's contribution is total, i.e., the SSB pays for the total price of the drug. A value of 10 was assumed for this variable since it is close to the centre of the price table in which the values were based on, thus making it a good average value.

The chanceWorkplaceStartPoint, chanceResidenceStartPoint and chanceHospitalStartPoint have the values 10%, 20% and 70% respectively. These values represent the probability of a patient going to a pharmacy from his residence, workplace or a hospital. Since the most common case for patients is to go to the pharmacy right after getting a prescription from a medic, the hospital has a larger percentage than the other locations, followed by the residence and finally the workplace.

The next three variables, normal3KmPerc, normal5KmPerc and normal10KmPerc represent the normal distribution of patients throughout pharmacies located at three different distances. The values are 60%, 30% and 10% respectively. These were chosen since the majority of patients chooses to go to the closest pharmacy available, and so the percentage is higher in the closest pharmacy and lower in the furthest one.

Following the normal distribution comes the fraudulent one with the variables fraudulent3KmPerc, fraudulent5KmPerc and fraudulent10KmPerc. The chosen values were 50%, 20% and 30%. In this distribution the objective was to create a scenario were more patients go to the furthest pharmacy since this is an indication of a fraudulent behaviour. So, the percentage of patients going to the second closest pharmacy is lower than that of the ones going to the furthest one.

At last a dubious distribution was created as to test the performance of the developed algorithm. For this particular case it was necessary to create a distribution that did not fall under any of the two previously described categories. In order to do this the following values were used: 50%, 25% and 25% for the dubious3KmPerc, dubious5KmPerc and dubious10KmPerc respectively.

In the following sections two case studies are presented with the obtained results for the rule based algorithm and the geo-location based algorithm respectively.

The dataset used is composed of approximately 800 thousand prescriptions and roughly 82 million records of geolocation data which is equivalent to about 820 thousand geographical distributions.

A. *Case Study 1*

In this case study the prescriptions are analysed. The metrics described above are calculated and discussed.

The first performance metric is the confusion matrix, that can be seen in Fig. 5.

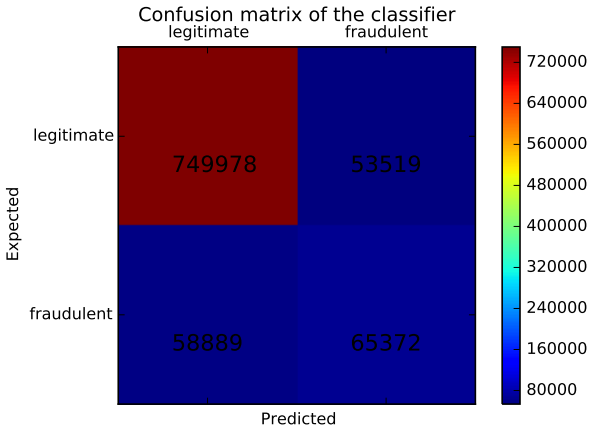

Fig. 5: Confusion matrix of the prescription dataset.

The accuracy using this dataset was of approximately 0.88, or 88% which is a good result. But as seen above, this measure can not be completely trusted.

In terms of precision the obtained result was of approximately 0.55 which is an average result and the recall was roughly 0.53 which is also an average result. As mentioned previously the goal is to maximize the recall since, in the topic of this work it is more important to obtain the maximum number of positive results, even if false positives, because it is less costly to falsely identify a patient as fraudulent than the opposite.

As can be seen the recall is not very high and this is due to the number of dubious cases which were identified as legitimate (negatives) when they should have been identified as positives (which means they were false negatives). This implies that the recall only identified roughly half of all the positive cases, since the other half is composed by the false negatives.

In order to improve the recall, one can reduce the percentage of dubious cases from 2% to 1% or 0.5%.

If 1% of dubious cases are used, a precision of 0.38 and a recall of 0.69 is obtained, meaning that a better recall is obtained. However the precision changed as well. This happens because precision and recall are inversely related, i.e., if one gets a higher result the other will necessarily have a lower result. Using 0.5% of dubious cases increases the recall even more, obtaining a value of 0.23 for the precision and 0.82 for the recall.

As the number of dubious cases diminishes the recall gets higher. Values lower than 0.5% would not be significant so they will not be calculated.

A representation of the above results can be seen in Table II.

Lastly a K Fold cross validation method was applied to the dataset with K = 10. This value of K was chosen since, as can be seen in [23], using a higher number of folders would make the sets overlap, thus not adding new information.

So, using a K = 10 the obtained result is, for the recall, 0.33. The recall was used since, as mentioned before, the primary

TABLE II: Variation of precision and recall with dubious case percentage

		Precision	Recall
Dubious	2%	0.55	0.53
Cases	1%	0.38	0.69
Percentage	0.5%	0.23	0.82

goal is to maximize the recall for the topic of this work. As can be seen this result is not very high, implying that the implemented algorithm would not work very well with new data. However, if the value of K is changed for example to 15 or 20, a different recall is obtained, closer to the one seen above, with the original dataset. So for K = 15 the recall is 0.53, roughly the same as seen above. Using K = 20 a recall of 0.53 is attained. So, as can be seen, K values higher than 10 result in the same recall.

The different values of K and the obtained recall can be seen in Table III.

TABLE III: Recall for different values of K

		Recall
K	10	0.33
	15	0.53
	20	0.53

B. Case Study 2

In this case study the geo-location dataset is analysed. The same method used in the first case study is applied in this one as well.

The first performance metric is the confusion matrix, that is represented in Fig. 6.

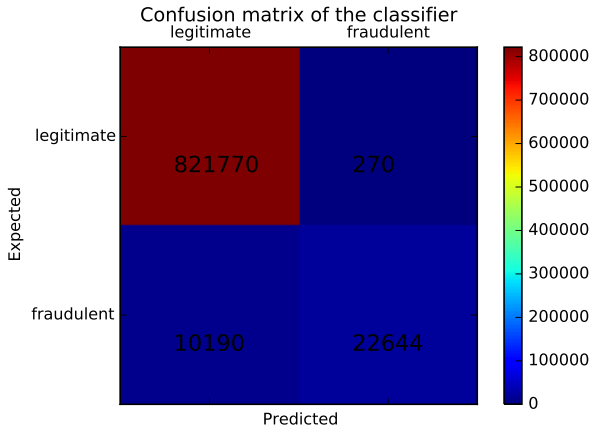

Fig. 6: Confusion matrix of the geo-location dataset.

The accuracy using this dataset was of approximately 0.99, i.e. 99%, which is a very good result. But, once again, this measure is not enough to accurately measure the system's performance.

In terms of precision the obtained result was of approximately 0.99 which is a very high result and the recall was roughly 0.69 which is a good result.

As before the goal is to maximize the recall. Using the same method as before, if the dubious cases are reduced in half, the obtained precision is of approximately 0.98 while the recall increases to roughly 0.82. Using a 0.5% dubious case rate, the precision and recall would be 0.95 and 0.90 respectively. As can be seen using this dataset it is also possible to increase the recall by changing the percentage of dubious cases. The presented values can be seen in a more structured way in Table IV.

TABLE IV: Precision and recall values with distinct dubious case percentages

		Precision	Recall
Dubious	2%	0.99	0.69
Cases	1%	0.98	0.82
Percentage	0.5%	0.95	0.90

The last measure to test is the K Fold cross validation. In this dataset the value K = 10 is also used, the reason being the same as stated in the previous case study.

Using K = 10 the attained recall is 0.69 which is a good value. This value is also roughly the same as the observed recall in the original dataset. As can be seen, this algorithm is more likely to perform very well with new data than the one used in the previous case study.

If different values of K are used, for example 15 and 20, the recall values are both 0.69 which implies that, once again, the algorithm should work very well with new data.

The results can be seen in Table V.

TABLE V: Recall values for variable K

		Recall
K	10	0.69
	15	0.69
	20	0.69

V. Conclusion

It was shown that the healthcare system is subjected to numerous fraudulent actions committed by a variety of entities. The most common types of fraud were presented and discussed in section II. From analysing the literature, two algorithms (geo-location based and medical rule based) were chosen to incorporate the designed solution. An architecture for a possible solution was constructed to meet the goals described in subsection I-A. A novel approach was taken by designing and implementing a data generator capable of creating data to be used in a plethora of algorithms. The detection algorithms were tested with a variety of metrics, including accuracy, precision, recall, and K-Fold cross validation. The obtained results were satisfactory, with the rule based algorithm having a lower score in the K-Fold cross validation and the geo-location based algorithm scoring a good result meaning on average the system performed well.

In conclusion it was seen that there are many algorithms to be used when solving the problem of detecting fraud in the healthcare system, as much as there are different types of fraud. Therefore it was important to use algorithms that detected distinct types of fraud in order to broaden the portion of fraud the system can predict and, consequently, making it more robust.

A. Future Work

In this section some possible improvements to the solution are presented.

- Transform the system into a modular design with each algorithm being a module, as to allow new algorithms to be implemented and added independently, in a simple way, as well as adding the accompanying data generator code for that specific algorithm;
- Implement more algorithms to broaden the detection capabilities of the system;
- Discover more information about how the prescriptions are structured as to create a more authentic dataset;
- Use real data to test and perfect the detection algorithms and to create a more realistic dataset;
- Design and implement a system to present the data in a visual way, plotting graphics to represent the distributions and clustering the prescriptions according to the type of fraud in a graphical way, to allow for a better understanding of the obtained results;

REFERENCES

[1] Y. Yang, M. Manoharan, and K. S. Barber, "Modelling and analysis of identity threat behaviors through text mining of identity theft stories," in *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*. IEEE, 2014, pp. 184–191.

[2] E. H. Humaid and T. Barhoum, "Water consumption financial fraud detection: a model based on rule induction," in *Information and Communication Technology (PICICT), 2013 Palestinian International Conference on*. IEEE, 2013, pp. 115–120.

[3] P. K. Panigrahi, "A framework for discovering internal financial fraud using analytics," in *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*. IEEE, 2011, pp. 323–327.

[4] A. Shen, R. Tong, and Y. Deng, "Application of classification models on credit card fraud detection," in *2007 International Conference on Service Systems and Service Management*. IEEE, 2007, pp. 1–4.

[5] Z. Yongbin, Y. Fucheng, and L. Huaqun, "Behavior-based credit card fraud detecting model," in *2009 Fifth International Joint Conference on INC, IMS and IDC*, 2009.

[6] S. Liu, L. M. Ni, and R. Krishnan, "Fraud detection from taxis' driving behaviors," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 464–472, 2014.

[7] V. Rawte and G. Anuradha, "Fraud detection in health insurance using data mining techniques," in *Communication, Information & Computing Technology (ICCICT), 2015 International Conference on*. IEEE, 2015, pp. 1–5.

[8] M. Ahmed and M. Ahamad, "Combating abuse of health data in the age of ehealth exchange," in *Healthcare Informatics (ICHI), 2014 IEEE International Conference on*. IEEE, 2014, pp. 109–118.

[9] M. Suleiman, R. Agrawal, C. Seay, and W. Grosky, "Data driven implementation to filter fraudulent medicaid applications," in *IEEE SOUTHEASTCON 2014*. IEEE, 2014, pp. 1–8.

[10] U. Srinivasan and B. Arunasalam, "Leveraging big data analytics to reduce healthcare costs," *IT Professional*, vol. 15, no. 6, pp. 21–28, 2013.

[11] J. Li, K.-Y. Huang, J. Jin, and J. Shi, "A survey on statistical methods for health care fraud detection," *Health care management science*, vol. 11, no. 3, pp. 275–287, 2008.

[12] A. Tagaris, G. Konnis, X. Benetou, T. Dimakopoulos, K. Kassis, N. Athanasiadis, S. Rüping, H. Grosskreutz, and D. Koutsouris, "Integrated web services platform for the facilitation of fraud detection in health care e-government services," in *2009 9th International Conference on Information Technology and Applications in Biomedicine*. IEEE, 2009, pp. 1–4.

[13] W.-S. Yang and S.-Y. Hwang, "A process-mining framework for the detection of healthcare fraud and abuse," *Expert Systems with Applications*, vol. 31, no. 1, pp. 56–68, 2006.

[14] N. Christiannini and J. Shawe-Taylor, "Support vector machines and other kernel-based learning methods," 2000.

[15] A. Tagaris, P. Mnimatidis *et al.*, "Implementation of a prescription fraud detection software using rdbms tools and atc coding," in *2009 9th International Conference on Information Technology and Applications in Biomedicine*. IEEE, 2009, pp. 1–4.

[16] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 275–300, 2004.

[17] S. Zhu, Y. Wang, and Y. Wu, "Health care fraud detection using nonnegative matrix factorization," in *Computer Science & Education (ICCSE), 2011 6th International Conference on*. IEEE, 2011, pp. 499–503.

[18] Q. Liu and M. Vasarhelyi, "Healthcare fraud detection: A survey and a clustering model incorporating geo-location information," in *29th world continuous auditing and reporting symposium*, 2013.

[19] "List of the top 200 drugs of 2016," http://www.pharmacy-tech-test.com/top-200-drugs.html, accessed: 2016-08-22.

[20] "Pharmaceutical market january 2016," Internal study conducted by hmR - Health Market Research.

[21] "Survey with information regarding the distance between residence and workplace," http://forum.autohoje.com/off-topic/91243-distancia-de-casa-ao-trabalho-4.html, accessed: 2016-09-21.

[22] M. Kirlidog and C. Asuk, "A fraud detection approach with data mining in health insurance," *Procedia-Social and Behavioral Sciences*, vol. 62, pp. 989–994, 2012.

[23] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation*, vol. 10, no. 7, pp. 1895–1923, 1998.