Design and Implementation of a Fraud Detection Expert System using Ontology-Based Techniques

A dissertation submitted to the University of Manchester
for the degree of Master of Science
in the Faculty of Engineering and Physical Sciences

2013

Giannis Potamitis

School of Computer Science

Table of Contents

Word count: 24,070 words

List of Figures

List of Tables

Abstract

The incidents of fraud are increasing year by year, with credit fraud occupying prominent role. This has driven scientists to keep investigating techniques that can be used to detect credit fraud.

The main contribution of this project involves finding the characteristics of various techniques which have been suggested in the literature for detecting credit fraud. An ontological knowledge base is constructed to conceptualize the findings of research process. To demonstrate the usefulness of this conceptualization an expert system is constructed. This is capable to advise software developers for the detection technique which they should implement in order to detect a specific type of credit fraud. A software developer, who wishes to use the expert system, will be asked a few questions associated with the characteristics of detection techniques. The answers to these questions will help the expert system in deciding the appropriate detection technique which best suits software developer's needs.

The hypothesis throughout the project is that the use of expert system can significantly reduce the amount of research that software developers – who wish to implement a fraud detection tool – need to undertake. This is validated by constructing an online questionnaire and invite software developers to participate in.

An additional contribution is achieved during the project. This involves finding and conceptualizing the characteristics of various different frauds and crimes in a second ontological knowledge base. This could be used – as a future work – to construct systems capable to inform people for the type of fraud or crime which they have been victimized. It is worth noting that the second ontological knowledge base acts as a generic version of the first one. This is because the first ontological knowledge base encapsulates the different types of credit fraud and their detection techniques only; whereas the second ontological knowledge base encapsulates a significant number of different frauds and crimes including credit fraud.

Declaration

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual Property Statement

I. The author of this dissertation (including any appendices and/or schedules to this dissertation) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

II. Copies of this dissertation, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has entered into. This page must form part of any such copies made.

III. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the dissertation, for example graphs and tables ("Reproductions"), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

IV. Further information on the conditions under which disclosure, publication and commercialisation of this dissertation, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/display.aspx?DocID=487), in any relevant Dissertation restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Guidance for the Presentation of Dissertations.

Acknowledgements

I would like to thank my supervisor Dr Sandra Sampaio for her continuous guidance throughout the project. She helped me understand the requirements of the project and indicated to me the right direction of research.

1 Introduction

This first chapter provides an introduction to the area which is related to this project. It starts with fraud in general and moves to credit fraud and detection since these are associated with project's main contribution. Subsequently, a discussion about the exact aims and deliverables of this project is being made.

1.1 Fraud

Gosset el al. (1999) state that the definition of fraud is difficult to be formed since the distinction between fraudulent and legitimate behaviours is not always obvious [1]. On the other hand, Alexopoulos et al. (2007) define fraud as "the deliberate and premeditated act perpetrated to achieve gain on false ground [2]. Sections 2.1 and 2.7 show that the consequences of fraud are not restricted to economic losses but they can also lead to violation of human rights, physical and psychological harms as well as premature deaths [2] [3]. Fraud can be perpetrated everywhere including financial institutions, insurance companies, corporations as well as the government – see chapter 2 for more details.

1.2 Credit Fraud

The main contribution of this project is related to credit fraud. Credit fraud is a term used to refer to the family of frauds which are perpetrated in credit industry. These are discussed in detail in section 2.1. For the purpose of this introductory chapter, a particular attention is taken to credit card fraud which is the most important and dangerous type of credit fraud.

1.2.1 The Use of Credit Card and its Stakeholders

Credit card usage has enormously been increased during the last years. According to [4], 120 million cards were created in Germany in 2004 which led to total credit card purchases of €375 billion at the same year [4]. With respect to the previous year – 2003 – there was an increase of 4% on the overall credit card usage [4].

Delamaire et al. (2009) defined credit card as "a method of selling goods or services without the buyer having cash in hand" [4]. A credit card transaction involves four entities. The first entity is the consumer; that is the person who owns the card and who carries out the legitimate transactions. The second entity is the credit card issuer; that is usually the consumer's bank – also known as issuing bank – which provides the credit services to consumer. The credit card issuer sends the bill to the consumer in order to request a payment for their credit card transactions. The third entity is the merchant who sells goods or services to the consumer by charging consumer's credit card. This charge is achieved through merchant's bank – the forth entity – which sends the request for the transaction to the issuing bank. The issuing bank will check whether the amount of the transaction does not reach the credit card's limit before authorizing that transaction. If the transaction is valid the issuing bank will block the requested amount from consumer's credit card account and send an authorization response to merchant bank. As soon as the authorization response is received by the merchant's bank, the merchant is notified; the transaction is marked as completed and the consumer can take the goods. The blocked amount on consumer's credit card account will be transferred into merchant's bank account in the following days.

1.2.2 Credit Card Fraud

Although the use of credit cards as a payment method can be really convenient for our daily transactions; people must be aware of the risks that they impose themselves while using their credit cards. More precisely, the incremental usage of credit cards gave the opportunity to fraudsters to exploit their vulnerabilities [4]. Credit card fraud refers to any illegal and unauthorized activity on the use of credit cards which is undertaken by a fraudster. According to [5] credit card fraud has been increased between 2005 and 2007. Moreover Bolton et al. (2002) claim that in United Kingdom the total losses of credit card fraud, for 2000, were £286 million [6]. In United States the total losses for 2009 were as high as $3.56 billion; an increase of 10.2% comparing to the previous year [7].

An interesting question arises as to who is responsible to pay for all those losses in case of a credit card fraud. Delamaire et al. (2009) claim that merchants are really vulnerable in case of a credit card fraud because they are required to pay for the losses due to the so-called charge-backs [4]. Charge-backs are requested by the consumer's bank as soon as the consumer reports a transaction as unauthorized. Quah et al. (2008) converges with the above statement by adding that merchants not only have to pay for the amount of the illegitimate transactions but also for any additional charges that are imposed by the credit card issuer [8]. Yet banks are required to pay the costs of investigating whether a transaction, which is reported as illegitimate by the consumer, is indeed illegitimate as well as the costs of having the appropriate equipments for detecting fraudulent transactions [8].

Although consumers are the least vulnerable in case of a credit card fraud there are states which enforce consumers to pay for the losses under particular circumstances. This happens in China in case the consumers do not realize that their credit cards have physically been stolen and fail to report the lost to their banks [8]. According to a discussion of the writer of this report with the manager of one of the Cypriot banks – whose details cannot be published for privacy reasons; the above policy applies to Cyprus as well. In particular the consumers are not forced to pay the losses of an illegitimate credit card transaction if they report the physical lost of card in time or if the card is not physically lost at all. In the first case there shall be no illegitimate transaction at all since the credit card will be locked before the fraudster manages to use it. In the second case where only the details of the credit card are stolen and not the physical card itself; the illegitimate transaction can be undertaken in places where the physical card is not required to be present like phone or internet. With today's technological advances that last type of fraud is very difficult to prevent and therefore the consumer is no longer responsible for any losses that may occur. Therefore those losses burden merchants and issuing banks.

1.3 Credit Fraud Detection

It has already been mentioned that the losses of a credit card fraud can affect all consumers, merchants and issuing banks. Therefore, it is important to establish techniques for detecting and preventing credit card fraud. The literature contains a variety of techniques which can be used to build fraud detection systems. Understanding the characteristics of all those techniques can be a tedious task. A technique which promises a high predictive accuracy may be an appealing candidate to be used in the fraud detection system. However, there are various different parameters that need to be considered before deciding which technique best suits the needs of a particular situation.

For instance, if the abovementioned technique which promises a high predictive accuracy cannot be applied into a large data set and if our data set is indeed large then that technique is obviously not appropriate for our situation.

1.4 Aims of this Project

This project aims on researching the various techniques that have been suggested in the literature for detecting credit fraud. The characteristics of the techniques are analyzed and encapsulated into a knowledge base system. The latter is capable to receive questions – also known as queries – and respond with answers based on the encapsulated information. An example of the question that a user can impose into the knowledge base, using plain English, may be: “What technique should I implement if I want to detect credit card fraud and if there is a lot of noise in the data set?”

As mentioned in the abstract chapter, an additional contribution is decided during the project. This is the construction of a generic fraud ontology as described in 1.4.1 below.

1.4.1 Deliverables

This subsection describes the deliverables of this project.

Credit Fraud Detection Ontology

As mentioned above, the findings of research process are encapsulated into a knowledge base also known as ontology. This is a repository of information which uses formal mathematical notations. The conceptualization of information into a formal mathematical representation allows the ontology to be queried and infer a response. The ontology includes the various techniques and their properties that can be used to detect credit fraud. More details about ontologies can be found in section 2.4.

Expert System

In order to demonstrate the effectiveness of the ontology, an expert system is constructed. The stakeholders of the expert system are the software developers who wish to build a tool capable to detect credit fraud. It asks software developers questions about the current situation like whether the available data sets are noisy or whether they are too big. The answers to these questions are then translated by the expert system into formal statements that can be used to query the ontology. The results returned by the ontology are reported to the users and can be used as an advice of which technique best suits their needs.

It is particularly important to notice that the expert system does not implement the detection techniques; instead it suggests the appropriate technique to be implemented by the users who shall be software developers. The rationale behind this is that the expert system can save software developers' time by reducing the amount of research that they need to undertake. As mentioned in the abstract chapter this forms the project's hypothesis. Without the existence of the expert system software developers would have to exhaustively search the literature in order to discover the best detection technique based on their needs. This is extremely challenging simply because the information in the literature is chaotic and demands good research and analytical capabilities in order to extract useful features from it.

To the best of writer's knowledge there is no any other system which is capable to assist software developers in implementing a tool for detecting credit fraud.

Generic Fraud Ontology

As mentioned in the abstract chapter an additional ontology is constructed during the project. This encapsulates the characteristics of various different frauds and crimes in general. This ontology forms the basis for the construction of a system capable to inform people for the type of fraud or crime which they have been victimized. Unfortunately there was not enough time to construct such a system and therefore only the generic fraud ontology was constructed. Nevertheless, such a system can be constructed as a part of a future work.

As mentioned in the abstract chapter, the relationship between the two ontologies is that the generic fraud ontology can be seen as an upper, generalized version of credit fraud detection ontology. This is because the credit fraud detection ontology encapsulates the different types of credit fraud and their detection techniques only; whereas the generic fraud ontology encapsulates a significant number of different frauds and crimes including credit fraud.

1.5 Conclusion and Report Organization

The first chapter of this report included a brief introduction on the essences of the project. The purpose of this project along with its deliverables have been discussed.

Chapter two details the information found in the literature related to credit fraud and detection. It also discusses all other frauds and crimes that are included in generic fraud ontology. Chapter three details the design process of project's deliverables. Chapter four details the implementation of expert system and the usefulness of generic fraud ontology. Chapter five demonstrates the way in which the expert system has been tested whereas chapter six evaluates the expert system. Finally, an overall project conclusion can be found in chapter seven.

2 Background and Literature Review

This chapter details the information found in the literature. It starts with a reference to the different fraud types of credit industry. Then a discussion on the existing techniques for detecting and preventing credit fraud is made. There is also a brief discussion on the technologies which were used during this project. Finally, other fraud types and crimes are discussed. These are relevant to the generic fraud ontology.

2.1 Credit Fraud

There are three main types of credit fraud in the literature. These are credit card fraud, bankruptcy fraud and credit application fraud [4] [9]. A detailed explanation of each fraud type follows.

2.1.1 Credit Card Fraud

This is the most common fraud type that occurs in credit industry. A fraudster uses a legitimate card to undertake illegitimate transactions. The cardholder is not aware of the fact that their card is being used without their permission. The fraudster takes advantage of cardholder's ignorance by undertaking as much transactions as possible before the cardholder realizes and reports the fraud to their bank [10].

According to Laleh et al. (2009) credit card fraud can be committed either offline or online [9]. These two ways are discussed below.

Offline Credit Card Fraud

Offline fraud occurs when a fraudster steals the physical card and uses it at the actual stores [9]. Although offline fraud is still popular nowadays; it is less common because there is a higher probability to fail. More precisely, the cardholders tend to realize the lost of the physical card and report that to their bank before the fraudster manages to undertake any illegitimate transactions with it. As soon as the stolen card is reported to the bank, the latter will lock the card so as it cannot be used anymore. It is particularly useful to notice that if the cardholder does not realize the lost of their card, a significant financial loss can occur. As mentioned in the introduction chapter, the policies of some banks enforce cardholders to pay for the losses which occur due to an unreported credit card theft. Notice that most of the UK banks tend to send the newly created cards via the post office. This is extremely dangerous because the cards may be stolen while they are on the way to cardholder's destination address [11].

Online Credit Card Fraud

During online fraud only the details of the card are stolen and not the card itself. This is also known as virtual card theft. The details of the card can be used in places where the card need not be physically present like internet or phone purchases [9]. This type of credit card fraud is very dangerous and more difficult to prevent because fraudsters can hold credit card's information for a long period of time before they use it [10]. There is no way for the cardholder to know in advance that their credit card information is stolen. Therefore this type of fraud may only be detected after one or more illegitimate transactions are taken place.

There are various ways that fraudsters adapt in order to steal the information of credit cards. Some of these ways are briefly discussed below.

Skimming

Patidar et al. (2011) define skimming as the "process where the actual data on a card's magnetic stripe is electronically copied onto another" [12]. Fraudsters use special-purpose devices – also known as skimmers – to capture the information of credit cards that are encapsulated inside their magnetic stripes [11] [12]. They can use the stolen card information to create counterfeit physical cards in order to use them at actual shops or simply supply the card information at online shops [11]. Skimming can be committed by an unfaithful employee, who may swipe customer's card using the skimmer device, while the customer is at the point of sale. In the past, skimmer devices have also been introduced on ATM cash machines. In addition to that, micro-cameras have been used to record the PIN code of a cardholder during ATM transactions.

Site Cloning

Fraudsters clone a legitimate website to deceive customers into placing an order with them. Since the fraudulent website seems identical to the legitimate one, the unsuspecting customers provide their credit card information to complete their order. Consequently fraudsters who obtained the customer's credit card information can commit credit card fraud whenever they wish to [12].

False Merchant Sites

According to Patidar et al. (2011) there are various websites that ask for credit card information in order to confirm customer's age [12]. These websites will never charge the credit cards directly but they may sell their information to fraudsters who will commit credit card fraud [12].

Credit Card Generators

These are automated programs which make use of banks' algorithms to generate credit card numbers [12]. Fraudsters can generate an arbitrary sequence of candidate numbers and then use other techniques – like trial and error – to figure out which numbers correspond to real credit card accounts.

Phishing

Refers to the spam emails that are sent by fraudsters in order to deceive their victims and obtain their personal information [12]. Fraudsters can impersonate a service provider or institute that victims collaborate with. In their email, fraudsters can make use of a convincing excuse to ask for victim's personal information including credit card details. The spam emails may also include links to fraudulent websites which again can deceive victims into revealing their personal information. Taking into account the enormous amount of spam emails that we receive at a daily basis, anyone can conclude that this type of fraud is still popular nowadays; although it has been out for many years.

2.1.2 Bankruptcy Fraud

Bankruptcy fraud occurs when consumers use their credit cards to spend more money than they can actually pay [4]. Credit cards can be seen as a way for consumers to borrow money from their banks. Normally consumers will use their credit cards to carry out daily transactions. At a regular basis – for instance once every month – the bank will send a bill to their customers in order to request a payment for their credit card transactions. Customers, who plan to commit bankruptcy fraud, will overdraft their credit card accounts and then declare themselves as being in a position of a personal bankruptcy [4]. In such a case the bank will have to pay for all the losses [4].

Xiong et al. (2013) state that bankruptcy fraud increases expeditiously and can cause serious losses to issuing banks [13]. In addition to that, they suggest the evaluation of credit card applications in order to verify the creditworthiness of applicants [13]. Such an evaluation can usually reveal the possibility of a customer to go bankrupt in the future. Xiong et al. (2013) also state that the abovementioned evaluation is not enough because customers with initial good creditworthiness can still be proved insolvent at a later stage [13]. Therefore even if an applicant, who satisfies the desirable levels of creditworthiness, is provided with a credit card account, the latter should keep being inspected by the bank in order to predict any possibility of future insolvency. More details about the techniques for predicting bankruptcy fraud can be found in section 2.2 and 2.3.

Whittaker et al. (2005) claim that a missed payment on a credit card bill is an indication of an insolvent customer [14]. Banks should take immediate measures to reduce the potential losses in case of a customer's bankruptcy. An example of those measures could be the reduction of allowed credit card limit. Of course, banks need to be very careful when taking restricting measures against their customers. The reason of this is that there is a danger to lose customers who did not intend to commit bankruptcy fraud but for some reason they were unable to pay their bills on time [14].

Credit Bureau

A way of evaluating the creditworthiness of a credit card applicant is by considering the reports of a credit bureau. Credit bureaux are organizations which gather information about consumers from various different sources like financial institutions, banks, credit unions, courts and bankruptcy filings [4]. Banks can request a report from a credit bureau by providing the details of a credit card applicant. Delamaire et al. (2009) state that a credit report can contain "personal particulars, details of non-compliance with contractual obligations, information from public directories and additional positive information such as repayment of loans according to contract at or before maturity" [4]. Information like current home address and occupation details may also be included in the credit report [4].

2.1.3 Credit Application Fraud

Credit application fraud occurs when a fraudster applies for a credit card using false information [4]. The credit application fraud is associated with another serious fraud, the identity fraud.

Identity Fraud

Identity fraud occurs when a fraudster uses a false identity with intension to commit another fraud [15] [16] [17]. Identity fraud can be perpetrated by inventing an identity which does not belong to a real person or by stealing the actual identity of a real person – also known as identity theft [4] [15] [16] [17] [18]. Inventing an identity is easy because there is no need for fraudsters to look for valid information of a real person [18]. Nevertheless, this type of identity fraud is very difficult to succeed nowadays because financial institutions tend to check whether the applicant's information corresponds to a physical person or not.

Identity theft, on the other hand, has a higher possibility to succeed; although it requires more effort to be committed due to the collection of victim's personal information [18]. Fraudsters gather all the necessary information to impersonate their victims. They can then apply for a credit card using victim's information or commit other frauds. If the fraudster applies for a credit card and the fraudulent application succeeds then the fraudster will be able to use the issued credit card to carry out transactions on behalf of the victim.

Bose (2006) states that identify theft grows rapidly year by year and that there were 9.9 million victims in America on 2005 [19]. There are several ways that fraudsters adopt to steal the personal information of their victims. They can burgle victim's houses, steal their garbage or mails, bribe employees who have access to identity information or use malicious software like spywares to obtain unauthorized access to victim's computers and gather their confidential information [15] [16] [17] [19].

The consequences of identity fraud in credit application can vary. If the fraudster invented an identity which did not belong to a real person and managed to receive a credit card, then the issuing bank would definitely lose their money because the fraudster would overdraft their credit card account and vanish without paying the bill [17]. On the other hand, if the fraudster used a real identity then the real person would be liable to pay the bill unless he or she manages to proof the identity theft. In addition to that, the creditworthiness of real person might be damaged, making them unable to receive credit cards or loans in the future [17]. It is worth mentioning that fraudsters who commit identity thefts can easily take over the bank accounts of real persons and use them to their advantage [17].

Chain of Trust

Abdelhalim et al. (2009) provide a broader definition of application fraud. They explain that “application fraud occurs when an individual or an organization applies for an identity certificate using someone else’s identity” [20]. By identity certificate they mean any formal document which can proof the identity of a person like passport, credit card, driving license etc [20]. They claim that application fraud is based on the way that identity certificates are used in the real world. More precisely they explain that there is a chain of trust between identity certificates which can easily be exploited by fraudsters [20]. According to them “the issuing of a credit card relies on the social security card, which in its turn relies on the passport, which again relies on the birth certificate” [20]. In other words if a fraudster manages to steal the birth certificate of a victim, he will be able to apply for a new passport following by a new social security card and finally by a new credit card [20].

2.2 Data Mining and Detection Techniques

This section describes the concept of data mining and the techniques which are found in the literature for detecting credit fraud. The main reason why these techniques are reviewed is that they form the basis of the credit fraud detection ontology and they are reported as an implementation advice by the expert system.

2.2.1 Data Mining

Data mining refers to a family of machine learning techniques capable to analyze and extract non-trivial patterns from data [21]. Data mining is also known as knowledge discovery because it can reveal previously unknown information which was hidden in the data of various databases [21]. The mined information can be proved very useful for the organizations who apply data mining. Based on the results, organizations may make important decisions which can help them survive in the competitive environment. For instance an organization can analyze the sale records of its customers in order to send attractive offers on the most popular products [22].

Hormozi et al. (2004) state that “data mining enables an organization to focus on the most important information in the database, which allows managers to make more knowledge decisions by predicting future trends and behaviours” [23]. Given that databases are too large; it is very inconvenient and impractical to look manually for hidden patterns on the data [23]. Therefore data mining can be introduced to facilitate the discovery of useful knowledge. Forrester Research firm reported that 52%, of 1000 companies in total, decided to employ data mining techniques in 2001 to improve their marketing strategies; an increase of 34% comparing to 1999 [23].

Data mining can also be used to detect fraudulent credit card transactions, predict which customers are more likely to default their contractual obligations by going bankrupt as well as identify fraudulent credit applications. Srivastava et al. (2008) state that the only way to detect credit card fraud is by analyzing the spending behaviour of customers using data mining techniques [24]. Customers tend to follow a standard spending profile and therefore any transaction which deviates from that standard can be considered as suspicious [24]. Suspicious transactions can be examined in detailed by bank officers to determine whether they are indeed fraudulent or not.

Like most of the machine learning algorithms, data mining techniques tend to learn models from data. There are three approaches on learning the data mining models. Those are supervised learning, unsupervised learning and semi-supervised learning; and they are described below.

Supervised Learning

This is the most common learning approach where the model is trained using pre-defined class labels [6]. In the context of credit card fraud detection the class labels may be the “legitimate” or “fraudulent” transactions. A supervisor provides a training data set whose transactions are classified in advanced as belonging to the “legitimate” or the “fraudulent” class. The training set can be used to build the predicting model. Any new transaction can be compared against the model to predict its class. If the new transaction follows a similar pattern to the illegitimate behaviour – as this is described by the trained model – it will be classified as a fraudulent transaction.

One limitation of supervised learning is that it requires confidentiality on the class of each training sample. If there is a fraudulent transaction X which is misclassified by the supervisor as legitimate then the constructed model will be problematic. The same happens for a legitimate transaction which is misclassified as a fraudulent [6]. Moreover an imbalanced distribution – also known as skewed distribution – of the class labels in the training set can result in a model which does not have a very good predictive accuracy. Skewed distribution is the situation where there are much fewer training samples of class *A* than class *B*. Maes et al. (2002) state that fraudulent transactions are usually much fewer than legitimate ones [25]. Therefore the problem of skewed distribution exists in the area of credit card fraud detection and supervised learning is seriously affected from that. In addition to that, supervised learning models cannot detect new frauds [6]. This is because the behaviour of the new fraud is unknown to the trained model and therefore the latter cannot detect it. Further training is needed on the model to learn the existence of the new frauds. Finally a substantial effort is required from experienced people – also known as supervisors – to derive the labelled training samples which will be used to construct the model [26].

Unsupervised Learning

Unsupervised learning involves no class labels for model construction. Bolton et al. (2002) explain that unsupervised learning techniques aim to discover those instances “whose behaviour is unusual” [6]. A model which represents the “baseline distribution of normal behaviour” is constructed without using class labels [6]. That model is then used to detect instances which deviate from that normal behaviour [6]. It is particularly useful to notice that unsupervised learning techniques can detect both old and new fraud types since they are not bounded to the fraud patterns which are encapsulated in the labelled training samples like supervised learning techniques do. Instead unsupervised learning techniques aim to detect anything which does not comply with the normal behaviour.

Semi-supervised Learning

As mentioned above, supervised learning requires all the training samples to have their class labelled. In contrast unsupervised learning needs no labelled samples at all. Semi-supervised learning lies between supervised and unsupervised learning since it involves a small number of labelled samples and a large number of unlabelled samples [26]. In the context of credit card fraud detection, semi-supervised learning techniques may involve labels for some of the legitimate transactions only. This can reduce the effort needed by supervisors to classify training data [26].

2.2.2 Detection Techniques

This subsection provides a brief discussion on various data mining techniques which can be used to detect credit fraud. It is particularly useful to notice that the algorithmic details of these techniques are out of the scope of this report. What is more important for this project is to understand the techniques at a higher level of abstraction and extract useful characteristics which can be used to build the knowledge base. The extracted characteristics can be found in chapter 3. Here there is a general discussion over the detection techniques.

Artificial Neural Networks (ANNs)

An artificial neural network imitates the way that human brain works [27]. It consists of a number of nodes – which are called neurons – and edges which interconnect those neurons [28]. Neurons are computational units which process some input information and produce some output [28]. The output of one neuron is passed as input to another. A neuron of human brain is activated if and only if the received signal is sufficiently strong [28]. Likewise an artificial neuron receives not only some input signals but also a weight which determines whether the input signals are sufficiently strong or not. If the signals are strong enough, an activation function will start executing to produce the output [28]. Figure 1 which is taken from [28] illustrates the structure of a single artificial neuron. It shows the inputs, weights, activation function and output.

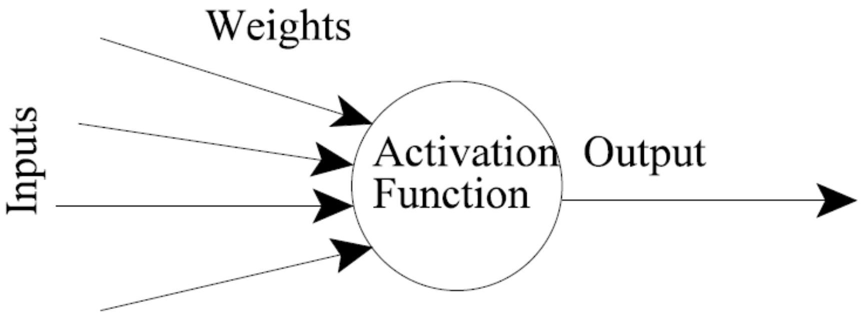


Figure 1: Artificial Neuron Structure **[28]**

More details about artificial neural networks can be found in [28].

Support Vector Machines (SVMs)

Support Vector Machines is a binary classification methodology. This means that an input sample can be classified into one out of two possible classes. It is suitable for credit card fraud detection because only two classes are needed; namely the “legitimate” and “fraudulent” class. SVM tries to calculate an optimal hyperplane which will separate the samples of the two classes [29]. There are various hyperplanes which can do that job but an optimal hyperplane will also maximize the margins between the samples of the two classes [30]. Figure 2 which is taken from [30] illustrates an example of two classes which are separated by an optimal hyperplane. Blue and black bullets correspond to the samples of the two distinct classes. Support vectors define the boundaries of each class by taking into account the sample which is closest to the hyperplane [30]. Clearly the separating hyperplane lies in the middle of support vectors by maximizing the margin between them [30]. A new sample is classified by measuring its distance from the hyperplane.

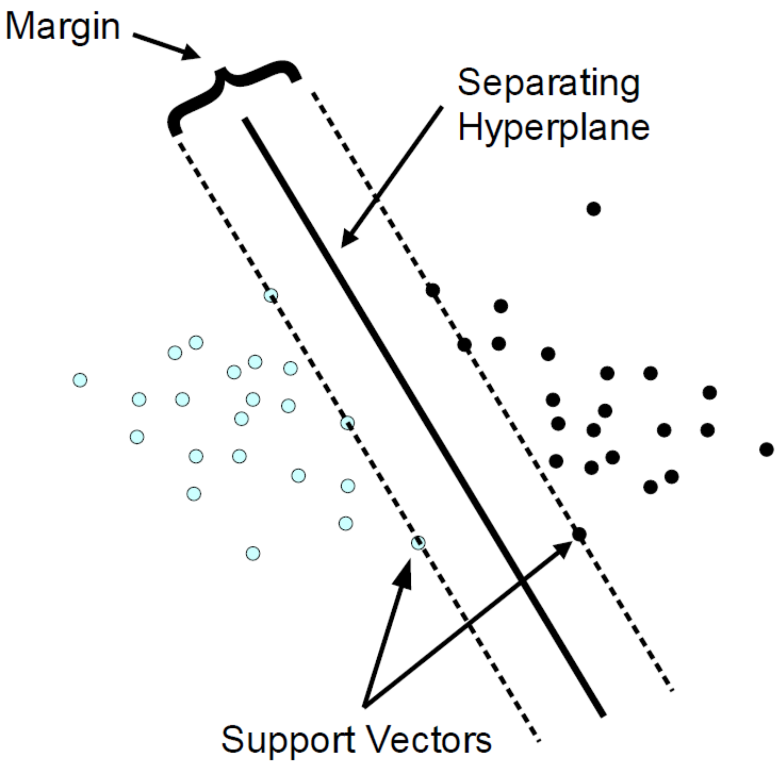


Figure 2: SVM Optimal Hyperplane **[30]**

According to Wu et al. (2007), SVM “has a sound theoretical foundation” which makes it a robust classification technique [31]. Nevertheless, an optimal hyperplane which separates linearly the samples of the two classes is not always possible to be found [32]. In that situation a kernel function can be used to map the non-linearly separable data into a higher dimension in which an optimal hyperplane can be found [32]. The main issue with kernel functions is that they increase the implementation complexity of SVMs.

More details about SVMs can be found in [29].

Bayesian Belief Networks (BBNs)

A Bayesian belief network is a probabilistic classifier which is based on directed acyclic graph (DAG) [27]. The nodes of the graph represent domain variables [33]. Those are actually the attribute values that exist on the dataset [33]. Probabilistic dependencies between the nodes are represented by the edges that connect those nodes [33]. Two nodes are said to be conditionally independent if and only if there are no any edges to interconnect them [27]. BBNs make use of the conditional probability theory. In the context of credit card fraud detection a BBN will represent the probability of a fraudulent transaction given that the variables of that transaction have some specific values.

According to Cheng et al. (2001) the main advantage of BBNs is that they can easily be interpreted by humans who can modify them in case they need to achieve a better predictive accuracy [33].

More details about BBNs can be found in [33].

Decision Trees (DTs)

A decision tree is a supervised learning data mining technique which repeatedly partitions the training samples into more identical groups based on a dissimilarity measure [32]. There are many different algorithms which can be used to split the training samples into branch-like groups [34]. The resulting model has a tree-like structure consisting of a root, a number of branches, nodes and leaves. Figure 3 which is taken from [35], illustrates an example of a decision tree for playing tennis. The root is the "Outlook" node while "Humidity" and "Wind" are the subsequent nodes of the tree. The branches are "Sunny", "Overcast", "Rain", "High", "Normal", "Strong" and "Weak". The leaves which are found in the bottom of the tree are "Yes" and "No". For each leaf *L*, there is a unique path from tree's root to *L* indicating the decision rule for classifying a new sample as belonging to the class of *L* [34]. An example of a decision rule is: "IF Outlook = 'Sunny' AND Humidity = 'Normal' then class = 'Yes'".

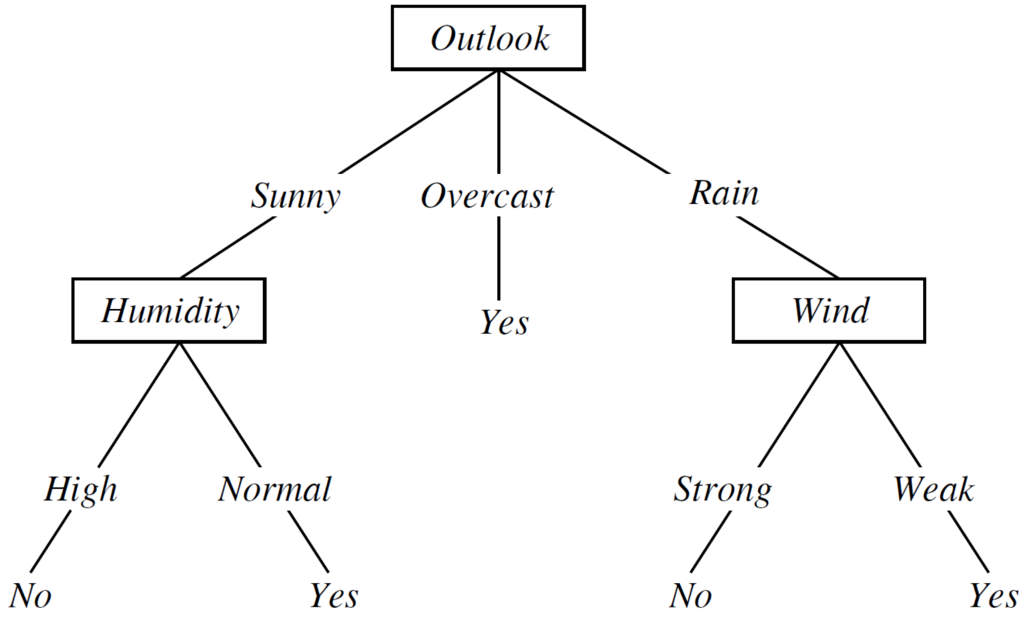


Figure 3: An Example of Decision Trees **[35]**

One important advantage of decision trees over other data mining techniques is the ease of interpretability by humans. Taking into account that a decision tree can be represented either as a tree-like structure or as a set of "IF..THEN" decision rules; one can conclude that it can be easily understood in either form [32].

On the other hand decision trees are known to be unstable. According to [36], a "small fluctuations in the data sample may result in large variations in the classifications assigned to the instances" [36]. Moreover the resulting decision tree may contain some errors or anomalies and pruning may be needed to resolve those issues [37]. Pruning is the process of removing erroneous branches, nodes or leaves. This is a tricky process because it may result in degradation of classification performance of the tree [37].

More details about DTs can be found in [34].

Outlier Detection (OD)

Hawkins et al. (2002) define outlier as "an observation that deviates so much from other observations as to arouse suspicion that is was generated by a different mechanism" [38]. Outlier detection refers to a family of unsupervised learning data mining techniques capable to discover rare patterns in data – also known as outliers [27]. Outliers can be detected without knowing the data set's distribution or needing any labelled training samples [39].

Figure 4 which is taken from [40], illustrates a graph which has an outlier – that is the pullet which deviates a lot from the rest of pullets.

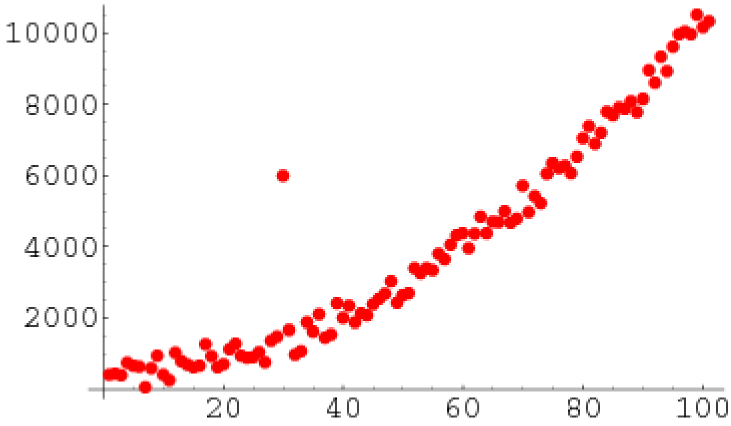


Figure 4: Example of an Outlier **[40]**

The complexity of outlier detection lies in the fact that a similarity metric needs to carefully be chosen [5]. This will calculate the similarity between different data [5]. Different metrics often lead to different outcomes and therefore choosing the right metric can proved a complex task [5].

In the context of credit card fraud detection, a fraudulent transaction can be seen as an outlier which behaves differently comparing to legitimate transactions; hence it can easily be spotted.

More details about outlier detection can be found in [41].

Peer Group Analysis (PGA)

Peer group analysis is an unsupervised learning technique which monitors “behaviour over time” [42]. In the context of credit card fraud detection, PGA identifies all those accounts *A* that used to behave similarly to a target account *c* at some time t_{past} in the past [5]. The accounts *A* are known as the “peer group” of *c.* The target account *c* is marked as suspicious if, and only if, at current time $t_{current}$ it demonstrates a different behaviour than this of its peer group *A* [6]. The rationale behind this approach is that a sudden change in the spending behaviour of a customer at some specific time in the year will not be marked as suspicious if a similar change occurs to its peer group at the same time [5]. The spending behaviour of most customers changes during special circumstances like Christmas and Easter periods; causing most fraud detection systems to produce false alarms [5]. PGA eliminates those false alarms by reporting only those accounts that are indeed suspicious comparing to their peer groups.

More details about PGA can be found in [42].

Hidden Markov Model (HMM)

Srivastava et al. (2008) defined HMM as “a double embedded stochastic process with two hierarchy level” [24]. Comparing to a conventional Markov model; HMM is much more expressive and can represent more complex stochastic processes [24]. HMMs are widely used in the area of speech recognition, computer vision and pattern recognition [43]. HMMs have also been applied in the area of credit card fraud detection by Srivastava et al. (2008) [24].

Figure 5 which is taken from [44] illustrates an example of a traditional Markov model. Markov models consist of states, observations and probabilistic transitions. There are three states in figure 5; those are "Bull", "Bear" and "Even". There are also three observations; "up", "down" and "unchanged". Each state emits a specific observation; for instance "Bull" emits "up". The transitions indicate the probability of switching between states. For example the probability of switching to state "Bear" is 0.2 given that the current state is "Bull". If a sequence of observations like "unchanged-down-up-down" is emitted, one can easily conclude that they have been produced by states "Even-Bear-Bull-Bear" [44].

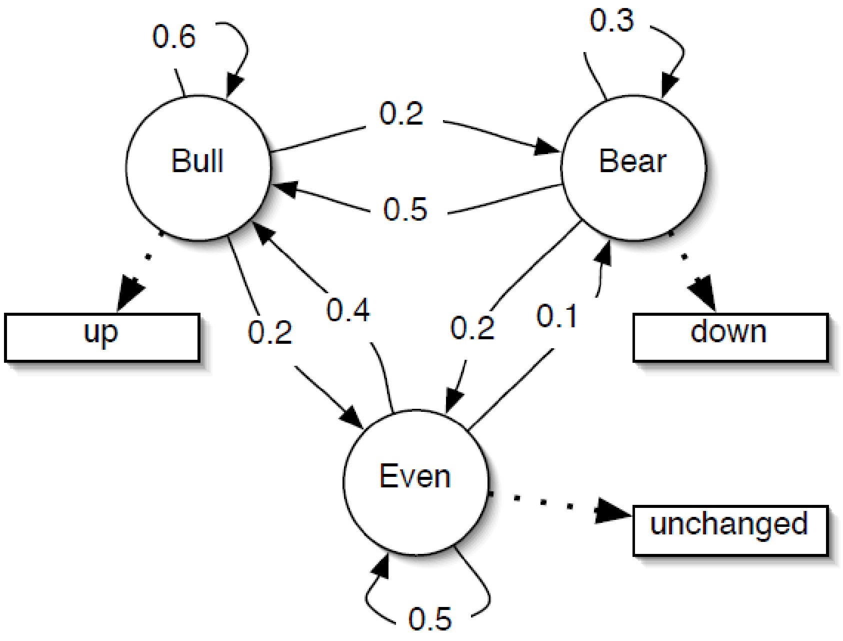


Figure 5: Traditional Markov Model Example **[44]**

On the other hand a hidden Markov model allows more than one observations to be emitted by each state [44]. This is done by declaring different probabilities for each observation of each state [44]. Figure 6 which is taken from [44] illustrates such an HMM. For instance state "Bull" can now emit "up", "down" and "unchanged" with a probability of 0.7, 0.1 and 0.2 respectively. There is a higher possibility for state "Bull" to emit an "up" but it can also emit other observations too [44]. HMMs are more expressive and can encapsulate more meaning than traditional Markov models [44].

Since an external observer can only see the sequence of emitted observations; it is not possible to know exactly the state sequence which produced those observations because states are hidden [24] [44]. Nevertheless, one can still calculate the probability that an emitted observation sequence has been generated by a possible state sequence [44].

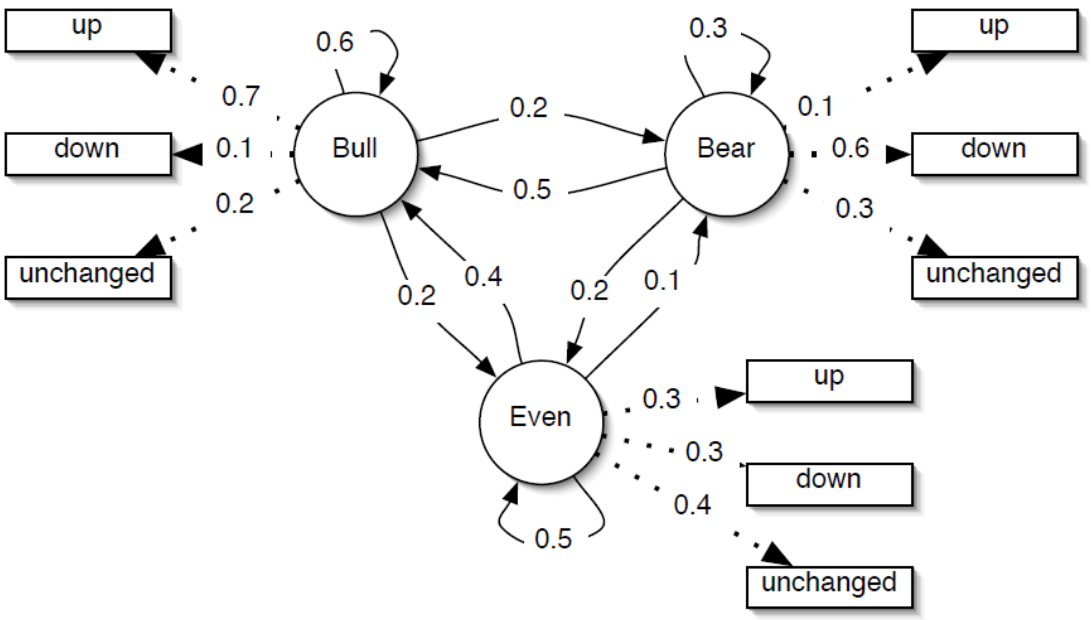


Figure 6: Hidden Markov Model Example **[44]**

More details about HMMs can be found in [44].

Artificial Immune System (AIS)

Artificial Immune Systems belong to the family of artificial intelligence which imitate the way that human's immune system works [45]. The main function of our immune system is to categorize all the cells found in the body as "self" or "non-self" [45] [46]. Non-self cells are then examined exhaustively in order to decide on a suitable defence [46]. The defensive mechanism which was used to protect the body from a non-self cell *x* is recorded for future reference. If a non-self cell similar to *x* invades the body at some time in the future, the same defensive mechanism will be used. Therefore the immune system is able to protect the body from non-self cells by applying an evolutionary learning mechanism [46]. It is particularly useful to notice that our immune system can also spot new types of non-self cells which are unknown and not seen before [45].

An AIS works in a similar way to our immune system. In the context of credit card fraud detection, an AIS can spot fraudulent transactions since they can be thought as the non-self cells [45]. This is done without the need for exhaustive labelled training. Instead there is a small number of labelled samples that correspond to legitimate transactions and a large number of unlabelled samples that correspond to either legitimate or fraudulent transactions. This makes AIS a semi-supervised learning technique which can spot both previously seen and new unseen fraudulent patterns.

More details about AIS can be found in [46].

Nearest Neighbour (kNN)

The Nearest neighbour classification is one of the simplest supervised learning techniques since it does not involve any model construction [47]. Instead labelled training samples are stored in a repository and they are retrieved whenever a new unlabelled sample needs to be classified [47]. The distance of the new unlabelled sample *s* to all the labelled training samples is calculated by using a suitable metric [47]. The rationale behind this is to identify the *k* labelled samples – where *k* is a predefined positive integer – that are as close to *s* as possible [47]. The predominant class label of these *k* samples is then inherited by *s* [31] [47].

2.2.3 Challenges

Implementing a fraud detection tool using data mining techniques involves a number of challenges which needs to carefully be considered.

Skewed Distribution

As already mentioned in 2.2.1; when labelled training samples are needed to construct the detection model there is a possibility of the unbalanced class distribution problem, also known as skewed distribution. In other words if we want to build a classifier which can categorize new samples as belonging to either class *A* or class *B* then we need to provide training samples with labels on these two classes. Skewed distribution occurs when there are much more labelled samples of class *A* than class *B*. This will cause the model to know very little about class *B* and much more about class *A* affecting in this way its predicting accuracy [25]. This is a very common problem in credit card fraud detection because the number of fraudulent transactions is usually much smaller than this of the legitimate transactions [25]. Therefore software developers need to find ways to deal with this problem when building fraud detection systems.

Noise

According to Maes et al. (2002) “noise is simply the presence of errors in the data, for example incorrect dates” [25]. Missing values are also considered as noise [48]. Noise can result in an erroneous model construction with bad predictive accuracy [25]. The process of removing noise is called data cleansing [25]. Depending on the concerned data set; data cleansing can be a very complex task [25].

Supplying Labelled Training Samples

Finding training samples and providing the right class labels for model construction can also be a very complex task. This is one of the biggest challenges of supervised learning techniques since labelled training samples may not always be available [26].

Overlapping Data

Overlapping occurs when a fraudulent transaction looks very similar to a legitimate one or when a legitimate transaction looks very similar to a fraudulent one [25]. This is also a problem because it can lead to an erroneous model construction.

Choosing Parameters

Most of data mining techniques require a number of parameters including thresholds to pre-set by the user. Different parameters can lead to completely different model performance [49]. This increases the complexity of model construction.

Feature Selection

Selecting the features – also known as attributes or columns – of the data set that should be used to construct the detection model can also be a challenge.

Many articles in the literature suggest the features that should be used to achieve better results.

Over-fitting

Generally the training data set always contains few errors or random values even after data cleansing. These are known as “small fluctuations” in the data [32]. Over-fitting occurs when the algorithm used in model construction, tries to learn as many information as possible from the training data set including this small fluctuations which do not represent the real situation [32]. This can lead to a very complex model with poor predictive accuracy.

2.3 Related Work

This section provides a brief reference to the literature articles which suggest the techniques of subsection 2.2.2 for detecting credit fraud. It is split into further subsections based on the different types of credit fraud. The implementations described in these articles are encapsulated in the credit fraud detection ontology and suggested by the expert system.

2.3.1 Related Work for Credit Card Fraud

The literature work which is related to credit card fraud is described below. This is done by categorizing the work based on the detection techniques.

Using ANN

Wiese et al. (2009) suggest an implementation of ANNs for detecting credit card fraud [50]. Their implementation takes into account a sequence of transactions that have occurred at some time in the past, in order to determine whether a new transaction is legitimate or fraudulent [50]. They believe that “looking at individual transactions” only is misleading since it cannot face any periodical changes in spending behaviour of a customer [50]. They call their approach as “Long Short-term Memory Recurrent Neural Network (LSTM)” [50].

Guo et al. (2008) suggest a different implementation of ANNs by converting the training samples into confidence values using a specific mathematical formula and then supply these values to train the ANN – instead of the original training samples [51]. They call their approach as “confidence-based neural network” and they claim that it can achieve promising results in detecting credit card fraud [51].

Another implementation of ANNs is suggested by Patidar et al. (2011) [12]. They use the genetic algorithm – the details of which can be found in [52] – in order to derive the optimal parameters of ANN [12]. Like many other data mining techniques, ANNs make use of a number of parameters which need to be specified by software developers. Although the values of theses parameters can seriously affect the predicting accuracy of ANN models; a standard practice for specifying these parameters has never been established [12]. The use of genetic algorithm which is suggested by Patidar et al. (2011) [12] can help in deciding these optimal parameters. They call their approach as "Genetic Algorithm Neural Network (GANN)" [12].

Using SVM

Chen et al. (2006) suggest an implementation of SVM which they call "Binary Support Vector System (BSVS)" [53]. One of the main problems of data mining techniques arises in situations where the training samples have an imbalanced distribution – also known as skewed distribution. In such a case the misclassification rate is increased whereas the predicting accuracy of the classifier is reduced. The approach of Chen et al. (2006) is insensitive to skewed distribution of training samples [53].

An innovative implementation of SVMs for detecting credit card fraud is also suggested by Chen et al. (2004) [54]. They suggest from the issuing banks to ask their new customers to fill some questionnaires that can help them understand the spending habits of the customers [54]. This is particularly useful since there is no any prior history on the spending behaviour of new customers and therefore the detection techniques cannot spot fraudulent transactions at the initial stage [54]. Therefore the answers to the questionnaires can be used in a similar manner to the historical information of each customer. They call their approach as "Questionnaire-Responded Transaction Model" (QRT Model) [54].

Using BBN

Maes et al. (2002) suggest an implementation of BBNs for detecting credit card fraud [25]. They claim that their approach can detect up to 8% more fraudulent transactions than ANNs can do [25]. To the best of writer's knowledge, this is the only article in literature which suggests the use of BBNs in credit card fraud.

Using DT

Sahin et al. (2011) provide three different implementations of decision trees for detecting credit card fraud [37]. These implementations are called C5.0, C&RT and CHAID [37]. Their differences lie in the way in which they construct the tree as well as the pruning algorithm which they use to remove erroneous branches and nodes [55]. According to the experiments made by Sahin et al. (2011), the best predicting accuracy was achieved by C5.0 with an average of 92.80%, following by CHAID with 92.22% and finally by C&RT with 91.34% [37]. In their experiments, the three DT implementations outperformed the SVM implementation which achieved an average accuracy of 88.38% [37].

Using Outlier Detection

YU et al. (2009) suggest an implementation of outlier detection technique [39]. The similarity metric that they use to detect outliers is called distance sum. This is mathematically explained in [39].

Yamanishi et al. (2004) suggest another implementation of outlier detection for detecting credit card fraud [56]. They call their approach as "SmartSifter" and claim that it can be applied in real time [56]. This means that a new transaction is checked as soon as it arrives before being authorized [56]. This is not the case for most fraud detection systems because real time detection is time consuming [56]. Most of them will check the newly authorized transactions at some time in the future – for example once a day – in a batch processing mode [56]. The main disadvantage of this approach is that a fraud is just detected but not prevented. If, for instance, a fraud was committed in a physical shop then the fraudster would take the products and run away before the bank discover this fraud. Therefore somebody – either the legitimate cardholder or merchant or bank – would need to pay the losses of this fraud.

Using PGA

Weston et al. (2008) suggest an implementation of PGA for detecting credit card fraud [42]. Their approach cannot detect fraud in real time but instead once every night [42]. To the best of writer’s knowledge there is no other implementation of PGA in the literature for this purpose.

Using HMM

Srivastava et al. (2010) suggest an implementation of HMM which promises a good predictive accuracy and a minimal misclassification error [24]. However, their approach does not perform well on new customers where historical information is not available [24]. Again there is no other implementation of HMM for credit card fraud to the best of writer’s knowledge.

Using AIS

Brabazon et al. (2010) propose an implementation of AIS for detecting credit card fraud which is committed online only [45]. Although their approach can identify 90% of legitimate transactions; 96% of fraudulent transactions are classified as legitimate and therefore their approach is at least unrealistic [45].

Another proposal of AIS for credit card fraud has been made by Gadi et al. (2008) [49]. They use the genetic algorithm [52] as well to derive the optimal parameters of their model [49].

2.3.2 Related Work for Bankruptcy Fraud

The literature work which is related to bankruptcy fraud is described below. Again, this is done by categorizing the work based on the detection techniques.

Using ANN

Wilson et al. (1994) propose an ANN implementation for predicting firms that are most likely to go bankrupt at some time in the future [57]. Their experiments reveal promising results.

Moreover, Pendharkar (2005) suggests another ANN implementation for predicting bankruptcy fraud [58]. One of the various parameters that ANN models need is a threshold value *t* [58]. This is used to determine the class of a new sample and is usually given a fixed value of 0.5 [58]. The approach described in [58] uses a variable threshold value which is calculated based on the characteristics of training samples. This improves the predictive accuracy of the model [58]. Pendharkar (2005) calls this approach as “threshold-varying artificial neural network (TV-ANN)”.

Using SVM

Min et al. (2005) suggest an SVM implementation for detecting bankruptcy fraud [59]. According to their experiments they achieved an overall predictive accuracy of 83% [59].

Wu et al. (2007) propose another SVM implementation [60]. They use the genetic algorithm [52] for parameter optimization [60]. They call their approach as “Genetic Algorithm SVM (GA-SVM)” and show that it can achieve a predictive accuracy of 97% with a minimal misclassification error [60].

Moreover, Xiong et al. (2013) suggest an SVM implementation which takes into account the score of credit bureaux and the payment history of customers to predict their bankruptcy likelihood in the future [13].

Using kNN

Chen et al. (2011) demonstrate the use of kNN for corporate bankruptcy prediction [61]. They use an algorithm called Particle Swarm Optimization (POS) – the details of which can be found in [62] – in order to calculate an optimal value for *k* and to get help on feature selection [61]. Recall that feature selection is the process of deciding which features – also known as attributes – are most appropriate to be used in the data mining process. They call their approach as “Adaptive Fuzzy K-Nearest Neighbour” [61].

Using DT

Bastos (2007) proposes a bankruptcy prediction technique using boosted decision trees [36]. “Boosting is a procedure that aggregates many ‘weak’ classifiers in order to achieve a high classification performance” [36]. A number of decision trees are used individually to produce a bankruptcy prediction for a given customer [36]. The “weighted majority vote” is then introduced to combine all those predictions together in order to derive the final outcome [36]. This boosting technique can overcome the instability problem of decision trees which is mentioned in 2.2.2.

Li et al. (2010) suggest the use of C&RT decision tree algorithm – which has been proposed by Leo et al. (1984) [63] – in order to predict corporate bankruptcy [55]. Their experimental results are very promising since they achieved a predictive accuracy of 90% [55].

2.3.3 Related Work for Credit Application Fraud

The literature work which is related to credit application fraud is described below. To the best of writer's knowledge there is no too much related work on credit application fraud in the literature.

Abdelhalim et al. (2009) suggest an innovative approach for detecting credit application fraud using the web as an information source [64]. More precisely they compare the information of credit applications with identity information which is extracted from the web [64]. The rationale behind this is to detect any inconsistencies which indicate that a credit application may be fraudulent [64]. For instance if an applicant with social security number *s* claims that he was born on 1965 but it can be found from the web that the same applicant with *s* may actually be born on 1970 then there is a serious identity inconsistency here which suggests the possibility of identity fraud. Abdelhalim et al. (2009) claim that the web, which is so powerful, allows the extraction of useful identity information for a given person if it is used correctly [64]. They supply the extracted information in a decision tree which is created on the fly in order to conclude whether a credit application is indeed fraudulent or not [64].

Moreover, Phua et al. (2009) propose a detection technique for credit application fraud called "Communal Analysis Suspicion Scoring" (CASS) [18]. This can detect fraudulent applications by generating suspicion scores [18].

2.4 Ontology

This section provides a brief discussion on ontologies. As already mentioned, two ontologies were constructed during this project. The first one conceptualizes the various types of credit fraud along with their detection techniques. The second one conceptualizes the characteristics of various frauds and crimes in general.

An ontology is widely defined as “a specification of a conceptualization” [65]. Conceptualization refers to the “abstract, simplified view of the world” [65]. A specific real-world domain can be represented at a higher level of abstraction using ontologies [65]. Therefore an ontology can be seen as a formal representation of concepts along with their relationships [65]. It can express semantics in a much richer way than other representation models [66]. Ontologies consist of classes, their instances and properties between these instances [66]. They also use logic languages like first order logic or description logic to formalize axioms and increase their expressiveness [66]. They are widely used in the area of Semantic Web to express meaning [66].

Individuals

Individuals – also known as instances – can be seen as the objects of the conceptualized domain [67].

Classes

The classes of an ontology are the “sets that contain individuals” [67]. A class *c* consists of formal mathematical statements which describe the conditions which an individual needs to satisfy for being member of *c* [67]. Similar to object oriented programming, a class may have a number of subclasses [67].

Properties

The properties are simply the relations between two individuals [67].

Figure 7 which is taken from [67], illustrates an example of classes, properties and individuals. There are three classes; “Person”, “Country” and “Pet”. Moreover there are seven individuals; “Gemma”, “Matthew”, “Italy”, “England”, “USA”, “Fluffy” and “Fido”. In addition to that, there are three properties; “hasSibling”, “livesInCountry” and “hasPet”. The property “hasPet” links individuals “Matthew” and “Fluffy” together. This really says that Matthew has a pet which is called Fluffy. A similar connection is described with the rest of the properties. Each class contains a number of individuals which satisfy the necessary and sufficient conditions for granting the membership of this class [67]. For instance, class “Person” includes “Gemma” and “Matthew” individuals. This really says that Gemma and Matthew are considered to be persons.

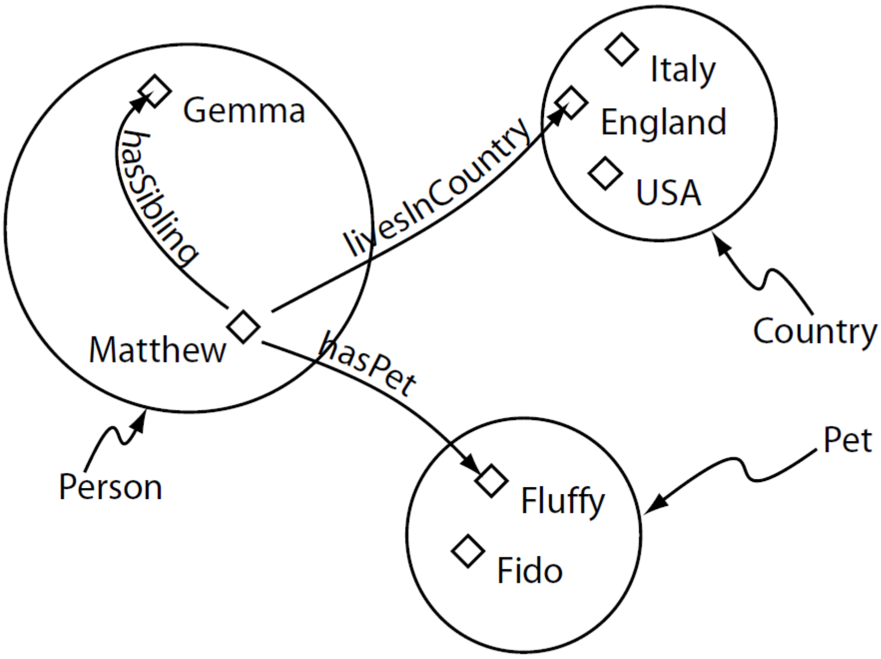


Figure 7: Classes, Properties and Individuals **[67]**

It is particularly useful to notice that an individual *d* may exist even if it does not satisfy the conditions of any class. In that case *d* is considered to be a member of a general class called “Thing” which does not impose any membership conditions. Actually all the classes are considered to be subclasses of class “Thing”. Moreover an individual may be member of more than one class if it can satisfy their conditions. This is the main difference between ontology modelling and object-oriented modelling.

2.4.1 Ontologies versus Databases

According to Martinez-Cruz et al. (2012) one can map ontology’s classes to database’s tables, ontology’s properties to database’s attributes and ontology’s axioms to database’s constraints [66]. Moreover ontology’s instances can be seen as the table’s records [66]. Although that mapping sounds sensible; ontologies are not the same as databases and should never be considered as such. Some of their differences are mentioned below.

Databases are designed for a specific application and therefore different applications need different database schemas [66]. This is not the same for ontologies which describe concepts on a specific domain and can be re-used in different applications [66]. Consequently the abstraction level of ontologies is higher than this of databases [66].

Using the properties and axioms of ontologies, one can express much more semantics comparing to the use of databases' types and constraints [66]. In addition to that, ontologies impose fewer constraints and provide more flexibility than databases. More precisely new independent instances – which do not belong to a specific class – may be created [66]. In ontological models, an instance is considered to belong to a specific class if, any only if, it satisfies all the constraints of this class [66]. However new instances which do not satisfy the constraints of any class can still be created [66]. On the other hand, database models do not allow the addition of a new record if it does not satisfy the constraints of a specific table [66].

As already mentioned, ontologies can express more semantics than databases [66]. This results to a comprehensive and precise conceptualization of a domain [66]. However ontologies suffer from performance degradation when they contain a large number of instances [66]. This happens because the ontological information is usually stored in a plain RDF file [66]. A solution to that is to store the instances in a database and have the ontology provided an interface for accessing the instances using the database [66].

2.4.2 OWL Ontology

OWL is the most recent ontology language which is created by the World Wide Web Consortium (W3C) [67]. It is becoming a standard tool for building ontologies related to the Semantic Web [66]. It supports new facilities and more operators like union and intersection as well as negation [67].

OWL has its own reasoner which can detect any inconsistencies in the logical model as well as infer the inheritance hierarchy [67]. For instance let two classes A and B with a set of restrictions R_a and R_b respectively. These restrictions are really the properties – or conditions – which an individual needs to have in order to grant the class membership. If we create an individual d by specifically declaring d as belonging to class A, then we can guarantee that d satisfies R_a. Now if we declare d as satisfying the additional set of properties R_b then the reasoner will automatically infer that d is also a member of B. This is particularly useful in constructing large ontologies since one can specify the single inheritances only and let the reasoner decide for any additional inheritances [67].

In this project OWL was used to build the two ontologies that have been described in subsection 1.4.1.

2.5 Description Logics

As already mentioned, ontologies use logic languages like description logics to formalize axioms and express semantics [66]. This section provides a brief discussion on description logics.

According to Baader et al. (2007), "Description logics (DLs) are a family of knowledge representation languages that can be used to represent the knowledge of an application domain in a structured and formally well-understood way" [68]. They can describe formal semantics in a logical structure and they are most widely used in ontology languages such as OWL [68]. The restrictions, properties and axioms of ontology languages can mainly be expressed using DLs.

A quick example to demonstrate DLs follows. Alice who is currently building an ontology *O* wishes to create a class *C* to describe all those individuals who are young persons and work in a bank or an insurance company and whose siblings are all unemployed. This can easily be done by providing a DL restriction *R* to class *C*. The restriction shall be as follows:

$$\text{Person} \wedge (\exists\ \text{hasAge.Young}) \wedge (\exists\ \text{worksIn.(Bank} \vee \text{InsuranceCompany))} \\ \wedge (\forall\ \text{hasSibling.Unemployed})$$

The restriction R contains conjunction (∧), disjunction (∨), "existential restriction constructor" ($\exists$ h.A) and "value restriction constructor" ($\forall$ h.A) [68]. We can also use negation (¬) [68].

It is particularly useful to notice that the restriction *R* can also be used as a DL query. For instance if Alice wishes to retrieve all those individuals of ontology *O* that satisfy *R* it can do so by using *R* as a DL query. DL queries can be used to reason over ontologies and retrieve some results. They are analogous to SQL queries for relational databases. So DLs are used as the main ingredient for constructing and querying ontologies.

More details about DLs and their building constructs can be found in [68].

2.6 Protègè

“Protégé is a free, open source ontology editor and knowledge-base framework” [69]. It supports the construction of OWL ontologies in an easy and user-friendly environment [69]. An OWL Ontology is stored in an RDF or OWL or XML file. Instead of constructing an ontology by manually editing its corresponding file; a user can use Protégé’s graphical user interface (GUI) to construct the ontology and let Protégé automatically export that ontology into the actual file [69]. Protégé also supports the visualization of ontologies in a hierarchical manner [69].

Protégé which is created using the Java platform supports the creation of plug-ins for adaptability and extensibility [69]. In this project, it is used to facilitate the construction of the two ontologies described in 1.4.1.

2.7 Other Frauds and Crimes

This section details the characteristics of other frauds and crimes that are used to construct the generic fraud ontology.

2.7.1 Telecommunications Fraud

Hilas et al. (2008) define telecommunications fraud “as any activity by which telecommunications service is obtained without intention of paying” [70]. Fraudsters who commit telecommunications fraud aim to receive free services or reduced-rate services in order to sell them for additional profit [71]. More precisely, if a fraudster – who managed to deceive the telecommunications service provider (TSP) – receives free calls then he or she can sell these calls illegally and make high profits [70] [71]. There are five types of telecommunications fraud. These are discussed below.

Superimposed Fraud

Superimposed fraud occurs when a mobile account is taken over by a fraudster [72]. This can be done by simply stealing victim’s SIM card or by producing a clone of that card – also known as cellular cloning [72]. The fraudster uses victim’s mobile account excessively by making or selling phone calls [72]. Of course, the bill is sent to the victim who is liable to pay.

Subscription Fraud

Subscription fraud is committed when a fraudster obtains telephone services “without intention to pay” for them [1] [71] [72]. Similar to superimposed fraud, fraudsters can use these services by making or selling phone calls [72]. A fraudster can subscribe to obtain telephone services, make excessive use of them and then vanish without paying the bill; burdening TSP with the losses [72].

Premium Rate Fraud

Premium rate fraud is perpetrated by exploiting the use of premium rate services. In general, a person *A* can setup a premium rate service *S* by reaching an agreement with TSP. More precisely, TSP is committed to pay *A* whenever somebody calls *S* [1]. If *A* is a fraudster, he or she will either make a “large number of short calls” or a “small number of long calls” to *S* [1]. The exact behaviour of *A* depends on whether TSP pays *A* based on the number or duration of calls to *S* [1]. The rationale behind this fraud lies in the fact that fraudsters which are making the calls to *S* will never pay their bills [1] [71]. Given that TSP is not aware of the fact that *A* is a fraudster; they will be liable to pay *A* for all these calls to *S* even if callers’ bills remain unpaid.

PABX Fraud

Private Automated Branch Exchange (PABX) is equipment which facilitates call routing within an organization [71]. PABX can route calls to internal or external lines via an automated menu [71]. In other words, we can think of PABX as an automatic call centre or a switchboard which can connect calls with other lines inside or outside the organization.

PABX fraud occurs when a fraudster obtains unauthorized access to a PABX [1] [71]. The fraudster can then exploit the routing facilities of PABX by making expensive external calls [1] [71]. The fraudster can also sell these calls or inflate the revenue of a premium rate service [71]. Notice that PABX owners are liable to pay for all these calls since it is their PABX which is being charged [71].

Cramming

Cramming occurs when fraudsters charge victims' mobile accounts with services they have never obtained [73] [74]. In general, TSP allows people to subscribe for services which are provided by a third party. In this way, the third party is allowed to charge people's mobile accounts accordingly. Fraudsters can deceive people to subscribe for fake services [73] [74]. This can easily be done by prompting people to send a text message to a specific number or give out their mobile number on the Internet [73].

The charges are most likely to be of small amounts and have a generic description so as they can easily be overlooked [73] [74]. Moreover, the charges may occur only once or many times [73] [74].

2.7.2 Securities Fraud

This subsection describes the different fraud types which occur in the area of securities market. There are three main fraud types. These are discussed below.

Insider Trading

Insider trading occurs when a person *A* trades – buys or sells – the stock of a corporation based on inside, non-public information [75] [76] [77] [78]. Person *A* is usually an employee of that corporation – also known as insider – or he or she is highly related with an insider [75] [78]. The knowledge of this information gives *A* an unfair advantage over outsiders since he or she can trade stock without risk [77]. In particular, if the information was that the revenue of the company was increased then *A* would buy more stock [78]. On the other hand, if the revenue was decreased then *A* would sell his or her stock immediately [78].

Insider trading affects securities market and can seriously damage corporation's reputation [75] [77].

Ponzi Scheme

Ponzi scheme is a fraudulent investment scheme which deceives investors by paying them unusual high profits at the initial state [76] [79]. This is simply done by giving the money of new investors, who have just been added to the scheme, to older investors [76] [79]. In other words, ponzi scheme is based on virtual profits instead of profits made by real business activities [76] [79]. Due to these virtual profits, people keep investing money in the scheme [76] [79]. Notice that ponzi scheme requires new investors continuously in order to keep working [76] [79]. As soon as there are no new investors, the scheme will collapse and investors will lose their money [76] [79].

Pump and Dump

Pump and dump scheme occurs when fraudsters inflate the price of stocks of a corporation [80] [81]. This can be done by exaggerating the financial health of the corporation or by claiming that they have inside information about this corporation [80] [81]. The fraudsters – who are usually company insiders – urge investors to quickly buy or sell corporation's stock by exaggerating the stock price using the Internet or social media [80] [81]. This makes investors to start buying or selling corporation's stocks, causing an unusual demand on these stocks which inflates – pumps – their price [80] [81]. As soon as fraudsters stop exaggerating the stocks' price, the latter will deflate – dump – and investors will lose their money [80] [81]. Pump and dump scheme is usually perpetrated with small corporations as it is easier to inflate the price of their stock [81].

2.7.3 Insurance Fraud

This subsection describes the different fraud types related to insurance industry. Insurance fraud can be committed by both the insurer and the insured [82] [83] [84]. More details can be found below.

By Insurer

One way in which insurance fraud is committed is when the insurer refuses to compensate the insured even if insured's claim is legitimate [82]. The insurer may lie to insured by stating that he or she is not eligible to receive compensation. Clearly, this is completely unethical and should be regarded as a fraud.

By Insured

There are two type of insurance fraud that can be committed by the insured [83] [84]. These are the soft and hard insurance fraud [83] [84].

Anyone who inflates the damages of a claim, which is fully legitimate, commits soft insurance fraud [83] [84]. For instance, the insured may exaggerate their losses after their house was burgled [84]. In addition to that, providing false information when applying for insurance can also be considered as soft insurance fraud [84]. For instance, if a person, who applies for health insurance, conceals a serious illness which he or she had in the past then he or she commits soft insurance fraud [84].

On the other hand, “hard [insurance] fraud is a deliberate attempt either to stage or invent an accident, injury, theft, arson” [84]. The insured invents or stages a disaster in order to receive compensation [83] [84]. Clearly, this is a more serious type of insurance fraud than soft insurance fraud.

2.7.4 Mortgage Fraud

This subsection details the different fraud types that can be perpetrated by people who apply for property loans. There are two types of mortgage fraud which are described below [85] [86].

Property Fraud

Property fraud occurs when the borrower applies for a property loan by hiding or providing false information [85] [86]. For instance, the borrower may conceal any other debts they have or exaggerate their income [85] [86]. The borrower’s motivation is to receive just one loan in order to buy a property; with intention to repay that loan at some time in the future [85] [86]. Property fraud is less serious than profit fraud – which is described below – given that the borrower repays the loan on time [85] [86].

Profit Fraud

Profit fraud occurs when a fraudster applies for one or more property loans with no intention to repay them [85] [86]. This is usually done in combination with identity fraud – which is described in 2.1.3 [86]. In other words, the fraudsters use false identity in order to get the loan and vanish [85] [86]. The fraudsters may also lie for their employment in order to increase the chances of loan approval [85] [86]. Profit fraud is usually committed in collaboration with unfaithful insiders [86]. More precisely, an unfaithful bank officer may approve a loan application even if he or she knows that the applicant is a fraudster [86].

Another interesting way of committing profit fraud is by virtually selling a property “multiple times between fake sellers and buyers” in order to “create the illusion” that the property has significant value [86]. These transactions are only done in paper and show that a property has been sold many times with huge amounts of money [86]. Using this approach – which is widely known as flipping – a fraudster can claim that he or she wishes to obtain a loan for buying that “high-valued” property [86]. If the loan application is approved, the fraudster will get the money and vanish; exposing the bank to high losses [85] [86].

2.7.5 Corporate Crime

This subsection details the different crimes which are perpetrated by corporations. There are three such crimes and they are described below [3].

Corporate Violence

Corporate violence which refers to “the exposure of employees to harmful and unsafe working conditions” may be regarded as a type of corporate crime [3]. The consequences of corporate violence to employees vary between injuries, health problems and “premature deaths” [3]. Employers who expose their employees to dangerous conditions can be regarded as criminals [3].

Economic Exploitation

There are various ways in which an employer can economically exploit their employees. The employer may refuse to pay employees for their overtimes and pensions as well as refuse to contribute to employee’s social security [3]. In addition to that, an employer may illegally reduce employees’ wages as well as pay wages which are lower than minimum [3].

Product Misrepresentation

Product misrepresentation which refers to the advertisement of false information can also be considered as a type of corporate crime [3]. Corporations which use false information about their products may seriously harm their customers [3]. Consider, for instance, a case where a corporation advertises a product as being nut free although it contains nuts. An allergic person who consumes that product may get serious health problems or even die. In such a case that corporation has committed a crime.

2.7.6 Governmental Crime

Governmental crime refers to the various types of crime which are committed by government itself [3]. Illegal actions of the government can cause violation of civil rights, economic losses as well as physical harm [3]. Following [3], there are two types of governmental crime and these are described below.

Political Crime

Political crime refers to all these crimes which are committed by politicians and/or political parties [3]. Politicians may exploit their political power by accepting briberies from corrupted citizens who wish to receive favourable treatment [3]. Moreover, some corrupted politicians pay enormous amounts of money to their electoral campaign [3]. Although this is treated as a legitimate behaviour, it is completely unfair because a significant proportion of this money is usually used to bribe voters [3]. As a result, the richest politician or political party has higher probability to win the elections [3].

State-organized Crime

This type of crime refers to the complicity of government in the organized crime [3]. Following Friedrichs (2009), examples of organized crime are assassination, conspiracy, embezzlement, kidnapping, smuggling, spying and terrorism [3].

2.7.7 Occupational Crime

Occupational crime refers to the various different crimes which are committed by people during their occupation [3]. There are two types of occupational crime and these are described below.

Academic Crime

Academic crime refers to the various inappropriate activities which are committed by academics including professors, researchers and students [3]. Academics who conduct unsafe or harmful experiments may be regarded as criminals; since they can cause serious damage to experiment participants [3]. In addition to that, academics may perform plagiarism or fabricate the results of their work to make them look more promising [3].

Another way in which academic crime occurs is when students apply for an entry to a university with false information [3]. Clearly, this can give them an unfair advantage over other students [3].

Employee Crime

This type of crime is committed by employees and results in employer's victimization [3]. Corrupted employees may take advantage of their position to steal money from their employers [3]. For instance, consider an unfaithful cashier in a supermarket who allows their friends to take items free of charge [3]. In addition to that, employee crime can also occur when employees steal the trade secrets of the company they are working to, with the ultimate goal to sell them to competitive companies [3]. According to Friedrichs (2009), trade secrets include "ideas, designs, and formulas" [3].

2.7.8 Income Tax Evasion

According to Friedrichs (2009), income tax evasion occurs when people fail to provide accurate income reports to the government; in order to avoid paying income taxes [3]. People may not provide income reports at all or even worst they may provide false income reports [3]. For instance, people may report less annual income than they actually gain or hide their employment [3]. Clearly, this can be considered as a form of crime since it victimizes honest citizens who pay income taxes as it is required [3].

2.7.9 Money Laundering

Money laundering is the concealment of real source of money which has been gained from illegal activities [87] [88]. Drug trafficking is one of the most common illegal activities which is highly related to money laundering [88].

There are various ways in which criminals can commit money laundering [89] [90]. According to Quirk (1997), "smurfing" is one [89]. In general, banks require from depositors to report their source of money when the amount to be deposited is higher than a certain threshold [89]. This procedure is also known as the "minimum cash reporting requirement" [89]. Following Quirk (1997), "Smurfing involves the use of multiple cash deposits, each smaller than the minimum cash reporting requirement" [89].

Transferring money to foreign countries, which have more relaxed rules on cash deposits, is another way in which money laundering is committed [89] [90]. In addition to that, corruption inside banks can help criminals to launder their money [89]. More precisely, unfaithful bank officials may receive briberies to accept dirty money for deposition [89]. If criminals manage to launder illegal money by depositing it to banks, then they can easily use it as if it was legitimate [88].

2.7.10 Computer Fraud

Computer fraud refers to the different types of fraud which are perpetrated using computers. Phishing and site cloning which are described in 2.1.1 can be considered as types of computer fraud. In addition to that, computer fraud can be committed using malicious software [91]. This is described below.

Fraud using Malicious Software

This refers to the use of malicious software like viruses, Trojan horses and worms which exploit computer vulnerabilities to obtain unauthorized access to a computer or network [91] [92] [93]. Hackers use malicious software to spy on computer users and steal their credentials as well as personal and identity information [91]. These can be used to commit other types of fraud including credit card and identity fraud.

2.7.11 Friendly Fraud

Friendly fraud occurs when a cardholder makes a transaction and then declares it as illegitimate [94] [95] [96]. According to [95], friendly fraud can be committed very easily [95]. Cardholders may use their credit cards to buy some goods or pay for some services and then claim that they don't recognise these payments [94] [95] [96]. If the fraud cannot be proved, the merchant or issuing bank is liable to compensate the cardholder [95]. The result is that cardholders receive free goods or services whereas merchants or issuing banks loss their money [95].

Notice that friendly fraud can be considered as another type of credit fraud – which is described in 2.1; although the literature does not classify it as thus.

2.8 Conclusion

In this chapter the background research – which has been undertaken during the project – was presented. The chapter started from the different fraud types that occur in credit industry. It then moved to a general description of the various techniques which can be used to detect the above fraud types. The related articles concerning these detection techniques were also mentioned. In addition to that, a brief discussion about ontologies and the technologies that were used in this project was made. Finally the characteristics of other frauds and crimes – which form the basis of generic fraud ontology – were discussed.

The next chapter details the design process of project's deliverables.

3 Design

This chapter details the process of designing the two ontologies mentioned in 1.4.1 as well as the infrastructure of expert system. It starts with the characteristics of techniques for detecting credit fraud. These form the basis of credit fraud detection ontology introduced in 1.4.1. It then moves to the design of this ontology and its associated expert system. Finally, details about the design of generic fraud ontology are followed.

3.1 General Characteristics of Detection Techniques

Technique	Learning	Has Para-meters	Ease of Interpre-tability	Size of Training Sample	Finds Rare patterns	Detects new frauds
Artificial Neural Networks	Supervised	Yes [27]	No [32]	Large [97]	No	No
Support Vector Machines	Supervised	Yes [55]	No [32]	Large & Small [31] [54] [97]	No	No
Bayesian Belief Networks	Supervised	Yes [49]	Yes [48]	Large	No	No
Decision Trees	Supervised	No [98]	Yes [32]	*N/A*	No	No
Outlier Detection	Unsupervised	Yes [39] [56]	No	*Don't Care: no model construction*	Yes [27]	Yes [27]
Peer Group Analysis	Unsupervised	Yes [42]	No	*Don't Care: no model construction*	Yes	Yes
Hidden Markov Model	Unsupervised	Yes [24]	No	Large & Small [50]	Yes	Yes
Artificial Immune System	Semi-supervised	Yes [45]	No	*N/A*	Yes	Yes
Nearest Neighbour	Supervised	Yes [31]	Yes [61]	Small [99] [31]	No	No

Table 1: Characteristics of Detection Techniques

Recall that the expert system advises software developers as to which detection technique they should implement by taking into account their needs. Software developers will provide answers to a number of questions in order to assist the expert system in deciding the most appropriate technique. These questions are derived from the various characteristics of detection techniques that have been extracted during the research process. Table 1 illustrates these characteristics.

Notice that references are used to indicate the sources from where each cell value was derived. The cells with no references indicate that the particular value has been derived from the writer’s understanding on the detection technique. Not applicable values – “N/A” – are used to indicate that it was not possible to find any information in the literature about the value of a specific cell. In order to understand “Don’t Care” values consider the following example. Let’s assume that a software developer is asked by the expert system whether the technique – that he or she wishes to implement – should need large or small number of training samples. Let’s further assume that the software developer replies with the “Small” option. Normally any technique that has a different cell value than “Small” – in the “Size of Training Sample” column of table 1 – should be excluded from the candidates’ list. However, the expert system should not exclude any technique *t* which has a “Don’t Care” value at that specific cell because *t* is not meant to be affected by user’s answers. For instance the outlier detection technique which does not involve any model construction – and hence does not need any training samples – should remain in the candidates’ list regardless of the user’s answer over “Size of Training Sample” question.

In order to better understand the contents of table 1, its first row is explained in more details. This includes the characteristics of Artificial Neural Networks (ANNs):

- ANNs support supervised learning
- ANN models require a number of parameters to be specified by the user and this increases their implementation complexity.
- ANNs cannot easily be interpreted. This means that it is not obvious for an expert user to understand why an ANN model predicted a specific outcome. There are other techniques like DTs which support a better interpretability.

- ANNs need a large number of training samples in order to correctly construct their predictive model. There are other techniques like SVMs and HMMs whose models can be constructed using any number of training samples – large or small.
- Finally ANN models cannot find rare patterns in data – also known as outliers – and cannot detect new fraud types. Notice that these last two characteristics can be satisfied by unsupervised and semi-supervised techniques only.

3.2 Specific Characteristics of Detection Techniques

Recall the different implementations of detection techniques which were found in the literature and are mentioned in 2.3. These implementations are the actual techniques which are reported by the expert system. The reason why expert system does not report the general techniques – for example ANN, SVM – but their actual implementations is because there are much more characteristics available when considering both the characteristics of general techniques as well as the characteristics of their specific implementations. This increases the probability to find an implementation which is more suitable to the user. Therefore, the questions which the expert system asks are based on both the general and specific techniques and the reported result includes the matching score of each specific technique only. More information on this can be found in 4.1.9.

The specific characteristics of various implementations are divided into subsections based on the type of credit fraud the implementations can detect.

3.2.1 Related to Credit Card Fraud

Tables 2, 3, 4 and 5 illustrate the specific characteristics of implementations which can detect credit card fraud.

Technique	Implementations	Deals with Over-lapping	Evaluation Data Size	Optimal Para-meter Selection	Guidelines on Feature Selection	Real-world test data
ANN	LSTM [50]	No [53]	Large [50]	No	Yes [50]	Yes [50]
	Confidence-based NN [51]	No [53]	Medium [50]	Yes [51]	No	No [51]
	GANN [12]	No [53]	N/A	Yes [12]	No	N/A
BBN	Bayesian Network [25]	No [53]	N/A	No	No	Yes [25]
DT	C&RT [37]	No [53]	N/A	Yes [37]	No	Yes [37]
	C5.0 [37]	No [53]	N/A	Yes [37]	No	Yes [37]
	CHAID [37]	No [53]	N/A	Yes [37]	No	Yes [37]
SVM	BSVS [53]	Yes [53]	N/A	No	No	N/A
	QRT Model [54]	Yes [54]	Medium [54]	No	No	Yes [54]
AIS	AIS-Plain [45]	No [53]	Huge [45]	No [45]	Yes [45]	Yes [45]
	GA-AIS [49]	No [53]	Large [49]	Yes [49]	No [49]	Yes [49]
HMM	HMM [24]	No [24]	Large [24]	Yes [24]	No	No [24]
OD	Based On Distance Sum [39]	Yes [53]	Large [39]	No	Yes [39]	Yes [39]
	SmartSifter [56]	N/A	N/A	No	No	N/A
PGA	PGA [42]	N/A	Huge [42]	No	Yes [42]	Yes [42]

Table 2: Characteristics of Implementations for Detecting Credit Card Fraud (A)

Technique	Implementations	Publi-cation Year	Accuracy Level	Error	Deals with change in spending behaviour	Can detect fraud in new customers
ANN	LSTM [50]	2009	98% [50]	13.13% [50]	Yes [50]	No
	Confidence-based NN [51]	2008	91.2% [51]	13.35% [51]	No	No
	GANN [12]	2011	*N/A*	*N/A*	No	No
BBN	Bayesian Network [25]	2002	71% [25]	12.5% [25]	No	No
DT	C&RT [37]	2011	91.34% [37]	*N/A*	No	No
	C5.0 [37]	2011	92.80% [37]	*N/A*	No	No
	CHAID [37]	2011	92.22% [37]	*N/A*	No	No
SVM	BSVS [53]	2006	89% [53]	*N/A*	No	No
	QRT Model [54]	2004	82% [54]	16% [54]	Yes [54]	Yes [54]
AIS	AIS-Plain [45]	2010	90.14% [45]	96.55% [45]	No	No
	GA-AIS [49]	2008	*N/A*	5.43% [49]	No	No
HMM	HMM [24]	2008	80% [24]	15% [24]	No	No [24]
OD	Based On Distance Sum [39]	2009	89.4% [39]	*N/A*	No	No [39]
	SmartSifter [56]	2004	80% [56]	*N/A*	No	No
PGA	PGA [42]	2008	*N/A*	*N/A*	No	Yes [42]

Table 3: Characteristics of Implementations for Detecting Credit Card Fraud (B)

Technique	Implementations	Possibility of Over-fitting	Deals with noise	Deals with skewed distribution	Supports Continuous values	Supports Discrete values
ANN	LSTM [50]	No [50]	Yes [32]	Yes [50]	Yes [32]	Yes
	Confidence-based NN [51]	Yes [100]	Yes [32]	*N/A*	Yes [32]	Yes
	GANN [12]	Yes [100]	Yes [32]	*N/A*	Yes [32]	Yes
BBN	Bayesian Network [25]	Yes [33]	Yes [48]	*N/A*	No [101]	Yes [102]
DT	C&RT [37]	Yes [37]	Yes [31]	No [37]	Yes [31]	Yes [32]
	C5.0 [37]	No [31]	Yes	No [37]	Yes	Yes [32]
	CHAID [37]	Yes [37]	No [36]	No [37]	No [32]	Yes [32]
SVM	BSVS [53]	No [31]	No [50]	Yes [53]	No [103]	Yes [103]
	QRT Model [54]	No [31]	No [50]	Yes [54]	No [103]	Yes [103]
AIS	AIS-Plain [45]	*N/A*	No [45]	No	Yes [45]	Yes [45]
	GA-AIS [49]	*N/A*	No [45]	No	*N/A*	*N/A*
HMM	HMM [24]	No	Yes [104]	No	Yes	No [24]
OD	Based On Distance Sum [39]	*Don't Care: no model construction*	No	Yes [39]	Yes [39]	No [39]
	SmartSifter [56]	*Don't Care: no model construction*	No	Yes	Yes [56]	Yes [56]
PGA	PGA [42]	*Don't Care: no model construction*	No	*Don't Care*	Yes [42]	No [42]

Table 4: Characteristics of Implementations for Detecting Credit Card Fraud (C)

Technique	Implementations	Model Construction Approach	Can detect fraud which occurs	Real-Time Detection
ANN	LSTM [50]	Multiple-User	Online & Offline	*N/A*
	Confidence-based NN [51]	Multiple-User	Online & Offline	*N/A*
	GANN [12]	Multiple-User	Online & Offline	*N/A*
BBN	Bayesian Network [25]	Multiple-User	Online & Offline	*N/A*
DT	C&RT [37]	Multiple-User	Online & Offline	*N/A*
	C5.0 [37]	Multiple-User	Online & Offline	*N/A*
	CHAID [37]	Multiple-User	Online & Offline	*N/A*
SVM	BSVS [53]	Personalised [53]	Online & Offline	*N/A*
	QRT Model [54]	Personalised [54]	Online & Offline	*N/A*
AIS	AIS-Plain [45]	Multiple-User	Online [45]	*N/A*
	GA-AIS [49]	Multiple-User	Online & Offline	No
HMM	HMM [24]	Personalised [24]	Online & Offline	Yes [24]
OD	Based On Distance Sum [39]	Personalised [39]	Online & Offline	*N/A*
	SmartSifter [56]	*N/A*	Online & Offline	Yes [56]
PGA	PGA [42]	Multiple-User	Online & Offline	No [42]

Table 5: Characteristics of Implementations for Detecting Credit Card Fraud (D)

The use of references is done in the same way as described in 3.1. This is true for the special values too – namely “Don’t Care” and “N/A”.

In order to better understand the contents of tables 2, 3, 4 and 5 their first rows are explained in more details. These are related with the characteristics of LSTM implementation which is based on ANN models:

- First of all, LSTM cannot deal with overlapping data.
- A large data set has been used to evaluate that implementation. With large data set, the writer of this report means that the number of testing samples that were used was between 5001 and 20000 inclusive. The other options are small – that is between 1 and 500 inclusive; medium – that is between 501 and 5000 inclusive; and huge – that is greater or equal to 20001.

- There is no any information in the LSTM's article for optimal parameter selection. There are other implementations which use mechanisms to discover optimal values for parameters.

- In the LSTM's article there are guidelines as to which features should be selected to achieve the best results.

- The evaluation has been done with real-world data taken from a bank.

- The LSTM article has been published on 2009.

- The accuracy level which has been achieved during the evaluation process was 98%. This is the true positive percentage. In other words, it shows the percentage of fraudulent transactions which were detected by that implementation.

- The error is 13.13%. Here there is no a clear definition of error because different implementations make different error measurements. Nevertheless, the most common measurements are the false positive and false negative rates. These are the legitimate transactions which are classified as fraudulent and the fraudulent transactions which are classified as legitimate.

- LSTM can deal with change in spending behaviour. Customers may periodically make purchases which do not conform to their usual spending behaviour; for instance during Christmas or Easter periods. Most of fraud detection systems cannot cope with that sudden change and hence they give false alarms.

- LSTM cannot detect credit card fraud which occurs in new customers with no previous knowledge about their usual spending behaviours. This is true for most fraud detection systems too.

- There is no possibility for LSTM to over-fit the training samples.

- LSTM deals with noisy data. This means that its predictive accuracy is not affected in case there is some noise in the data.

- LSTM deals with skewed distribution. In other words, it is not affected by any imbalance distribution of training samples.

- LSTM can support both continuous- and discrete-valued attributes. Continuous-valued attributes are those which take numerical values like integers, doubles, floats etc. Discrete-valued attributes are those which take a finite set of descriptive values like “small”, “medium” or “high” only. It is particularly useful to notice that there are techniques which cannot support both types of attributes. A known procedure called discretization which maps continuous values into discrete values may be used when continuous-valued attributes are not supported by a technique.

- The model construction approach for LSTM is “Multiple-User”. This means that a generic model is constructed by taking into account a bunch of labelled training transactions which have been undertaken by many customers. The opposite of this approach is the “Personalised” where a specific model is constructed for each individual customer *c* by taking into account the transactions of *c* only [53]. In other words each customer *c* has a personalised model which encapsulates the normal spending behaviour of *c*. Chen et al. (2006) claim that personalised approaches can lead to better predictive accuracy [53].

- LSTM can detect credit card fraud which occurs either online via internet or offline via a physical shop.

- Finally the information whether LSTM can detect credit card fraud in real-time is not explicitly stated in its article; hence the “N/A” value. Unfortunately there are two implementations only which explicitly state that they can detect fraud in real time. Real-time detection is very important in credit card fraud because the earlier the detection of frauds the fewer the losses that they cause.

3.2.2 Related to Bankruptcy Fraud

Similar to 3.2.1, tables 6, 7 and 8 illustrate the specific characteristics of implementations which can detect bankruptcy fraud.

Technique	Implementations	Deals with Over-lapping	Evalu-ation Data Size	Optimal Para-meter Selection	Guidelines on Feature Selection	Real-world test data
ANN	Bankruptcy Prediction using Neural Networks [57]	N/A	Small [57]	No	No	Yes [57]
	TV-ANN [58]	N/A	Medium [58]	Yes [58]	Yes [58]	Yes [58]
DT	Credit Scoring with Boosted Decision Trees [36]	N/A	Medium [36]	No	No [36]	Yes [36]
	C&RT [55]	N/A	Small [55]	No	Yes [55]	Yes [55]
SVM	Bankruptcy Prediction using SVM [59]	N/A	N/A	Yes [59]	Yes [59]	Yes [59]
	GA-SVM [60]	N/A	N/A	Yes [60]	Yes [60]	Yes [60]
	Bankruptcy Prediction Using SVM and Credit Bureaux Score [13]	N/A	Large [13]	Yes [13]	Yes [13]	Yes [13]
kNN	Adaptive Fuzzy K-Nearest Neighbour [61]	Yes	Small [61]	Yes [61]	Yes [61]	Yes [61]

Table 6: Characteristics of Implementations for Detecting Bankruptcy Fraud (A)

Technique	Implementations	Accuracy Level	Error	Possibility of Over-fitting	Deals with noise	Deals with skewed distribution
ANN	Bankruptcy Prediction using Neural Networks [57]	96.83% [57]	7.43%	Yes [100]	Yes [100]	No [57]
	TV-ANN [58]	73.91% [58]	*N/A*	Yes [100]	Yes [100]	No [58]
DT	Credit Scoring with Boosted Decision Trees [36]	87.56% [36]	*N/A*	Yes [37]	Yes	No [36]
	C&RT [55]	90.30% [55]	*N/A*	Yes [37]	Yes [31]	No [55]
SVM	Bankruptcy Prediction using SVM [59]	82.01% [59]	33.86% [59]	No [59]	No [50]	No [59]
	GA-SVM [60]	97.86% [60]	2.15% [60]	No [60]	No [50]	No [60]
	Bankruptcy Prediction Using SVM and Credit Bureaux Score [13]	93.03% [13]	26.09% [13]	No [31]	Yes [13]	No [13]
kNN	Adaptive Fuzzy K-Nearest Neighbour [61]	81.69% [61]	36.62% [61]	Yes	No [61]	Yes [61]

Table 7: Characteristics of Implementations for Detecting Bankruptcy Fraud (B)

Technique	Implementations	Publi-cation Year	Supports Continuous values	Supports Discrete values	Tested on Personal or Corporate Bankruptcy Prediction	Considers the Score from Credit Bureaux
ANN	Bankruptcy Prediction using Neural Networks [57]	1994	Yes [32]	Yes [32]	Corporate [57]	No
	TV-ANN [58]	2005	Yes [32]	Yes [32]	Corporate [58]	No
DT	Credit Scoring with Boosted Decision Trees [36]	2007	Yes [36]	Yes [36]	Personal [36]	No
	C&RT [55]	2010	Yes [31]	Yes [32]	Corporate [55]	No
SVM	Bankruptcy Prediction using SVM [59]	2005	No [103]	Yes [103]	Corporate & Personal [59]	No
	GA-SVM [60]	2005	No [103]	Yes [103]	Corporate [60]	No
	Bankruptcy Prediction Using SVM and Credit Bureaux Score [13]	2013	No [13]	Yes [13]	Personal [13]	Yes [13]
kNN	Adaptive Fuzzy K-Nearest Neighbour [61]	2011	Yes [99]	Yes [99]	Corporate [99]	No

Table 8: Characteristics of Implementations for Detecting Bankruptcy Fraud (C)

Notice that most of the columns of these tables are common to those of subsection 3.2.1. Therefore the reader should be able to understand their meaning since they have already been introduced in 3.2.1. For the purpose of this subsection, only the two new columns are explained in more details. These are “Tested on Personal or Corporate Bankruptcy Prediction” and “Considers the Score from Credit Bureaux” contained in table 8. The first column states whether a particular implementation has been evaluated using corporate or personal testing samples. In other words, it says whether the testing samples describe corporate or personal data. An implementation which has been evaluated using corporate testing samples gives us the confident that can detect corporate bankruptcy fraud better than personal bankruptcy fraud. The same applies with personal testing samples and personal bankruptcy fraud detection.

The second column, on the other hand, states whether the implementation takes into account the score from credit bureaux when detecting or predicting bankruptcy fraud. As table 8 suggests, only one implementation takes into account this.

3.2.3 Related to Credit Application Fraud

Similar to 3.2.1 and 3.2.2 tables 9, 10 and 11 illustrate the specific characteristics of implementations which can detect credit application fraud. To the best of writer's knowledge there are only two implementations which can detect credit application fraud. Notice that these implementations cannot be considered as belonging to any generic category of techniques – for instance ANN, SVM etc – and hence tables 9, 10 and 11 do not contain such information.

Implementation	Learning	Uses Web as Information Source	Evaluation Data Size	Optimal Parameter Selection	Guidelines on Feature Selection	Real-world test data
Application Fraud Detection Using the Web [64]	Unsupervised [64]	Yes [64]	Small [64]	No	Yes [64]	No [64]
CASS [18]	Supervised [18]	No	Small [18]	Yes [18]	Yes [18]	Yes [18]

Table 9: Characteristics of Implementations for Detecting Credit Application Fraud (A)

Implementation	Publication Year	Accuracy Level	Error	Possibility of Over-fitting	Deals with noise	Deals with skewed distribution
Application Fraud Detection Using the Web [64]	2009	84.5% [64]	13.33 % [64]	No	No	*N/A*
CASS [18]	2009	74.5% [18]	*N/A*	*N/A*	*N/A*	Yes [18]

Table 10: Characteristics of Implementations for Detecting Credit Application Fraud (B)

Implementation	Supports Continuous values	Supports Discrete values	Real-Time Detection
Application Fraud Detection Using the Web [64]	Yes	Yes	*N/A*
CASS [18]	Yes	Yes	Yes [18]

Table 11: Characteristics of Implementations for Detecting Credit Application Fraud (C)

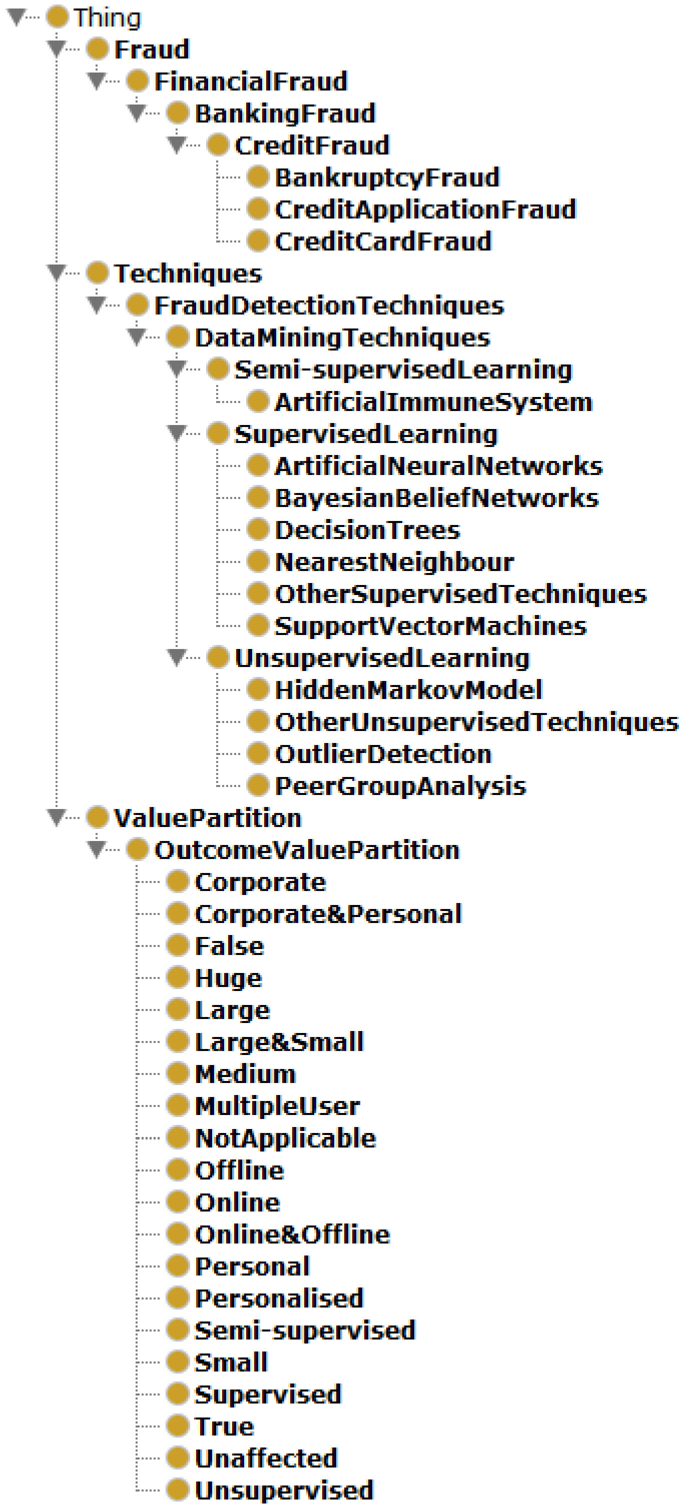
Again, most of the columns of these tables are common to those of subsections 3.2.1 and 3.2.2. Therefore the reader should be able to understand their meaning since they have already been introduced in 3.2.1. For the purpose of this subsection, only the new column is explained in more details. This is the "Uses Web as Information Source" contained in table 9. The web can be used as a source to gather useful information for assisting application fraud detection. This is done by the first implementation of tables 9, 10 and 11. It is worth mentioning, however, that involving the web adds an extra complexity to the construction of detection system.

3.3 Credit Fraud Detection Ontology

This section demonstrates the design of credit fraud detection ontology described in 1.4.1. It illustrates its class hierarchy and properties; and also presents some exemplary DL queries which can be performed on this ontology.

As mentioned earlier the ontology was constructed using Protègè.

3.3.1 Class Hierarchy



- Thing
  - Fraud
    - FinancialFraud
      - BankingFraud
        - CreditFraud
          - BankruptcyFraud
          - CreditApplicationFraud
          - CreditCardFraud
  - Techniques
    - FraudDetectionTechniques
      - DataMiningTechniques
        - Semi-supervisedLearning
          - ArtificialImmuneSystem
        - SupervisedLearning
          - ArtificialNeuralNetworks
          - BayesianBeliefNetworks
          - DecisionTrees
          - NearestNeighbour
          - OtherSupervisedTechniques
          - SupportVectorMachines
        - UnsupervisedLearning
          - HiddenMarkovModel
          - OtherUnsupervisedTechniques
          - OutlierDetection
          - PeerGroupAnalysis
  - ValuePartition
    - OutcomeValuePartition
      - Corporate
      - Corporate&Personal
      - False
      - Huge
      - Large
      - Large&Small
      - Medium
      - MultipleUser
      - NotApplicable
      - Offline
      - Online
      - Online&Offline
      - Personal
      - Personalised
      - Semi-supervised
      - Small
      - Supervised
      - True
      - Unaffected
      - Unsupervised

Figure 8: Class Hierarchy of Credit Fraud Detection Ontology

Figure 8 illustrates the class hierarchy of credit fraud detection ontology. Notice that all classes are subclasses of a generic class called "Thing". Moreover, the hierarchy starts from more generic super classes and moves to more specific subclasses. For instance, "Fraud" is super class of "FinancialFraud", which is super class of “BankingFraud” which is super class of “CreditFraud” which is super class of all the three credit fraud types. The same applies to “Techniques” and “DataMiningTechniques”.

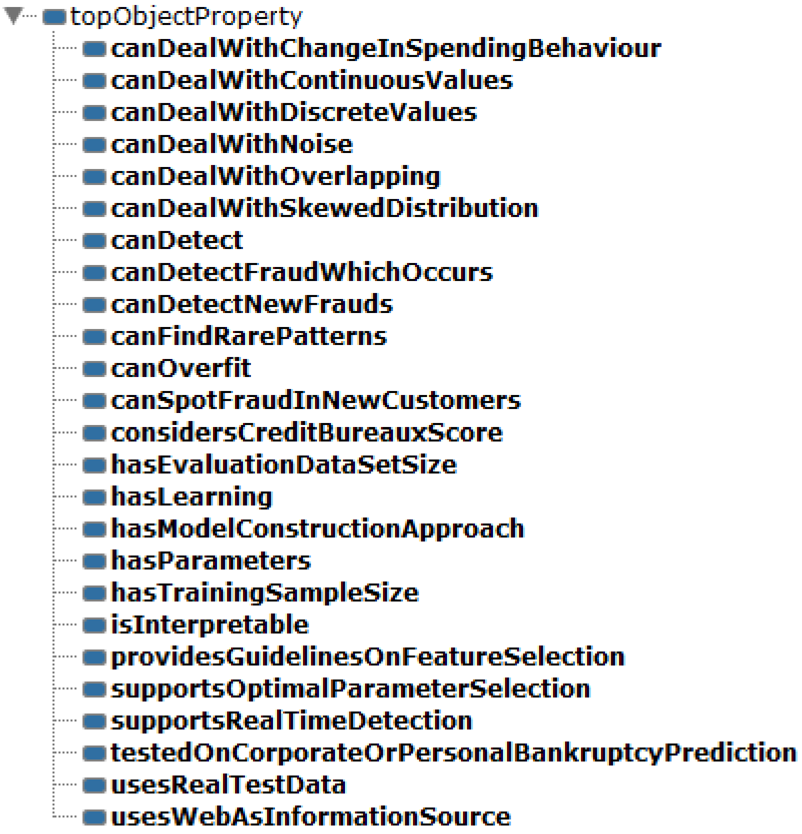
Notice also the existence of two classes; one called “OtherSupervisedTechniques” which is a subclass of “SupervisedLearning” and another one called “OtherUnsupervisedTechniques” which is a subclass of “UnsupervisedLearning”. Recall that the two implementations for detecting credit application fraud – which are demonstrated in 3.2.3 – cannot be considered as belonging to any particular class – for example ANN, SVM etc – of data mining techniques. Therefore, the purpose of “OtherSupervisedTechniques” and “OtherUnsupervisedTechniques” is to hold the implementations of 3.2.3.

In addition to that, figure 8 demonstrates a realization of value partition pattern which is suggested in [67]. This pattern can be used when we want to introduce useful values in our ontology. Here, the cell values of all tables demonstrated in 3.1 and 3.2 are introduced using the value partition pattern. Notice that the “Unaffected” class refers to “Don’t care” values. In order to better understand how these values are used in the ontology, the reader should read the following subsections – that is 3.3.2 and 3.3.3.

3.3.2 Properties

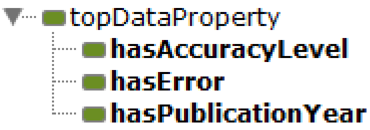
The characteristics of techniques which are described in sections 3.1 and 3.2 can be mapped into ontology’s properties.

Figures 9 and 10 illustrate the object and data properties of credit fraud detection ontology respectively. Object properties are those which link two individuals together whereas data properties are those which link one individual with a data type like string, integer, double etc. As figure 10 illustrates there are only three data properties which take numerical values. The rest are object properties. It is particularly useful to notice that all the properties of figures 9 and 10 are derived from tables 1-11.



```
topObjectProperty
    canDealWithChangeInSpendingBehaviour
    canDealWithContinuousValues
    canDealWithDiscreteValues
    canDealWithNoise
    canDealWithOverlapping
    canDealWithSkewedDistribution
    canDetect
    canDetectFraudWhichOccurs
    canDetectNewFrauds
    canFindRarePatterns
    canOverfit
    canSpotFraudInNewCustomers
    considersCreditBureauxScore
    hasEvaluationDataSetSize
    hasLearning
    hasModelConstructionApproach
    hasParameters
    hasTrainingSampleSize
    isInterpretable
    providesGuidelinesOnFeatureSelection
    supportsOptimalParameterSelection
    supportsRealTimeDetection
    testedOnCorporateOrPersonalBankruptcyPrediction
    usesRealTestData
    usesWebAsInformationSource
```

Figure 9: Object Properties of Credit Fraud Detection Ontology



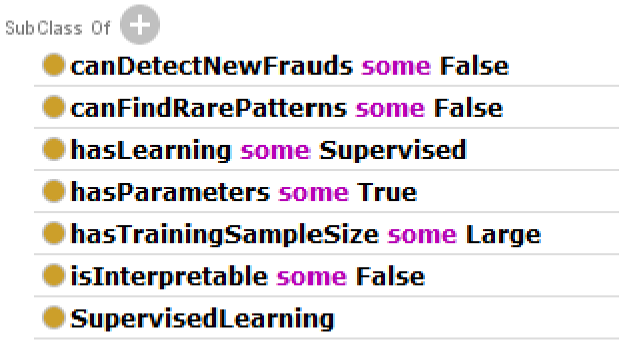
```
topDataProperty
    hasAccuracyLevel
    hasError
    hasPublicationYear
```

Figure 10: Data Properties of Credit Fraud Detection Ontology

3.3.3 Expressing Semantics

By using an example, this subsection explains how the semantics of credit fraud detection ontology have been expressed.

Figure 11 illustrates the semantics of ANN models as these are shown in table 1. These are the restrictions of ANN class. Recall – from 2.4.2 and 2.5 – that the restrictions are used to specify the conditions which an individual needs to satisfy in order to be a member of a class. In addition to that, if an individual is specifically created as a member of a class then we know that it satisfies all the conditions of that class.



SubClass Of

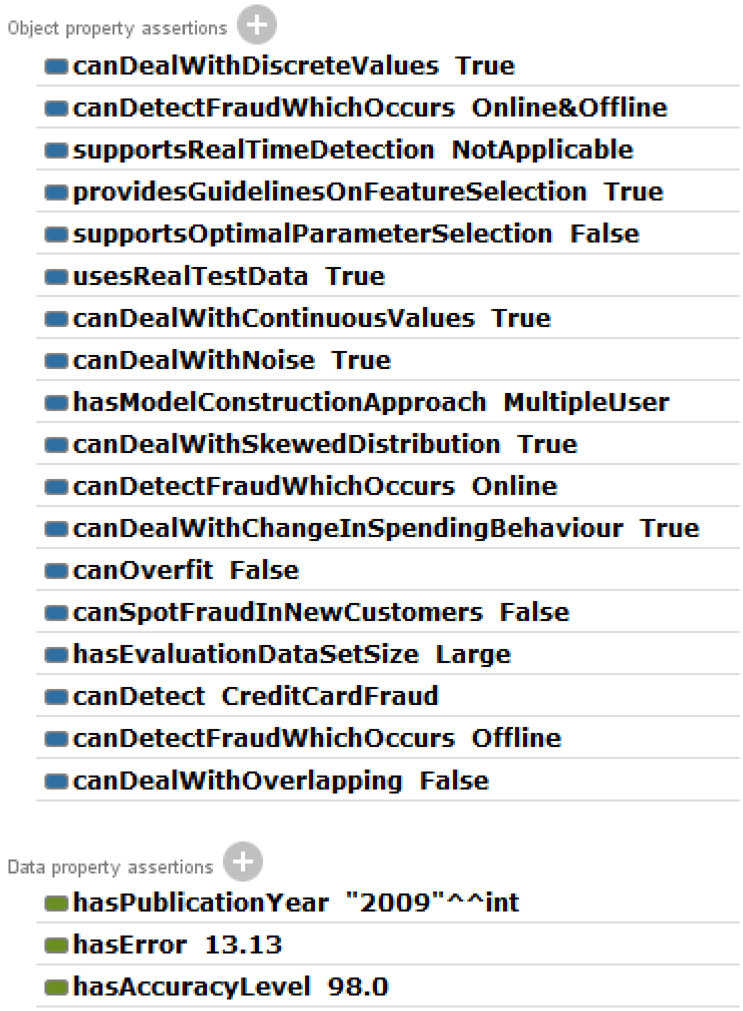
- canDetectNewFrauds some False
- canFindRarePatterns some False
- hasLearning some Supervised
- hasParameters some True
- hasTrainingSampleSize some Large
- isInterpretable some False
- SupervisedLearning

Figure 11: The Semantics of ANNs

The word “some” refers to the existential quantifier (∃). As an example consider the “canDetectNewFrauds some False” restriction. This is applied to “ArtificialNeuralNetworks” class. Recall that “False” is a subclass of “OutcomeValuePartition” class and that “canDetectNewFrauds” is an object property which links two individuals together. The above restriction means that any individual which is a member of “ArtificialNeuralNetworks” class is related with an individual which is a member of “False” class via “canDetectNewFrauds” property. In other words, if an individual *A* belongs to “ArtificialNeuralNetworks” class then there exists an individual *f* which belongs to “False” class such that *A* is related to *f* via “canDetectNewFrauds” property. This can be expressed in a more formal way as following:

$$A \in \mathit{ArtificialNeuralNetworks} \Rightarrow (\exists\ f: f \in \mathit{False} \land \mathit{canDetectNewFrauds}\ (A, f))$$

It is worth mentioning that all the implementations shown in section 3.2 are represented as individuals in credit fraud detection ontology. Figure 12 illustrates the semantics of LSTM individual which is a member of “ArtificialNeuralNetworks” class.



Object property assertions

canDealWithDiscreteValues	True
canDetectFraudWhichOccurs	Online&Offline
supportsRealTimeDetection	NotApplicable
providesGuidelinesOnFeatureSelection	True
supportsOptimalParameterSelection	False
usesRealTestData	True
canDealWithContinuousValues	True
canDealWithNoise	True
hasModelConstructionApproach	MultipleUser
canDealWithSkewedDistribution	True
canDetectFraudWhichOccurs	Online
canDealWithChangeInSpendingBehaviour	True
canOverfit	False
canSpotFraudInNewCustomers	False
hasEvaluationDataSetSize	Large
canDetect	CreditCardFraud
canDetectFraudWhichOccurs	Offline
canDealWithOverlapping	False

Data property assertions

hasPublicationYear	"2009"^^int
hasError	13.13
hasAccuracyLevel	98.0

Figure 12: The Semantics of LSTM Individual

Notice that the various values – "False", "MultipleUser" etc – are now specific individuals and not classes. For instance "False" is an individual which is a member of "False" class. Actually "False" class contains only one individual and that is the "False" individual. The same happens for all the subclasses of "ValuePartition" class. Although this synonymy is generally not a good practice; it provides much simplicity in querying the ontology and hence it has been adopted.

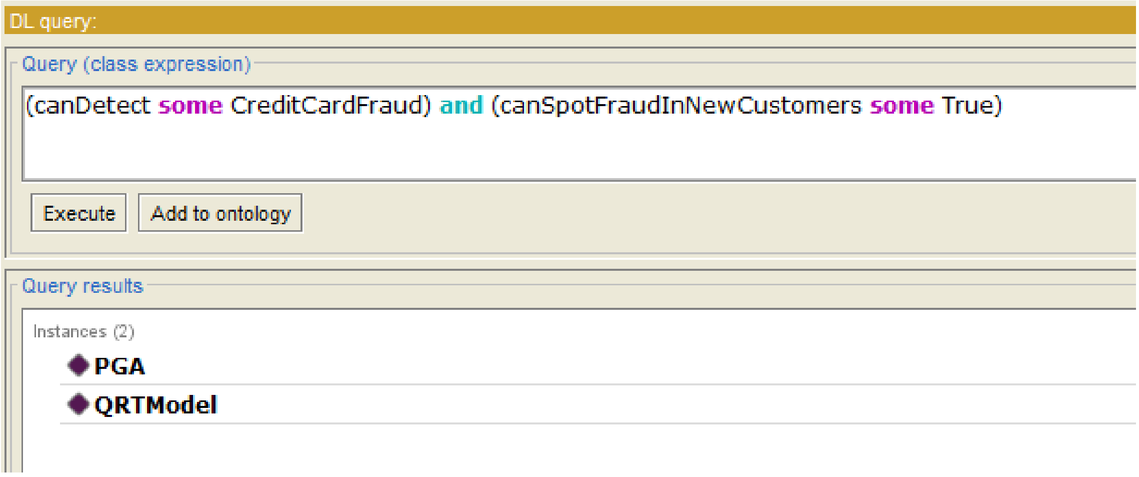
It is particularly useful to notice that the existential quantifier "some" is no longer needed when setting the individuals' semantics. This is not a surprise since the linked individuals have already been introduced. In other words, we do not refer to a class of *some* individuals but to a particular named individual. Notice that the "supportsOptimalParameterSelection" object property links "LSTM" and

"False" individuals together. On the other hand, the "hasPublicationYear" data property links "LSTM" with "2009" integer value.

3.3.4 Examples of DL Queries

This subsection demonstrates two examples of DL Queries which can be used to reason over credit fraud detection ontology in order to get some results.

The DL query of figure 13 requests all those individuals which can detect credit card fraud and can spot fraud in new customers. The resulting individuals are "PGA" and "QRTModel".

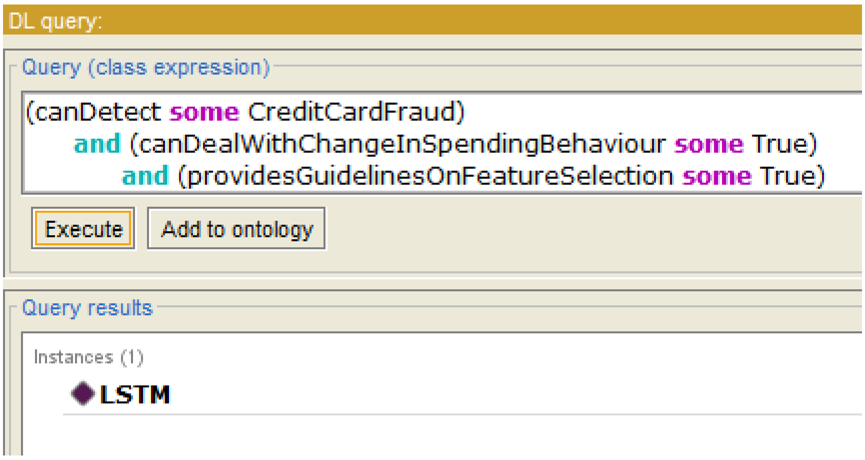


```
(canDetect some CreditCardFraud) and (canSpotFraudInNewCustomers some True)
```

Instances (2)
- PGA
- QRTModel

Figure 13: Example of DL Query on Credit Fraud Detection Ontology (A)

On the other hand, figure 14 requests all those individuals which can detect credit card fraud, can deal with change in spending behaviour and provide guidelines on feature selection. The resulting individual is "LSTM" only.



```
(canDetect some CreditCardFraud)
     and (canDealWithChangeInSpendingBehaviour some True)
          and (providesGuidelinesOnFeatureSelection some True)
```

Instances (1)
- LSTM

Figure 14: Example of DL Query on Credit Fraud Detection Ontology (B)

3.4 Expert System Design

This section contains information concerning the design of expert system. It is worth mentioning that the requirement on the expert system was that it should be a web application.

3.4.1 Model – View – Controller Separation

The main infrastructure of expert system is based on Model – View – Controller (MVC) separation. This is a design pattern which decouples model and view by using an intermediary; that is controller. In other words, view can only access model indirectly via controller. Figure 15 illustrates MVC separation.

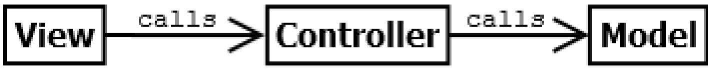


Figure 15: Model – View – Controller Separation

3.4.2 System Class Diagram

Figure 16 illustrates the class diagram of expert system. It is divided into three packages; these are model, view and controller. Since the expert system needs to be a web application, view contains its various web pages. These are not shown in figure 16 because they cannot be considered as design classes. In addition to that, figure 16 illustrates a simplified version of expert system by containing its important classes along with their important operations and attributes only. Classes, attributes and operations which are not really important are omitted for the sake of simplicity. Moreover, the data types of attributes, parameters and operations are usually omitted and only shown if they are really needed. All of the above led to a compacted version of system class diagram which is easier to understand and interpret.

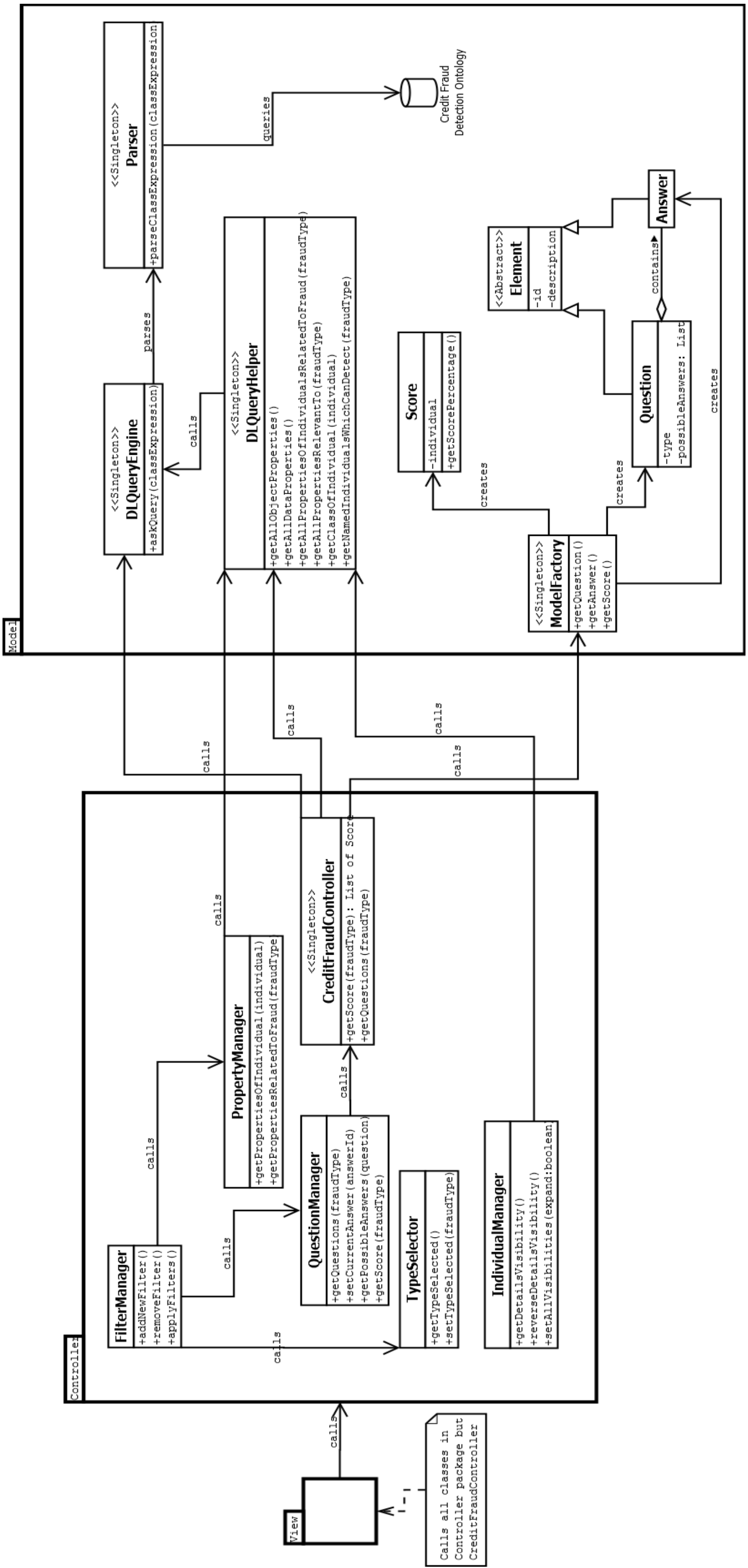


Figure 16: Class Diagram of Expert System

In order to better understand figure 16, a closer look is taken to both model and controller. Figure 17 illustrates the model part of figure 16.

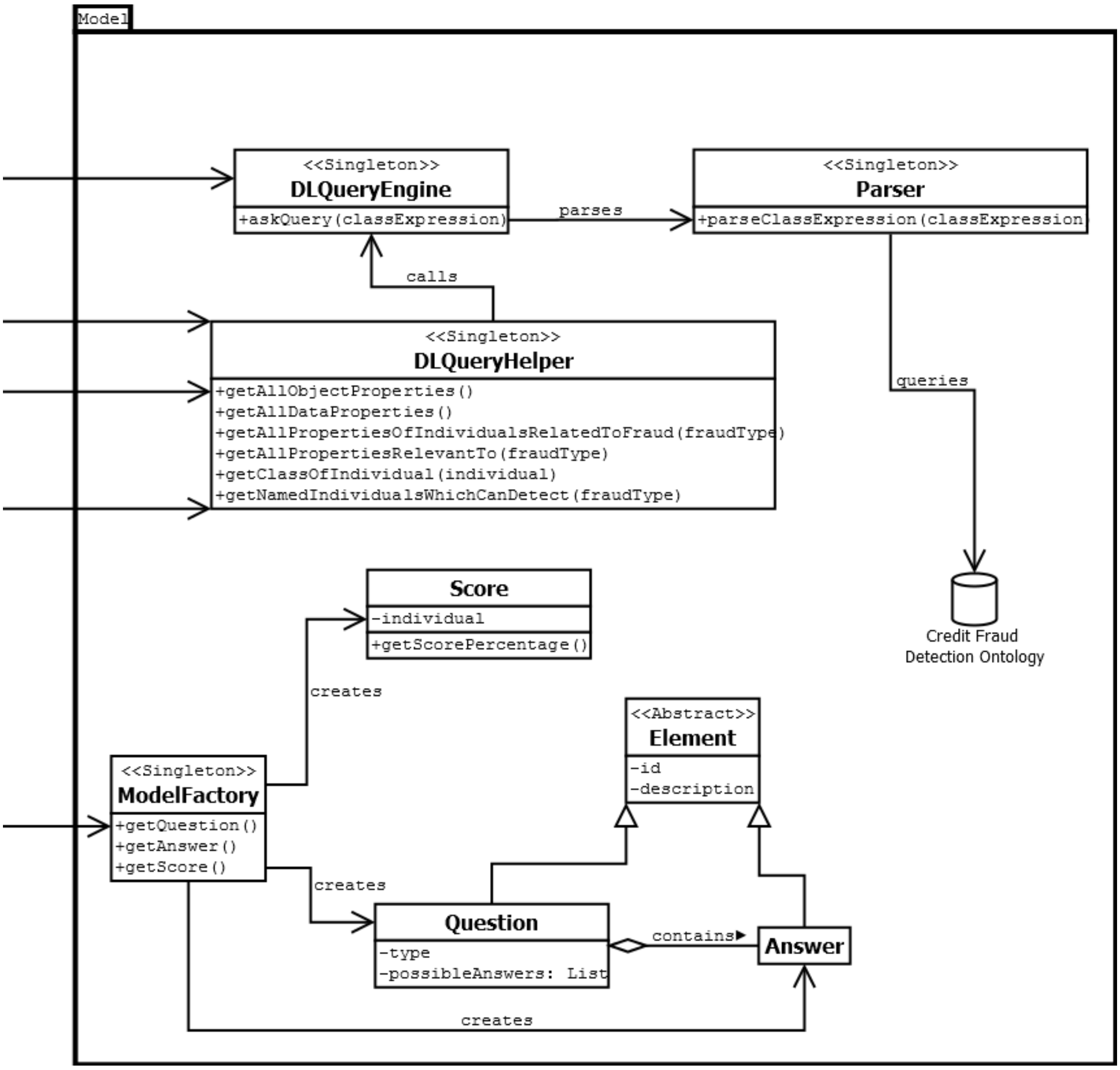


Figure 17: Model Part of Expert System Class Diagram

First of all, the classes “Parser”, “DLQueryEngine”, “DLQueryHelper” and “ModelFactory” are singletons. This means that only one instance of them should ever be needed and hence only one instance should ever be instantiated [105]. Class “Parser” is responsible to parse DL queries and send them to the ontology. Class “DLQueryEngine” is responsible to send DL queries to “Parser” as well as map the result returned from “Parser” into objects that can easily be used by the higher layers; namely controller and view. It is worth mentioning that both classes “DLQueryEngine” and “Parser” have been taken from [106] and modified to comply with project’s needs.

The classes “Question” and “Answer”, which both inherit from the abstract class “Element”, are responsible to contain the various questions and their possible answers which are asked to the users in order to understand their needs. In addition to that, class “Score” is responsible to contain the matching score of an individual – that is a detection technique. A list of scores is presented to the users to help them decide the best technique for them. For more information see 4.1.9.

The class “ModelFactory” is a pure fabrication class which is responsible to create instances of “Question”, “Answer” and “Score” classes. Factories are generally considered as a good practice because they decouple the logic of creating objects from the classes which use these objects [105].

Figure 18 illustrates the controller part of figure 16.

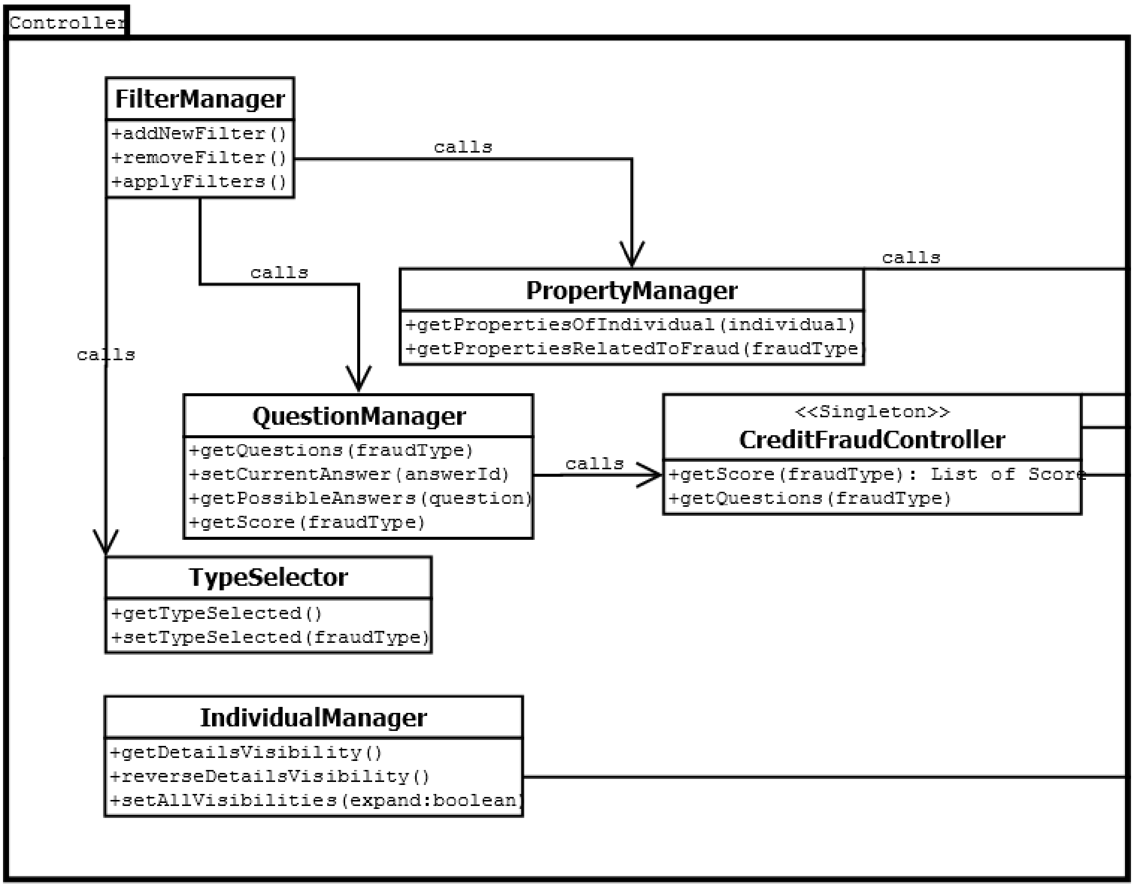


Figure 18: Controller Part of Expert System Class Diagram

The class “CreditFraudController” contains the logic for calculating the matching score of each detection technique. In addition to that, it creates and returns a list of all the questions which should be asked to the user based on the type of credit fraud to be detected. The class “TypeSelector” is responsible to contain the information as to which type of credit fraud the user wishes to detect. Moreover, the class “QuestionManager” acts as an indirection between View and “CreditFraudController”; and it holds user answers on questions.

In addition to matching scores, the expert system allows users to filter the detection techniques by their characteristics. This is facilitated by “FilterManager” class. For more information on filtering see 4.1.11.

Finally, the expert system allows users to inspect the various characteristics of detection techniques. This is facilitated by “PropertyManager” and “IndividualManager” classes. For more information on this see 4.1.10.

It is worth mentioning that all the classes in controller layer – except “CreditFraudController” – are accessed directly by view layer.

3.5 Generic Fraud Ontology

This section demonstrates the design of generic fraud ontology described in 1.4.1. It illustrates its class hierarchy and properties; and also shows how its semantics are expressed.

3.5.1 Class Hierarchy

Figure 19 illustrates the class hierarchy of generic fraud ontology. It starts with generic classes which describe high level concepts and moves to more specific subclasses. This facilitates extensibility.

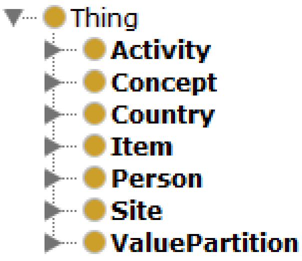
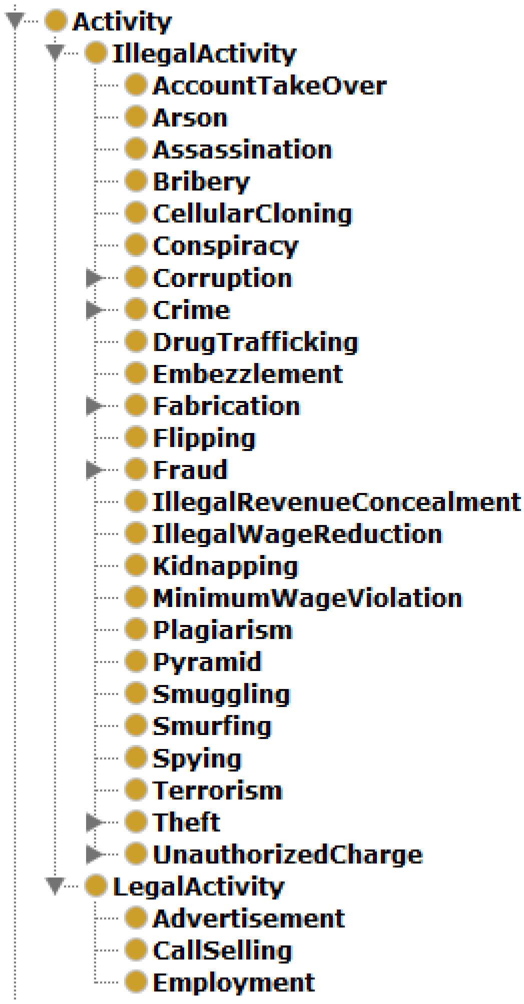


Figure 19: Class Hierarchy of Generic Fraud Ontology

As figure 20 suggests, class “Activity” has two subclasses. These are “IllegalActivity” and “LegalActivity”. As their names suggest, they describe illegal and legal activities respectively.

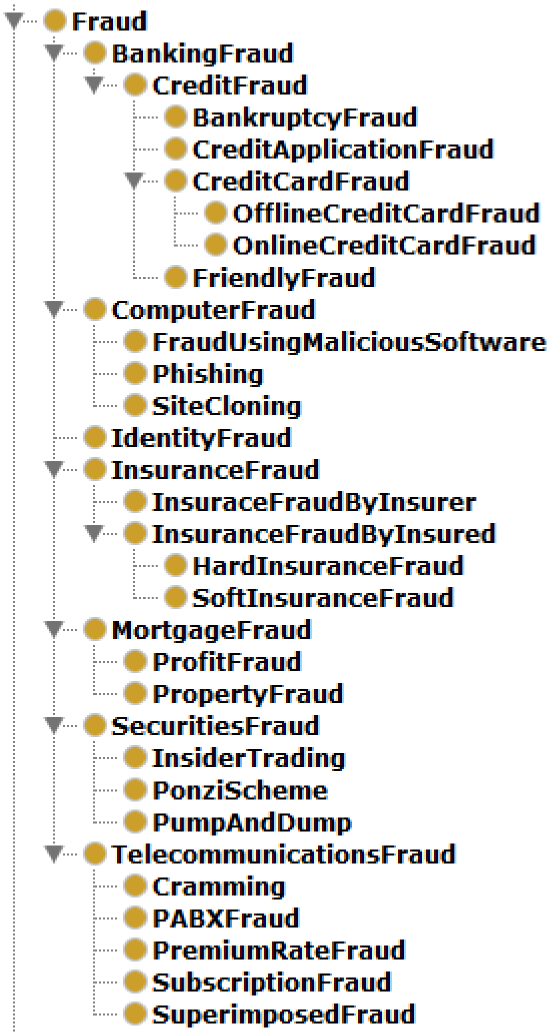


- Activity
  - IllegalActivity
    - AccountTakeOver
    - Arson
    - Assassination
    - Bribery
    - CellularCloning
    - Conspiracy
    - Corruption
    - Crime
    - DrugTrafficking
    - Embezzlement
    - Fabrication
    - Flipping
    - Fraud
    - IllegalRevenueConcealment
    - IllegalWageReduction
    - Kidnapping
    - MinimumWageViolation
    - Plagiarism
    - Pyramid
    - Smuggling
    - Smurfing
    - Spying
    - Terrorism
    - Theft
    - UnauthorizedCharge
  - LegalActivity
    - Advertisement
    - CallSelling
    - Employment

Figure 20: Class Hierarchy of “Activity” Class

The subclasses of “IllegalActivity” called “Fraud” and “Crime” form the heart of generic fraud ontology. These include the various frauds and crimes discussed in 2.7.

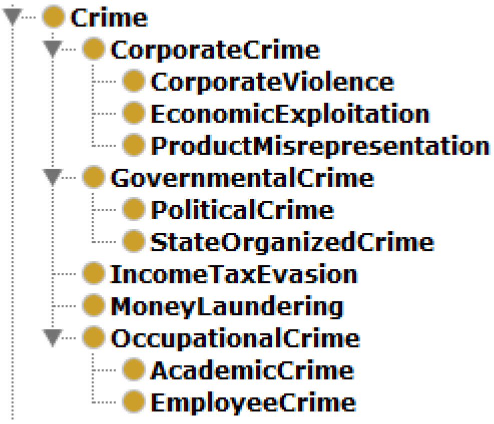
Figure 21 illustrates the full hierarchy of “Fraud” class as this is described in 2.7. Notice that the three types of credit fraud are also included in this ontology.



```
Fraud
    BankingFraud
        CreditFraud
            BankruptcyFraud
            CreditApplicationFraud
            CreditCardFraud
                OfflineCreditCardFraud
                OnlineCreditCardFraud
            FriendlyFraud
    ComputerFraud
        FraudUsingMaliciousSoftware
        Phishing
        SiteCloning
    IdentityFraud
    InsuranceFraud
        InsuraceFraudByInsurer
        InsuranceFraudByInsured
            HardInsuranceFraud
            SoftInsuranceFraud
    MortgageFraud
        ProfitFraud
        PropertyFraud
    SecuritiesFraud
        InsiderTrading
        PonziScheme
        PumpAndDump
    TelecommunicationsFraud
        Cramming
        PABXFraud
        PremiumRateFraud
        SubscriptionFraud
        SuperimposedFraud
```

Figure 21: Full Hierarchy of “Fraud” Class

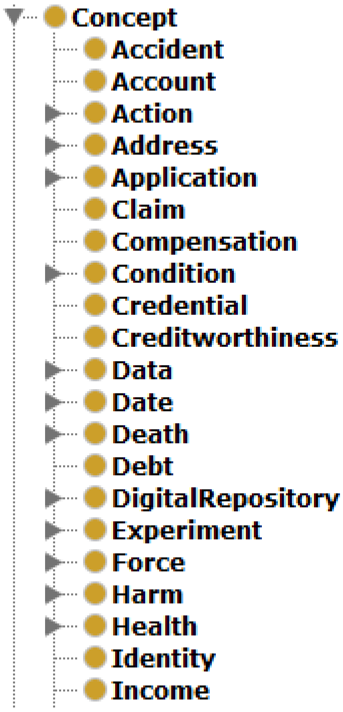
Figure 22 illustrates the full hierarchy of “Crime” class as this is described in 2.7.



```
Crime
    CorporateCrime
        CorporateViolence
        EconomicExploitation
        ProductMisrepresentation
    GovernmentalCrime
        PoliticalCrime
        StateOrganizedCrime
    IncomeTaxEvasion
    MoneyLaundering
    OccupationalCrime
        AcademicCrime
        EmployeeCrime
```

Figure 22: Full Hierarchy of “Crime” Class

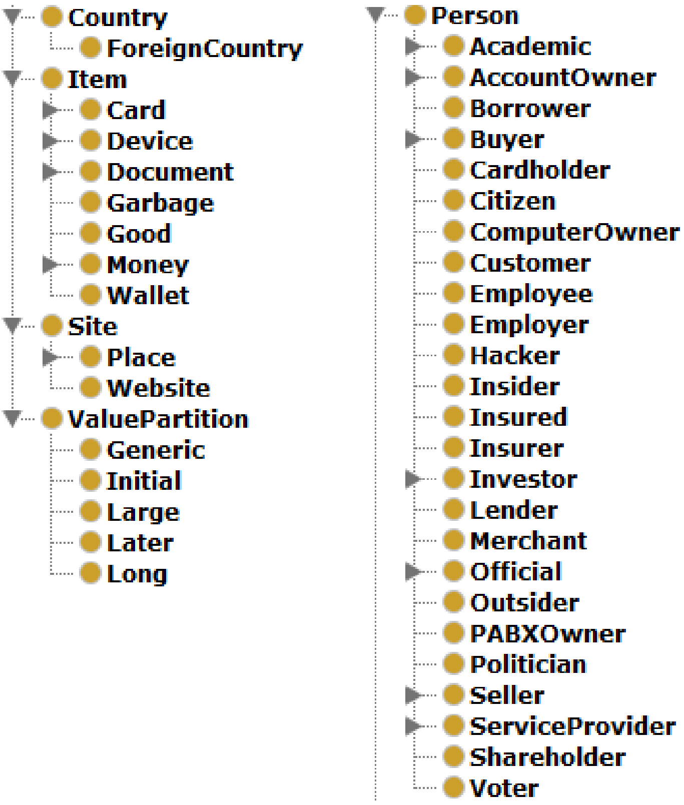
The “Concept” class of figure 19 contains subclasses which describe ideas as well as intangible things – like concepts. Figure 23 illustrates a part of its hierarchy only. The reason is that “Concept” class has a large number of subclasses and hence it is not possible to illustrate them all in this report.



Concept
- Accident
- Account
- Action
- Address
- Application
- Claim
- Compensation
- Condition
- Credential
- Creditworthiness
- Data
- Date
- Death
- Debt
- DigitalRepository
- Experiment
- Force
- Harm
- Health
- Identity
- Income

Figure 23: Part of Hierarchy of “Concept” Class

Figure 24 illustrates the direct subclasses of the rest of the classes in generic fraud ontology. It is worth mentioning that class “Item” describes tangible things for example “Card” and “Device”. Also the “ValuePartition” class describes general values and it is a realization of value partition pattern [67] which is mentioned before in 3.3.1.



- Country
  - ForeignCountry
- Item
  - Card
  - Device
  - Document
  - Garbage
  - Good
  - Money
  - Wallet
- Site
  - Place
  - Website
- ValuePartition
  - Generic
  - Initial
  - Large
  - Later
  - Long

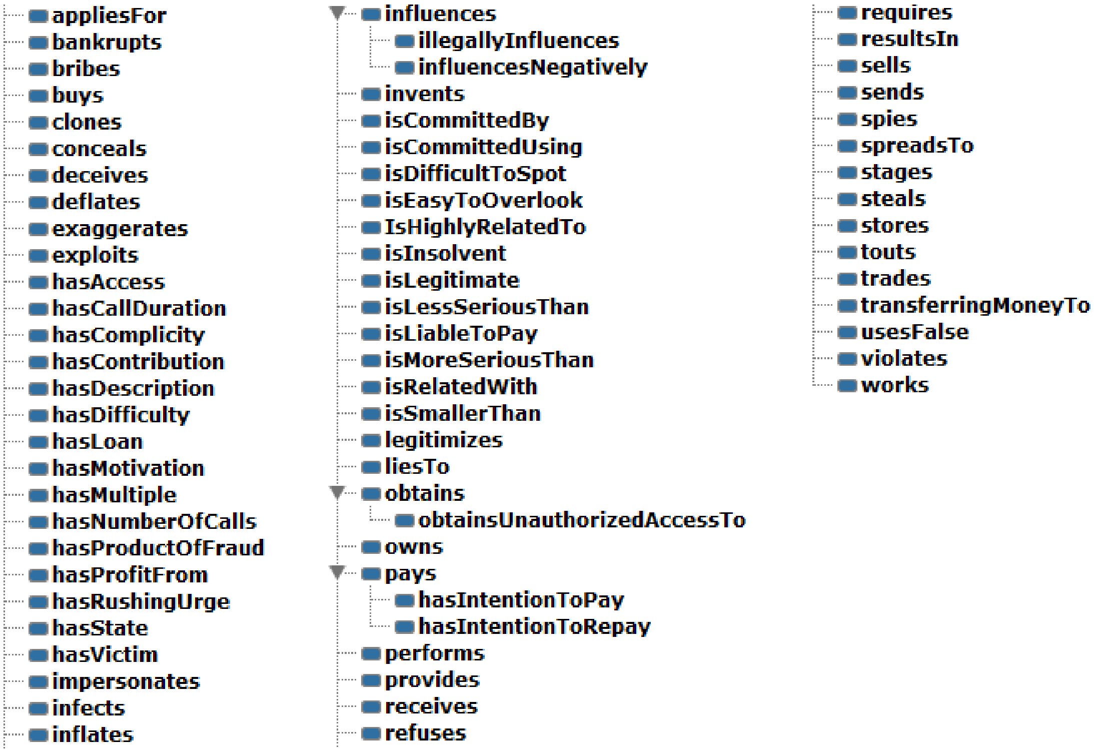
- Person
  - Academic
  - AccountOwner
  - Borrower
  - Buyer
  - Cardholder
  - Citizen
  - ComputerOwner
  - Customer
  - Employee
  - Employer
  - Hacker
  - Insider
  - Insured
  - Insurer
  - Investor
  - Lender
  - Merchant
  - Official
  - Outsider
  - PABXOwner
  - Politician
  - Seller
  - ServiceProvider
  - Shareholder
  - Voter

Figure 24: Hierarchy of Other Classes

Notice that all the classes and their associated subclasses illustrated in this subsection are used to express the semantics of various frauds and crimes. More details on this can be found in 3.5.3.

3.5.2 Properties

Figure 25 illustrates the various properties of generic fraud ontology. These along with the classes shown in 3.5.1 are used to express ontology's semantics. Subsection 3.5.3 – which follows – provides more details on how the semantics are expressed.



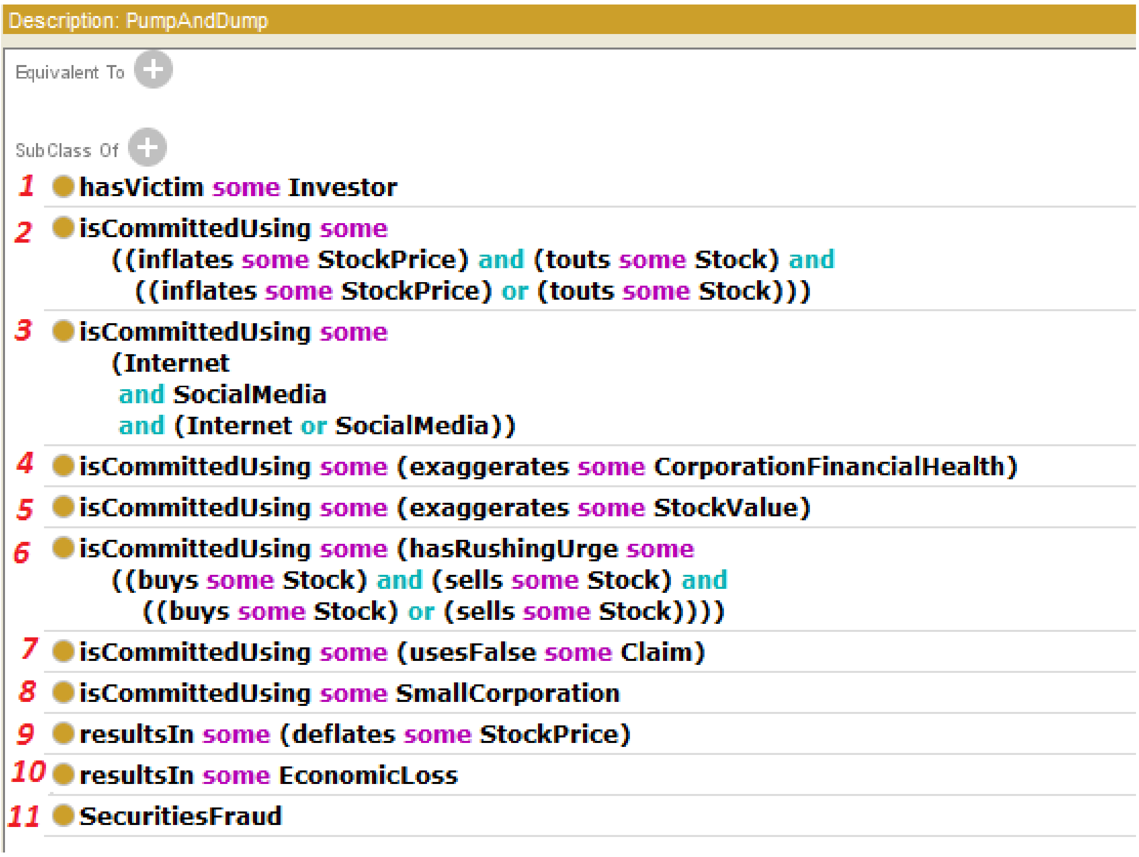
- appliesFor
- bankrupts
- bribes
- buys
- clones
- conceals
- deceives
- deflates
- exaggerates
- exploits
- hasAccess
- hasCallDuration
- hasComplicity
- hasContribution
- hasDescription
- hasDifficulty
- hasLoan
- hasMotivation
- hasMultiple
- hasNumberOfCalls
- hasProductOfFraud
- hasProfitFrom
- hasRushingUrge
- hasState
- hasVictim
- impersonates
- infects
- inflates
- influences
  - illegallyInfluences
  - influencesNegatively
- invents
- isCommittedBy
- isCommittedUsing
- isDifficultToSpot
- isEasyToOverlook
- IsHighlyRelatedTo
- isInsolvent
- isLegitimate
- isLessSeriousThan
- isLiableToPay
- isMoreSeriousThan
- isRelatedWith
- isSmallerThan
- legitimizes
- liesTo
- obtains
  - obtainsUnauthorizedAccessTo
- owns
- pays
  - hasIntentionToPay
  - hasIntentionToRepay
- performs
- provides
- receives
- refuses
- requires
- resultsIn
- sells
- sends
- spies
- spreadsTo
- stages
- steals
- stores
- touts
- trades
- transferringMoneyTo
- usesFalse
- violates
- works

Figure 25: Properties of Generic Fraud Ontology

3.5.3 Expressing Semantics

This subsection demonstrates how the semantics of various frauds and crimes are expressed in generic fraud ontology. This is done by illustrating the semantics of two frauds and crimes only. Due to space limitation, it is not possible to show the semantics of all the frauds and crimes which are available in generic fraud ontology. Nevertheless, the rest of the semantics are illustrated in the appendix chapter. It is believed that readers shall be able to understand the contents of appendix chapter if they read it in combination with section 2.7.

Figure 26 illustrates the semantics of “pump and dump” fraud which is a type of securities fraud.



Description: PumpAndDump

Equivalent To ⊕

SubClass Of ⊕

```
1  ● hasVictim some Investor
2  ● isCommittedUsing some
        ((inflates some StockPrice) and (touts some Stock) and
          ((inflates some StockPrice) or (touts some Stock)))
3  ● isCommittedUsing some
        (Internet
         and SocialMedia
         and (Internet or SocialMedia))
4  ● isCommittedUsing some (exaggerates some CorporationFinancialHealth)
5  ● isCommittedUsing some (exaggerates some StockValue)
6  ● isCommittedUsing some (hasRushingUrge some
        ((buys some Stock) and (sells some Stock) and
          ((buys some Stock) or (sells some Stock))))
7  ● isCommittedUsing some (usesFalse some Claim)
8  ● isCommittedUsing some SmallCorporation
9  ● resultsIn some (deflates some StockPrice)
10 ● resultsIn some EconomicLoss
11 ● SecuritiesFraud
```

Figure 26: Semantics of “Pump and Dump” Fraud

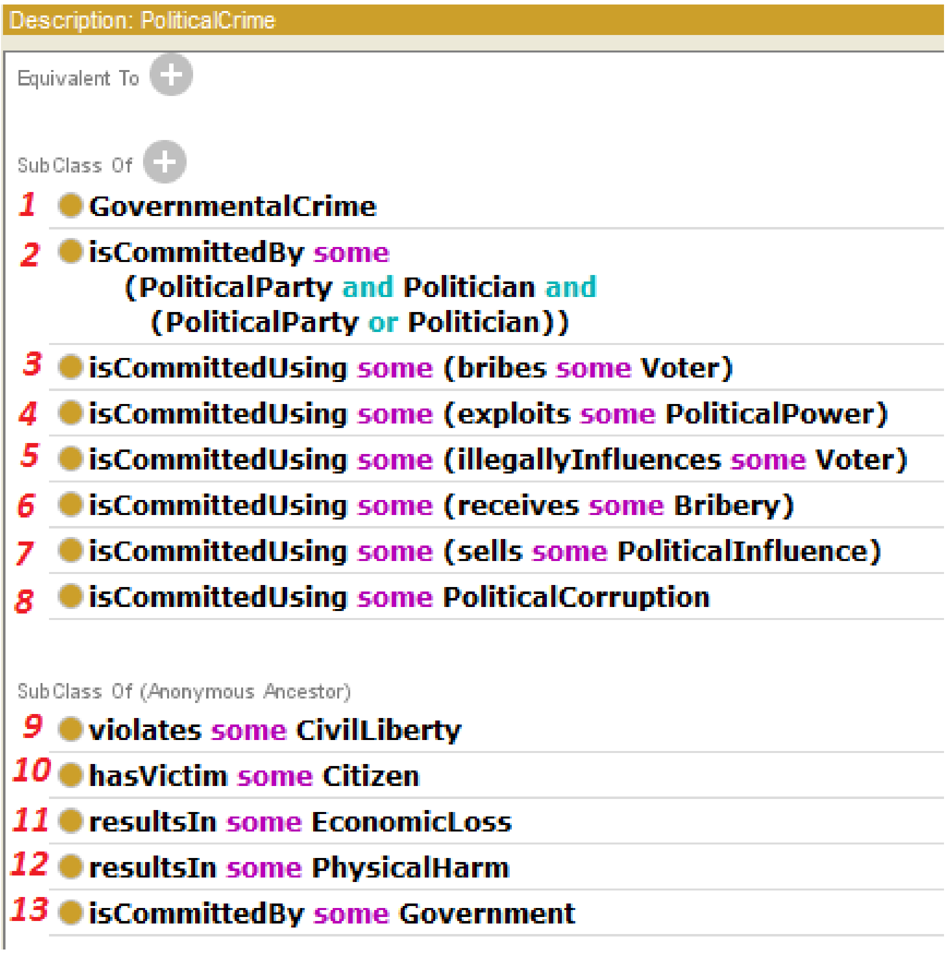
Line 1 of figure 26 simply says that the victim of this fraud is any investor who invests money in the pump and dump scheme [80] [81]. Line 2 says that this fraud is committed by inflating the stock price and touting the stock as described in 2.7.2 [80] [81]. Notice the use of “and” and “or” keywords as well as the repetition that this line involves. In DL, this line really means that pump and dump fraud can be committed by either inflating the stock price or touting the stock; or by performing these two actions in combination. This approach can be used whenever we want to say that either an event *A* is true or an event *B* is true or both events *A* and *B* are true.

Similarly line 3 says that pump and dump can be committed using the Internet, other social media or both [80] [81]. In addition to that, line 4 and 5 say that the financial health of corporation and the value of its stocks may be exaggerated during pump and dump fraud [81].

As mentioned in 2.7.2, investors may be prompted to buy or sell stock as quickly as possible [80] [81]. This is what line 6 says. Moreover, line 7 simply says that the fraudsters use false claims to deceive investors whereas line 8 says that fraudsters use small corporations to perpetrate pump and dump [80] [81].

Lines 9 and 10 say that this fraud eventually results in the deflation of stock price and the economic loss of investors – who are the actual victims as mentioned above [80] [81]. Finally line 11 simply says that pump and dump is a type – subclass – of securities fraud.

Figure 27 illustrates another example of semantics. This is associated with political crime which is described in 2.7.6.



```
Description: PoliticalCrime

Equivalent To

SubClass Of
1  GovernmentalCrime
2  isCommittedBy some
       (PoliticalParty and Politician and
         (PoliticalParty or Politician))
3  isCommittedUsing some (bribes some Voter)
4  isCommittedUsing some (exploits some PoliticalPower)
5  isCommittedUsing some (illegallyInfluences some Voter)
6  isCommittedUsing some (receives some Bribery)
7  isCommittedUsing some (sells some PoliticalInfluence)
8  isCommittedUsing some PoliticalCorruption

SubClass Of (Anonymous Ancestor)
9  violates some CivilLiberty
10 hasVictim some Citizen
11 resultsIn some EconomicLoss
12 resultsIn some PhysicalHarm
13 isCommittedBy some Government
```

Figure 27: Semantics of “Political Crime”

Line 1 simply says that political crime is a type – subclass – of governmental crime whereas line 2 that is committed by either politicians or political parties or both [3]. Lines 3-8 say the various different ways that political crime can be committed. More details on this can be found in 2.7.6.

It is worth mentioning that lines 9-13 are inherited from “GovernmentalCrime” class and simply point out the victims of governmental crime as well as its various consequences.

The rest of the semantics are included in the appendix chapter.

3.6 Conclusion

This chapter demonstrated the process of designing the various project’s deliverables. It started with the design of credit fraud detection ontology by presenting the various characteristics of techniques for detecting credit fraud. It then moved to the design of expert system by illustrating its class diagram which contained its important classes and operations. Finally, details about the design of generic fraud ontology were included.

The next chapter demonstrates the implementation of expert system as well as the usefulness of generic fraud ontology.

4 Implementation

This chapter contains information regarding the implementation of expert system. In addition to that, the usefulness of generic fraud ontology is demonstrated within this chapter.

4.1 Expert System Implementation

This section provides details about the implementation of expert system. It starts by discussing the technologies and methodology that have been followed during the implementation process. It then provides some screenshots to demonstrate the look and feel of expert system.

4.1.1 Java

The expert system has been implemented using Java programming language. Although the use of Java was compulsory for this project; it has many advantages over other programming languages. The main advantage is that it is platform-independent [107]. This means that Java programs can run in any platform without compatibility issues [107]. In addition to that, Java is “simpler and easier to learn” comparing to other programming languages since it provides automated facilities like “automatic memory allocation and garbage collection” [107]. Java is also object-oriented and therefore supports the construction of programs with modular structure [107].

The main disadvantage of Java programs, however, is that they are slower than programs written in other programming languages because they need Java Virtual Machine (JVM) in order to be executed [107]. They cannot be executed directly on the physical machine [107].

4.1.2 JavaServer Faces (JSF)

As already mentioned in 3.4, expert system was required to be a web application. This has been achieved by using JavaServer Faces framework.

According to [108], “JavaServer Faces (JSF) is a Java-based web application framework intended to simplify development integration of web-based user interfaces”. JSF simplifies the creation of web pages and provides separation of concerns [109]. In other words, it separates presentation logic from application logic by providing special-purpose UI components which can facilitate the construction of web applications [109]. JSF allows presentation logic and application logic to interact by reserving their distinction. More information on JSF can be found in [109].

4.1.3 OWL API

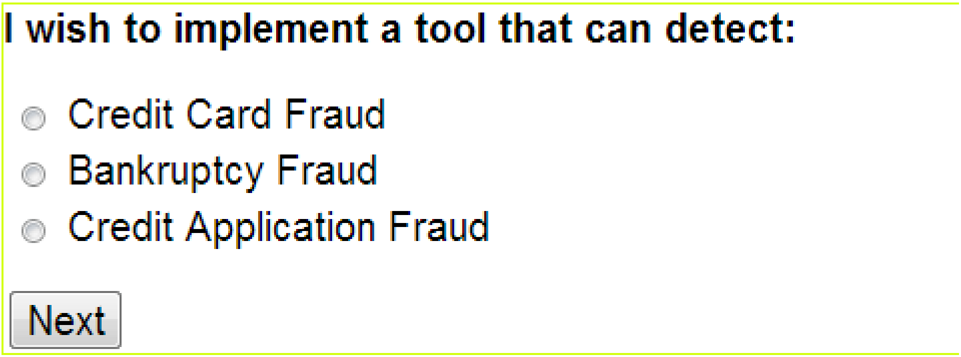
The OWL API is a Java application programming interface (API) which was created at the University of Manchester [106]. It facilitates the creation and manipulation of OWL ontologies and supports queries [106]. It maps query results into Java objects which can easily be used. In this project, the OWL API has been used to facilitate the interaction between expert system and credit fraud detection ontology.

4.1.4 Iterative and Incremental Development

An iterative and incremental development [110] has been applied during this project, in the following way: Expert system's development has been divided into three iterations; one for each type of credit fraud which it can support. During each iteration, the ontology was being expanded to include the characteristics of the new detection techniques. After that, the expert system was being updated to support the new credit fraud type and additional tests were being written to ensure that nothing has been broken and that the new functionality was working as expected. Therefore each iteration was resulting to a working version of the expert system. This was being demonstrated to the supervisor in order to receive feedback for potential improvements.

4.1.5 Selecting Fraud Type

Selecting type of credit fraud is the first thing that users need to do if they want to get advice from the expert system. Figure 28, illustrates the user interface of expert system for selecting fraud type.

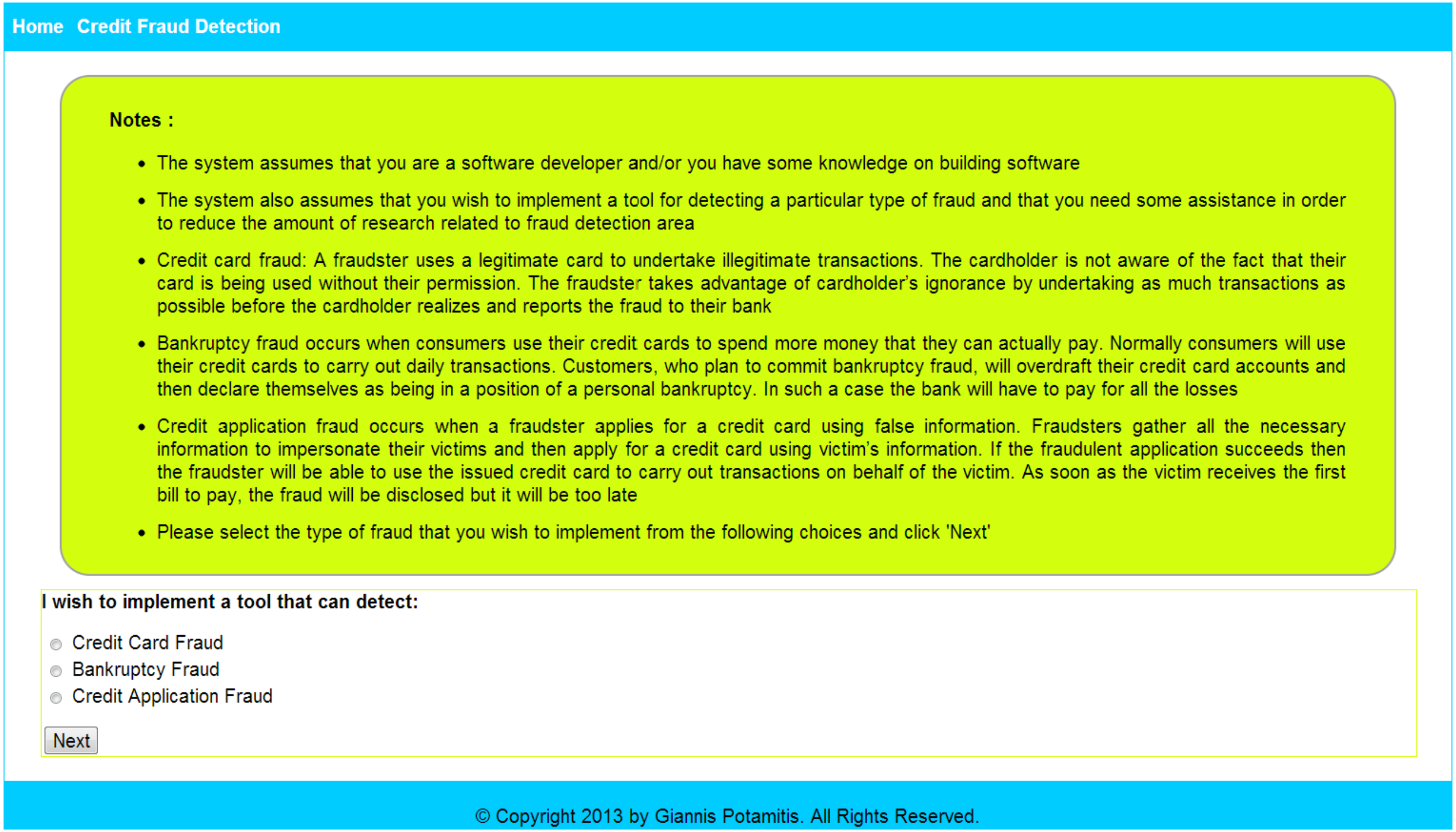


I wish to implement a tool that can detect:

○ Credit Card Fraud
○ Bankruptcy Fraud
○ Credit Application Fraud

Figure 28: Selecting Fraud Type

It is worth mentioning that each web page of expert system contains a section with some notes which help users to understand the contents of the page. Figure 29 illustrates the whole page for selecting fraud type including notes – these are found in the green section of the page.



Notes :

- The system assumes that you are a software developer and/or you have some knowledge on building software
- The system also assumes that you wish to implement a tool for detecting a particular type of fraud and that you need some assistance in order to reduce the amount of research related to fraud detection area
- Credit card fraud: A fraudster uses a legitimate card to undertake illegitimate transactions. The cardholder is not aware of the fact that their card is being used without their permission. The fraudster takes advantage of cardholder's ignorance by undertaking as much transactions as possible before the cardholder realizes and reports the fraud to their bank
- Bankruptcy fraud occurs when consumers use their credit cards to spend more money that they can actually pay. Normally consumers will use their credit cards to carry out daily transactions. Customers, who plan to commit bankruptcy fraud, will overdraft their credit card accounts and then declare themselves as being in a position of a personal bankruptcy. In such a case the bank will have to pay for all the losses
- Credit application fraud occurs when a fraudster applies for a credit card using false information. Fraudsters gather all the necessary information to impersonate their victims and then apply for a credit card using victim's information. If the fraudulent application succeeds then the fraudster will be able to use the issued credit card to carry out transactions on behalf of the victim. As soon as the victim receives the first bill to pay, the fraud will be disclosed but it will be too late
- Please select the type of fraud that you wish to implement from the following choices and click 'Next'

I wish to implement a tool that can detect:

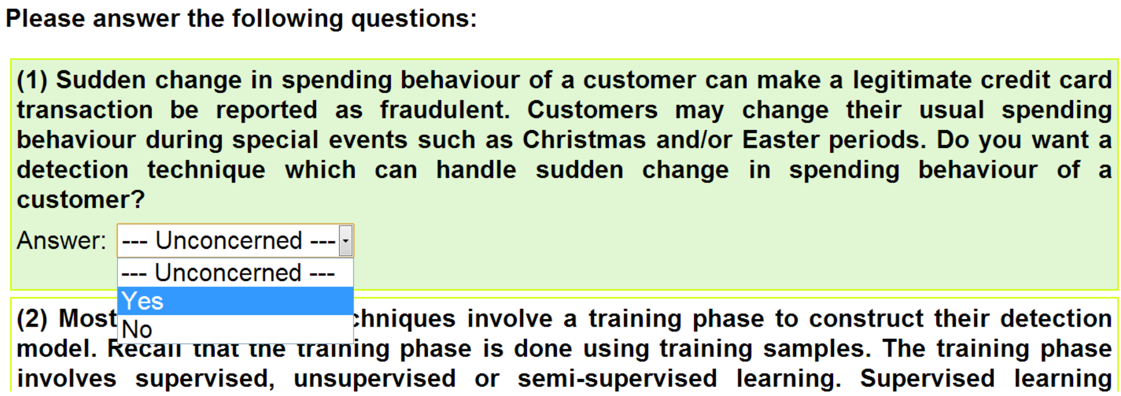
○ Credit Card Fraud
○ Bankruptcy Fraud
○ Credit Application Fraud

© Copyright 2013 by Giannis Potamitis. All Rights Reserved.

Figure 29: Big View of Selecting Fraud Type

4.1.6 Answering the Questions

After fraud type selection, users are prompted to answer a number of questions. Some of these questions are common to all different fraud types whereas some others are specific to the fraud type selected. Figure 30 illustrates the first question which is asked when users wish to detect credit card fraud.



Please answer the following questions:

(1) Sudden change in spending behaviour of a customer can make a legitimate credit card transaction be reported as fraudulent. Customers may change their usual spending behaviour during special events such as Christmas and/or Easter periods. Do you want a detection technique which can handle sudden change in spending behaviour of a customer?

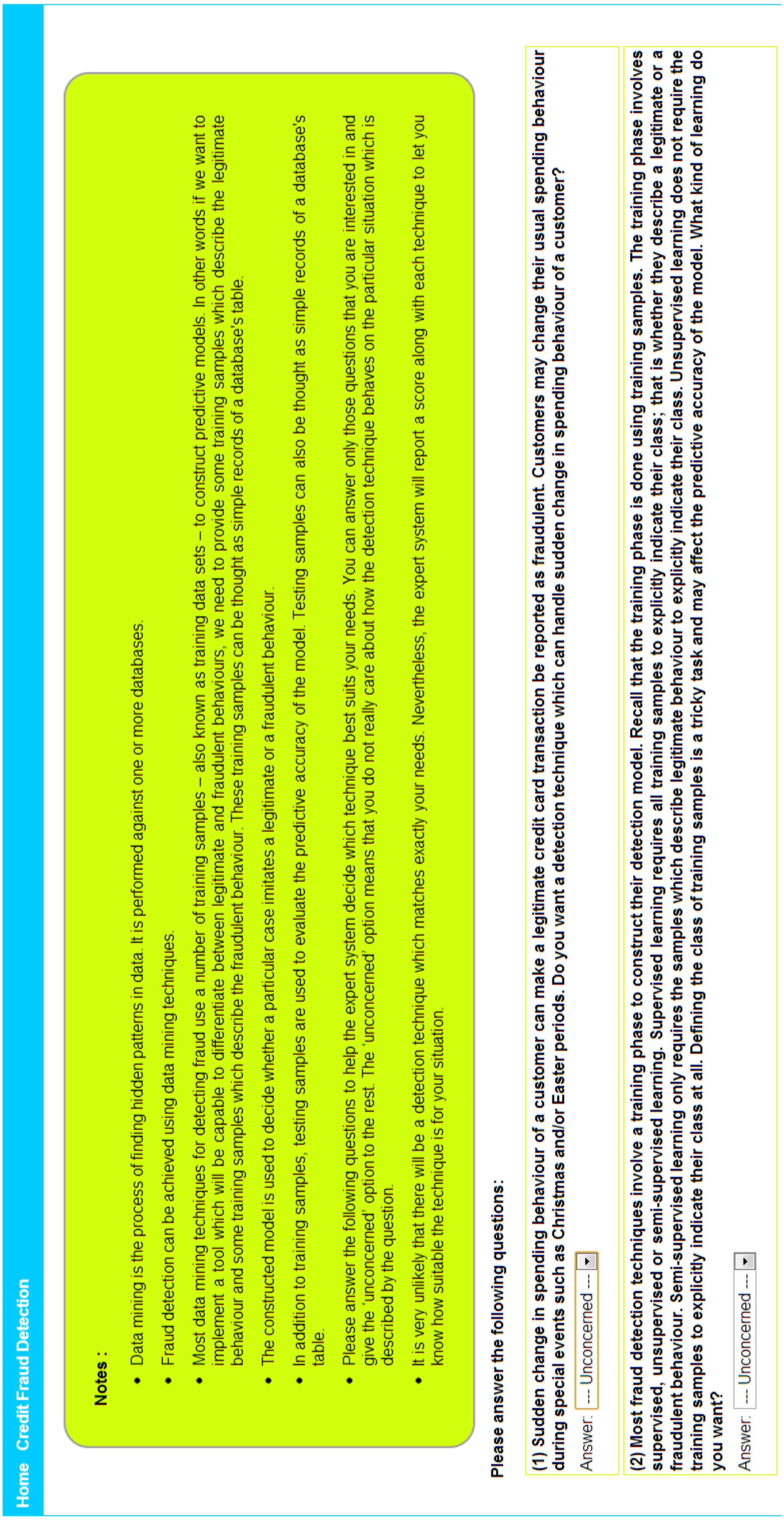
Answer: --- Unconcerned ---

(2) Most chniques involve a training phase to construct their detection model. Recall that the training phase is done using training samples. The training phase involves supervised, unsupervised or semi-supervised learning. Supervised learning

Figure 30: Answering the Questions

It is worth mentioning that all questions have the “Unconcerned” option as their default answer. The “Unconcerned” option means that the user does not really care how the detection techniques behave on that particular situation which is described by the question.

Figure 31 illustrates a bigger view of answering the questions. This includes the notes section as well. Here, the notes section contains more technical information; for instance what training samples are. It is believed that if users – who are software developers – read the notes section carefully, they should be able to understand the questions quite easily.



Notes :

- Data mining is the process of finding hidden patterns in data. It is performed against one or more databases.
- Fraud detection can be achieved using data mining techniques.
- Most data mining techniques for detecting fraud use a number of training samples – also known as training data sets – to construct predictive models. In other words if we want to implement a tool which will be capable to differentiate between legitimate and fraudulent behaviours, we need to provide some training samples which describe the legitimate behaviour and some training samples which describe the fraudulent behaviour. These training samples can be thought as simple records of a database's table.
- The constructed model is used to decide whether a particular case imitates a legitimate or a fraudulent behaviour.
- In addition to training samples, testing samples are used to evaluate the predictive accuracy of the model. Testing samples can also be thought as simple records of a database's table.
- Please answer the following questions to help the expert system decide which technique best suits your needs. You can answer only those questions that you are interested in and give the 'unconcerned' option to the rest. The 'unconcerned' option means that you do not really care about how the detection technique behaves on the particular situation which is described by the question.
- It is very unlikely that there will be a detection technique which matches exactly your needs. Nevertheless, the expert system will report a score along with each technique to let you know how suitable the technique is for your situation.

Please answer the following questions:

(1) Sudden change in spending behaviour of a customer can make a legitimate credit card transaction be reported as fraudulent. Customers may change their usual spending behaviour during special events such as Christmas and/or Easter periods. Do you want a detection technique which can handle sudden change in spending behaviour of a customer?

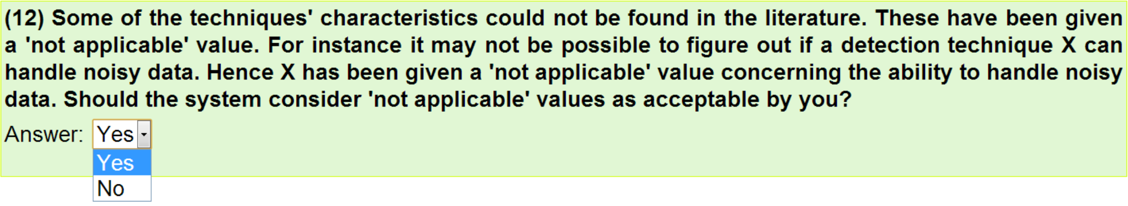
Answer: --- Unconcerned --- ▾

(2) Most fraud detection techniques involve a training phase to construct their detection model. Recall that the training phase is done using training samples. The training phase involves supervised, unsupervised or semi-supervised learning. Supervised learning requires all training samples to explicitly indicate their class; that is whether they describe a legitimate or a fraudulent behaviour. Semi-supervised learning only requires the samples which describe legitimate behaviour to explicitly indicate their class. Unsupervised learning does not require the training samples to explicitly indicate their class at all. Defining the class of training samples is a tricky task and may affect the predictive accuracy of the model. What kind of learning do you want?

Answer: --- Unconcerned --- ▾

Figure 31: Big View of Answering the Questions

It is worth mentioning that there are 12 questions in total for credit card fraud, 10 for bankruptcy fraud and 8 for credit application fraud. The last question is common to all fraud types. This is illustrated in figure 32 and it is related to not applicable values contained in the ontology.



(12) Some of the techniques' characteristics could not be found in the literature. These have been given a 'not applicable' value. For instance it may not be possible to figure out if a detection technique X can handle noisy data. Hence X has been given a 'not applicable' value concerning the ability to handle noisy data. Should the system consider 'not applicable' values as acceptable by you?

Answer: Yes

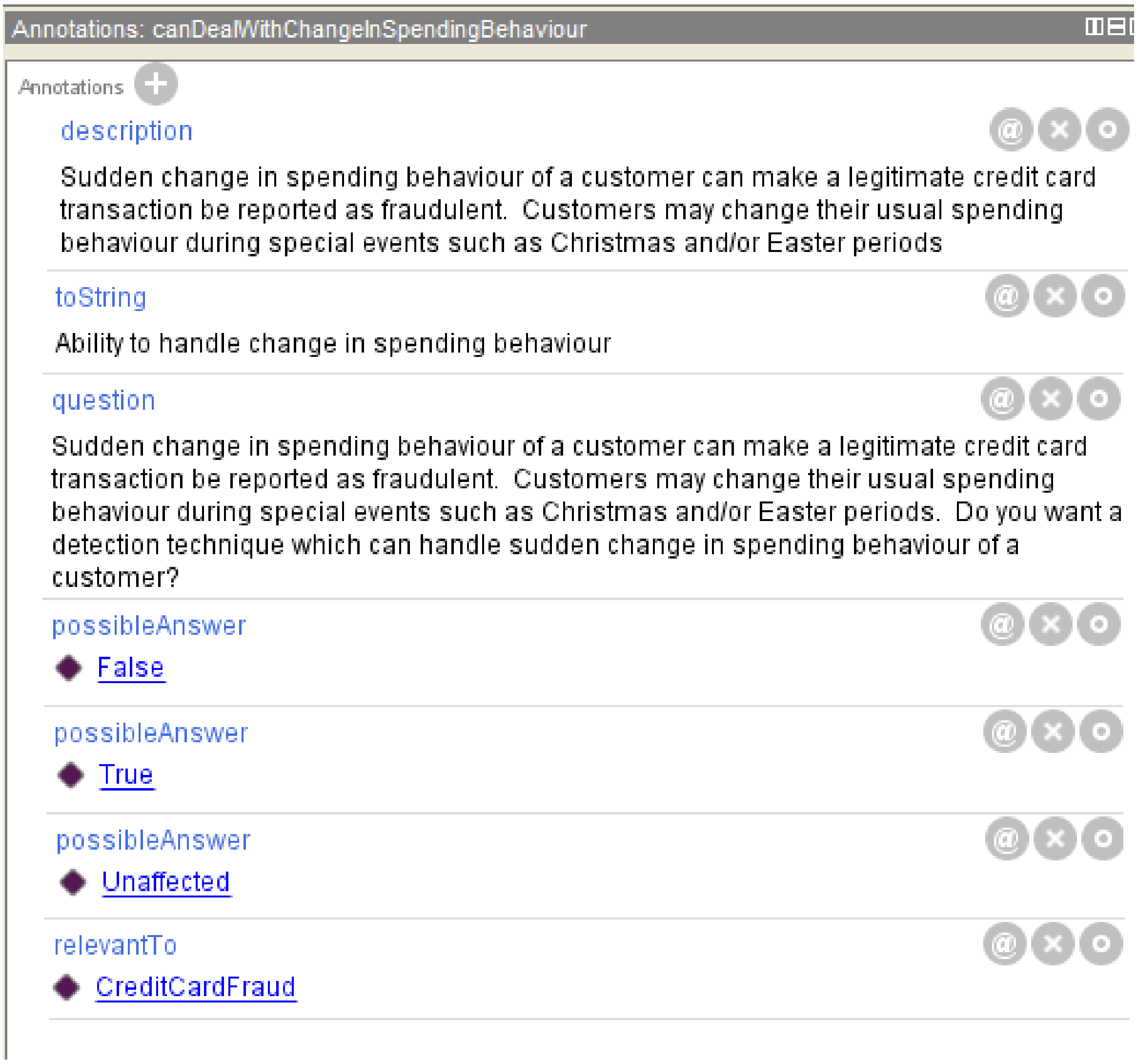
Figure 32: “Not Applicable Value” Question

The answer to this question affects the way in which the matching scores of detection techniques are calculated. In order to better understand this, consider the following example: Recall that “LSTM” mentioned in 2.3.1 and 3.2.1 has a “not applicable value” concerning the ability to detect credit card fraud in real-time. This is shown in table 5. Notice that there is a question which asks users whether the detection technique should be able to detect fraud in real-time. This is associated with real-time property of table 5. Let’s call this question *X*. If the user answers “Yes” to question *X* as well as to “not applicable value” question then the score of “LSTM” will be increased. This is because the user wishes to consider “not applicable” values as acceptable and since “LSTM” has a “not applicable” value on real-time property then the system assumes that “LSTM” satisfies the user answer at question *X*. On the other hand, if the user answers “No” to “not applicable value” question and gives any answer to question *X* then the score of “LSTM” will not be increased because the user does not wish to consider “not applicable” values as acceptable. Therefore “LSTM” can never satisfy the user answer at question *X*; no matter what this answer is.

It may be easier to understand how “not applicable value” question works if the reader refers to subsection 4.1.9 which provides the pseudo code for calculating scores.

4.1.7 Using Ontology Annotations

All the questions and their possible answers are encapsulated inside credit fraud detection ontology. This is done using annotations. Annotations can be applied to all elements of an ontology including properties, classes and individuals. Figure 33 illustrates the annotations of "canDealWithChangeInSpendingBehaviour" property.



Annotations: canDealWithChangeInSpendingBehaviour

Annotations

description

Sudden change in spending behaviour of a customer can make a legitimate credit card transaction be reported as fraudulent. Customers may change their usual spending behaviour during special events such as Christmas and/or Easter periods

toString

Ability to handle change in spending behaviour

question

Sudden change in spending behaviour of a customer can make a legitimate credit card transaction be reported as fraudulent. Customers may change their usual spending behaviour during special events such as Christmas and/or Easter periods. Do you want a detection technique which can handle sudden change in spending behaviour of a customer?

possibleAnswer

◆ False

possibleAnswer

◆ True

possibleAnswer

◆ Unaffected

relevantTo

◆ CreditCardFraud

Figure 33: Annotations Example

The first annotation is called "description" and is used to explain the purpose of this property. More information about how the "description" annotation is used in the expert system can be found in 4.1.10.

The second annotation called "toString" has the same purpose as the "toString()" method of Java. More precisely, this annotation specifies how the property should ever be shown in the screen. In other words, if we want to print the "canDealWithChangeInSpendingBehaviour" property on the screen then we will print the text of "toString" annotation. Clearly, this is more user-friendly than printing the property exactly as appears in the ontology. It is worth mentioning that the “toString” annotation is used to all ontology’s elements which need to be presented to the user including classes and individuals.

The next annotation called “question” contains the exact question as this is asked by the expert system. Notice that every question which is asked by the expert system is directly associated with a property in the ontology using the “question” annotation. However, not all properties in the ontology have a “question” annotation since not all properties should be asked by the expert system. For instance it would not make any sense if the “hasAccuracyLevel” property – shown in figure 10 – was linked to a question. This is because if you ask a user a question like “what accuracy level should the detection technique have”; the user will probably answer “the maximum level possible”. Therefore it is vital to ask questions which reveal the real needs of users and not questions whose answers are obvious. The writer of this report believes that only important questions are ever asked by the expert system.

The next three annotations contain the possible answers that the question has. Notice that “No”, “Yes” and “Unconcerned” are the “toString” annotations of “False”, “True” and “Unaffected” respectively. Notice also that figure 30 shows the “toString” annotation of these possible answers to the user.

Finally the “relevantTo” annotation specifies the type of credit fraud that this property is associated with. Whenever the expert system wishes to retrieve all the questions for a particular type of credit fraud, say *X*, it will retrieve all properties whose “relevantTo” annotation has value *X*. It will then map the values of “question” and “possibleAnswer” annotations into Java objects.

Creating Question and Answer Objects

Recall “Question” and “Answer” classes illustrated in figure 17. Objects of these classes need to be created and sent to presentation layer which will ask the questions to the user.

It is important for the reader to understand the variable values of these objects. Consider the “canDealWithChangeInSpendingBehaviour” property as an example. Table 12 shows the variable values of “Question” and “Answer” objects which are created to represent this property.

<table>
<tr><th colspan="2">Question Object</th></tr>
<tr><th>Variable</th><th>Value</th></tr>
<tr><td>id</td><td>“canDealWithChangeInSpendingBehaviour”</td></tr>
<tr><td>description</td><td>“Sudden change in spending behaviour...”</td></tr>
<tr><td>type</td><td>“object” – since it is an object property</td></tr>
<tr><td>possibleAnswers</td><td><b>List of Answer Objects</b>
<table>
<tr><td rowspan="3"><b>1</b></td><th>Variable</th><th>Value</th></tr>
<tr><td>id</td><td>“True”</td></tr>
<tr><td>description</td><td>“Yes”</td></tr>
</table>
<table>
<tr><td rowspan="2"><b>2</b></td><td>id</td><td>“False”</td></tr>
<tr><td>description</td><td>“No”</td></tr>
</table>
<table>
<tr><td rowspan="2"><b>3</b></td><td>id</td><td>“Unaffected”</td></tr>
<tr><td>description</td><td>“Unconcerned”</td></tr>
</table>
</td></tr>
</table>

Table 12: Variable Values of Question and Answer objects

Notice that all the “id” variables contain the element names exactly as appear in the ontology. In addition to that, the “description” variable of question object is obtained from “question” annotation whereas the “description” variable of answer objects is obtained from “toString” annotation.

4.1.8 Converting User Answers into DL Queries

Converting user answers into DL queries can easily be done due to the way that questions and answers are represented by the expert system – as discussed in 4.1.7. In order to understand the way in which user answers are converted into DL queries consider the following example. Let’s assume that a user answered “Yes” in the question associated with “canDealWithChangeInSpendingBehaviour” property. This answer needs to be mapped into DL query in order to retrieve, from ontology, all these techniques which can deal with change in spending behaviour. The expert system applies the following simple algorithm in order to convert this answer into DL query:

1. get the id of question

2. add the word “some”
3. and then add the id of answer.

Thus the above user answer will be converted to “canDealWithChangeInSpendingBehaviour some True” which is a valid DL query.

4.1.9 Calculating Scores

The pseudo code for calculating the matching scores of detection techniques – based on user answers – is shown in figure 34.

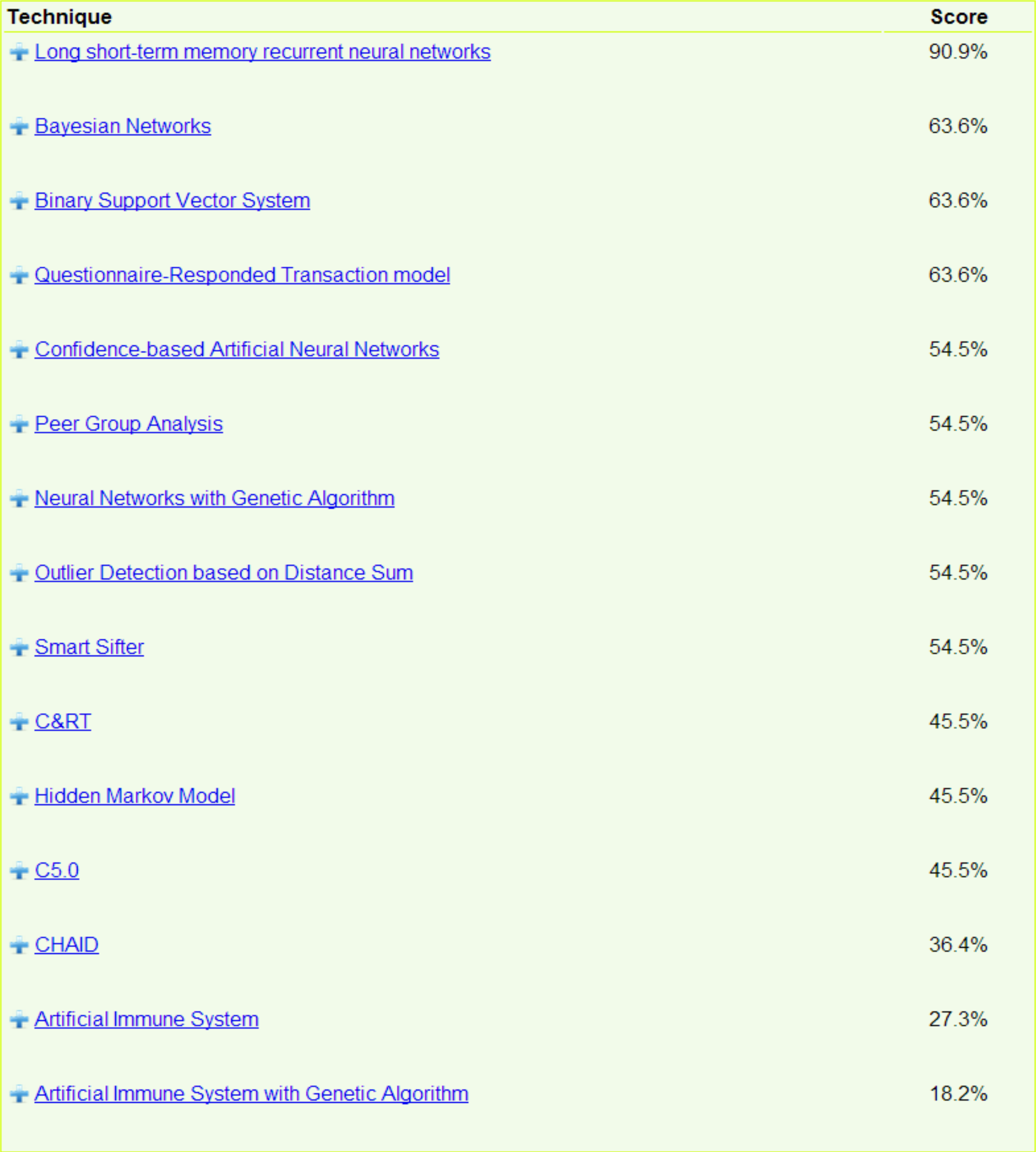
```
getScore (fraudType: String, userAnswers: Map of Question and Answer objects):

    1. Retrieve all individuals, from the ontology, which can detect that
       particular fraudType
    2. For each individual initialize its matching score to 0
    3. Find the answer to “not applicable value” question from
       userAnswers
    4. For each user answer A:
        a. If A is “Unconcerned” then increment all scores by 1 -
           since all individuals can satisfy “Unconcerned” answers -
           and continue with the next user answer
        b. Else convert A into DL query as explained in 4.1.8 and go to
           step 4.c
        c. If the answer to “not applicable value” question is “Yes”
           then expand DL query by adding “or ([question_id] some
           NotApplicable)”
        d. Expand DL query by adding “or ([question_id] some
           Unaffected)”
        e. Ask DL query to credit fraud detection ontology
        f. For each individual which was returned after DL query was
           asked, increment its matching score by 1
    5. Sort the list of all individuals based on their matching scores
    6. Return all individuals along with their matching scores
```

Figure 34: Pseudo Code for Score Calculation

As line 4.a suggests, in case the user answer is “Unconcerned” then the matching scores of all individuals will simply be incremented. This is because the user does not care how the technique behaves in the particular situation described by the question and hence all techniques should be considered as valid.

Line 4.c makes sure that if the user has given a positive answer to “not applicable value” question then individuals with “not applicable” values are also returned. In addition to that, line 4.d is also important because as shown in 3.1 and 3.2 some techniques’ characteristics take “Don’t Care” value. These techniques should always be considered as satisfactory no matter what the user answer is. Therefore the DL query needs to be expanded in order to return techniques with “Don’t care” values too.

Figure 35 illustrates an example of matching scores as these are presented by the expert system.



Technique	Score
Long short-term memory recurrent neural networks	90.9%
Bayesian Networks	63.6%
Binary Support Vector System	63.6%
Questionnaire-Responded Transaction model	63.6%
Confidence-based Artificial Neural Networks	54.5%
Peer Group Analysis	54.5%
Neural Networks with Genetic Algorithm	54.5%
Outlier Detection based on Distance Sum	54.5%
Smart Sifter	54.5%
C&RT	45.5%
Hidden Markov Model	45.5%
C5.0	45.5%
CHAID	36.4%
Artificial Immune System	27.3%
Artificial Immune System with Genetic Algorithm	18.2%

Figure 35: Example of Matching Scores

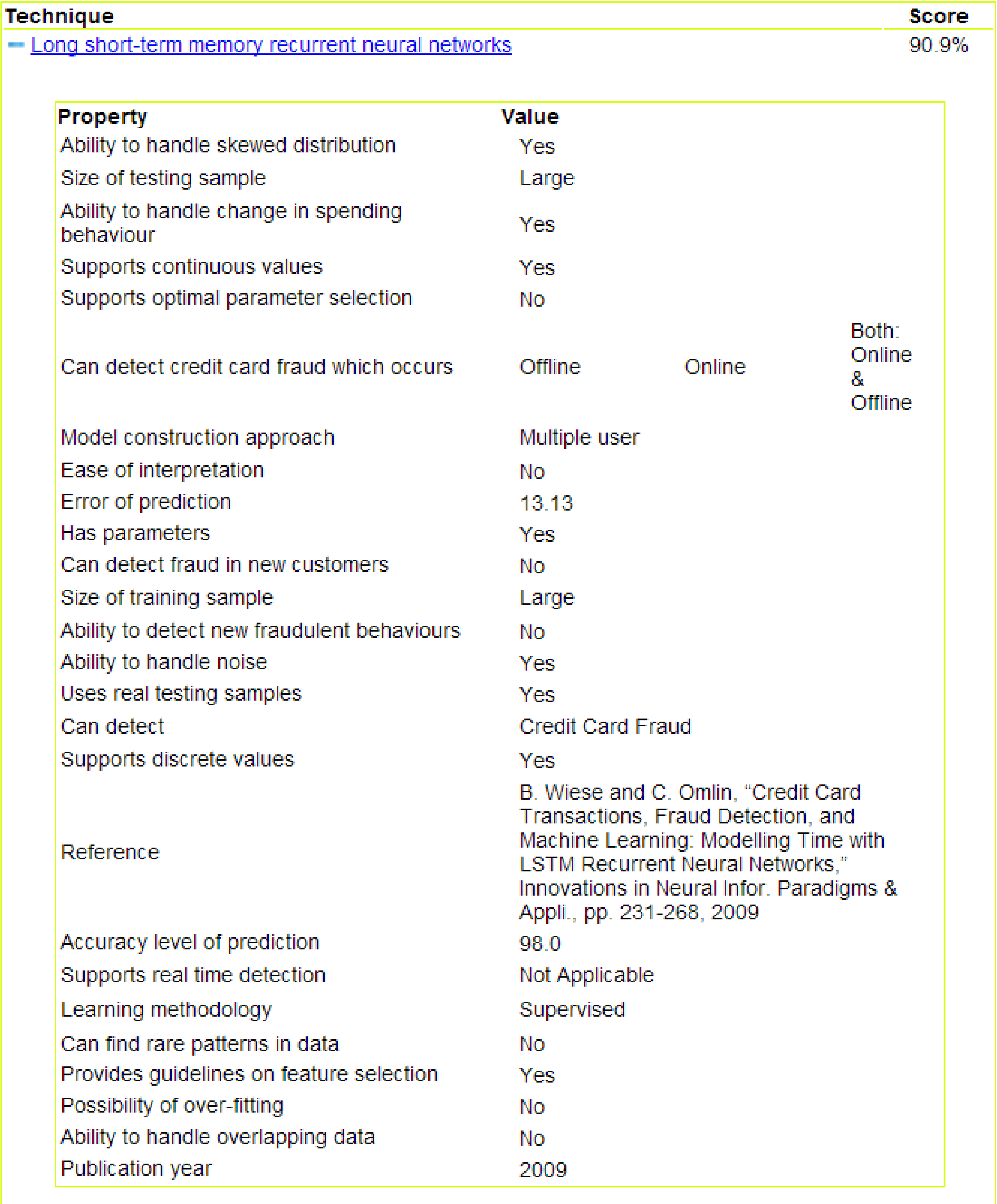
According to figure 35, the best matching score is 90.9% and it is achieved by “Long Short-term Memory Recurrent Neural Networks” (LSTM) implementation of ANNs. This means that “LSTM” satisfies 90.9% of user answers.

Notice that the user can inspect the characteristics of various detection techniques by clicking on their name. More details on this can be found in the next subsection – 4.1.10.

4.1.10 Showing Characteristics

As mentioned in 4.1.9 when a user clicks on a technique, its characteristics are presented. Figure 36 illustrates the characteristics when clicking on “LSTM”.

These are the characteristics as described in 3.2.1. The expert system has obtained these characteristics from credit fraud detection ontology.



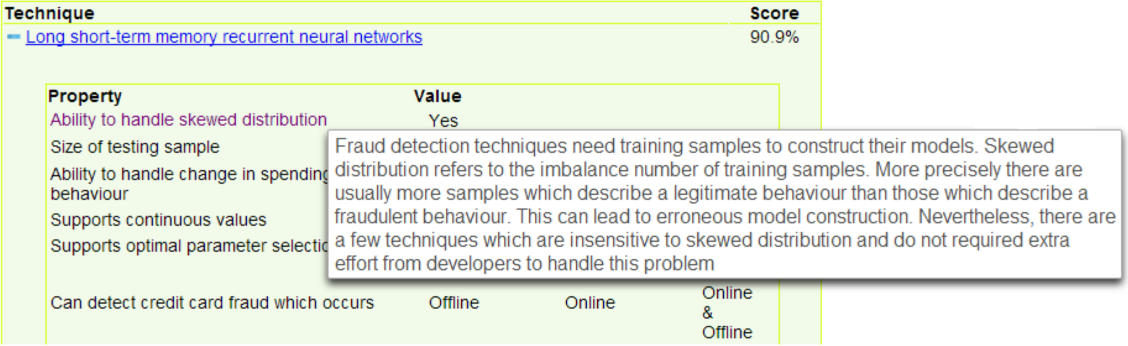
Technique	Score
− Long short-term memory recurrent neural networks	90.9%

Property	Value		
Ability to handle skewed distribution	Yes		
Size of testing sample	Large		
Ability to handle change in spending behaviour	Yes		
Supports continuous values	Yes		
Supports optimal parameter selection	No		
Can detect credit card fraud which occurs	Offline	Online	Both: Online & Offline
Model construction approach	Multiple user		
Ease of interpretation	No		
Error of prediction	13.13		
Has parameters	Yes		
Can detect fraud in new customers	No		
Size of training sample	Large		
Ability to detect new fraudulent behaviours	No		
Ability to handle noise	Yes		
Uses real testing samples	Yes		
Can detect	Credit Card Fraud		
Supports discrete values	Yes		
Reference	B. Wiese and C. Omlin, “Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks,” Innovations in Neural Infor. Paradigms & Appli., pp. 231-268, 2009		
Accuracy level of prediction	98.0		
Supports real time detection	Not Applicable		
Learning methodology	Supervised		
Can find rare patterns in data	No		
Provides guidelines on feature selection	Yes		
Possibility of over-fitting	No		
Ability to handle overlapping data	No		
Publication year	2009		

Figure 36: Characteristics of LSTM

Notice, from figure 36, that a reference to the literature article – which describes “LSTM” – is also included in the set of characteristics. Therefore, the users can refer to this article to learn more about the detection technique. It is worth mentioning that references are stored in the ontology by attaching a “reference” annotation to each detection technique.

Helpful messages are shown whenever the user points to a property or value. This can help users to easier understand the characteristics of various techniques. Figures 37 and 38 illustrate two examples of helpful messages.



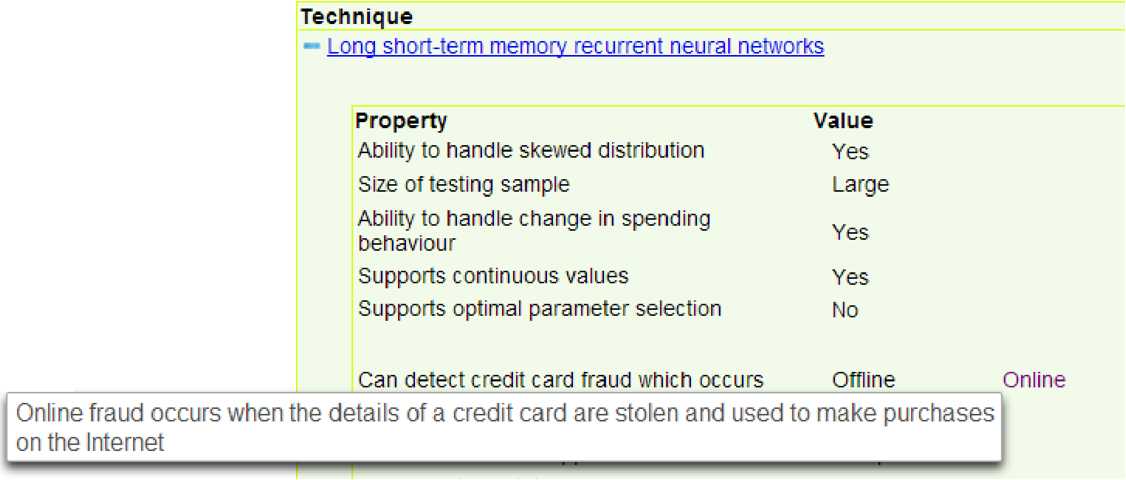
Technique Score

Long short-term memory recurrent neural networks 90.9%

Property	Value
Ability to handle skewed distribution	Yes
Size of testing sample	
Ability to handle change in spending behaviour	
Supports continuous values	
Supports optimal parameter selectio	
Can detect credit card fraud which occurs	Offline Online Online & Offline

Fraud detection techniques need training samples to construct their models. Skewed distribution refers to the imbalance number of training samples. More precisely there are usually more samples which describe a legitimate behaviour than those which describe a fraudulent behaviour. This can lead to erroneous model construction. Nevertheless, there are a few techniques which are insensitive to skewed distribution and do not required extra effort from developers to handle this problem

Figure 37: Helpful Messages (A)



Technique

Long short-term memory recurrent neural networks

Property	Value
Ability to handle skewed distribution	Yes
Size of testing sample	Large
Ability to handle change in spending behaviour	Yes
Supports continuous values	Yes
Supports optimal parameter selection	No
Can detect credit card fraud which occurs	Offline Online

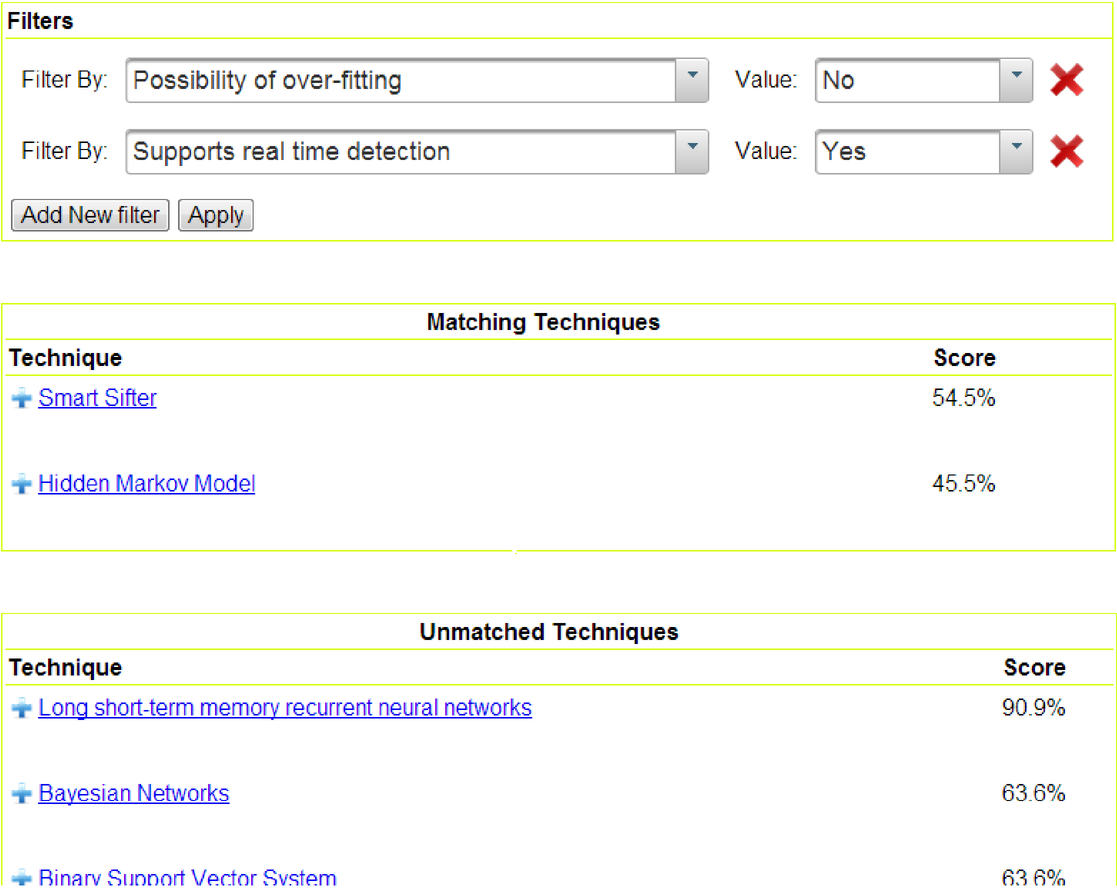
Online fraud occurs when the details of a credit card are stolen and used to make purchases on the Internet

Figure 38: Helpful Messages (B)

It is worth mentioning that the helpful messages are stored in the ontology using the “description” annotation shown in 4.1.7.

4.1.11 Filtering Facility

In addition to matching scores, the expert system provides a filtering facility which allows users to perform more advanced search on the characteristics of detection techniques. Figure 39 illustrates a filtering example.



Filters

Filter By: Possibility of over-fitting Value: No ✖

Filter By: Supports real time detection Value: Yes ✖

Matching Techniques

Technique	Score
Smart Sifter	54.5%
Hidden Markov Model	45.5%

Unmatched Techniques

Technique	Score
Long short-term memory recurrent neural networks	90.9%
Bayesian Networks	63.6%
Binary Support Vector System	63.6%

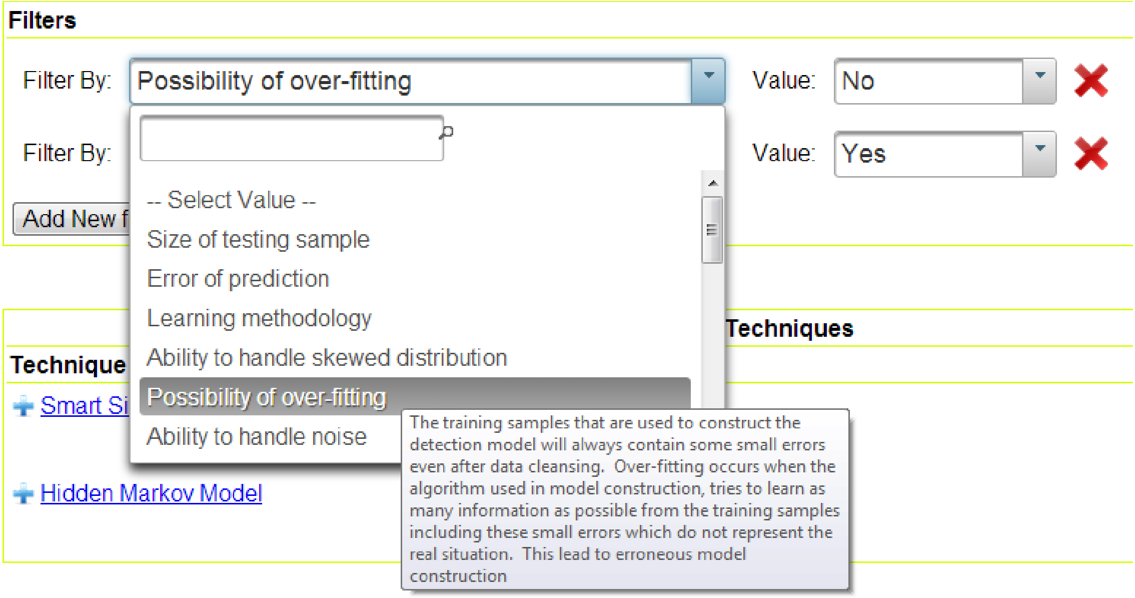
Figure 39: Filtering Example

The users can specify as many filters as they want. The techniques are divided into two categories. The category of techniques which satisfy filters – matching techniques – and the category of techniques which do not satisfy filters – unmatched techniques. Notice that “LSTM” is now placed in the category of unmatched techniques even if it has the highest score.

The following example demonstrates the usefulness of filtering facility. Let’s assume that a user wants a detection technique which can handle noisy data. The user answers all the questions of expert system by making sure that he or she answers “Yes” in the “ability to handle noisy data” question. The expert system calculates the matching scores based on user answers and reports these to the user. Let’s assume that the technique which has the highest score is “BSVS” because it satisfies *most* of user answers. Nevertheless, “BSVS” cannot handle noisy data – recall table 4. In this case, the user can employ the filtering facility to identity the techniques which can handle noisy data. Notice that these techniques will have lower score than “BSVS” but since they satisfy the most important user need, they can be considered as being more suitable to this particular user.

Recall that not all properties included in the ontology are associated with a question. The user can filter by all the available properties including those which are not associated with any question. It is therefore believed that the filtering facility allows users to make a wiser choice as to which detection technique is most appropriate for them.

It is worth mentioning that the helpful messages described in 4.1.10 are also presented to the user while using the filtering facility. Figure 40 illustrates this.



Filters

Filter By: Possibility of over-fitting Value: No

Filter By: Value: Yes

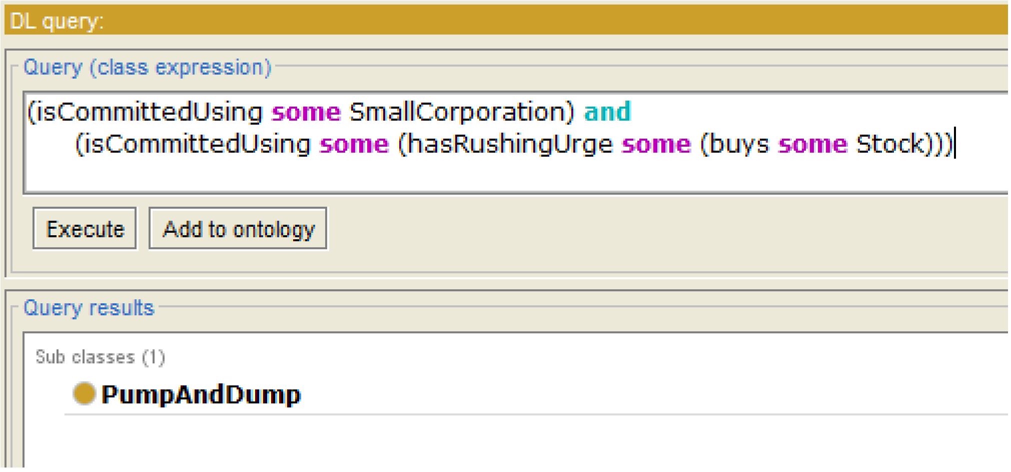
The training samples that are used to construct the detection model will always contain some small errors even after data cleansing. Over-fitting occurs when the algorithm used in model construction, tries to learn as many information as possible from the training samples including these small errors which do not represent the real situation. This lead to erroneous model construction

Figure 40: Helpful Messages while Filtering

4.2 Usefulness of Generic Fraud Ontology

In order to demonstrate the usefulness of generic fraud ontology, consider the following example: Let’s assume a system which is capable to inform users for the type of fraud which they have been victimized. The user provides some input information to the system concerning actions that he or she has observed. The system converts this information into DL query and sends it to generic fraud ontology which replies with the fraud type which matches that query. After that, the system reports this fraud type and its characteristics to the user.

Let's assume that Alice received a phone call from Bob who prompted her to immediately buy some stocks from a small company *C*. Bob claimed that the financial health of *C* was very promising and that this was a big economic opportunity for Alice. Alice, who had some concerns, used the system described above to figure out if there is any fraud related to the above events. The system converted the information provided by Alice into DL query and sent it to generic fraud ontology. Figure 41 illustrates a possible DL query and the returned result. According to figure 41, the above example matches "pump and dump" fraud.



```
(isCommittedUsing some SmallCorporation) and
    (isCommittedUsing some (hasRushingUrge some (buys some Stock)))
```

● PumpAndDump

Figure 41: Querying Generic Fraud Ontology

Notice that the above example assumes that the system has the appropriate user interface which allows users to provide the information they want. It also assumes that the system is capable to convert this information into DL queries. Using ontology annotations in a similar way as described in 4.1.7 may be useful for the construction of such a system.

4.3 Conclusion

This chapter demonstrated the implementation of expert system. In addition to that, an example to demonstrate the usefulness of generic fraud ontology was included in this chapter.

The next chapter provides an example of how the expert system has been tested.

5 Testing

This chapter provides an indication about the way in which the expert system has been tested.

5.1 Unit Testing

Unit testing has been employed to test the functionality of expert system. This has been achieved by writing tests for all individual units – namely the Java methods – during each iteration. The tests were being rerun during subsequent iterations to ensure that nothing has been broken.

In order to ensure exhaustive test coverage on long units which had complicated structure – for instance a lot of branches; divide and conquer approach has been employed. More precisely, all the possible test cases were being identified and then tests were being written to cover these cases. A good example on this is shown in the following subsection – that is 5.1.1.

5.1.1 Testing Score Calculation

This subsection demonstrates the way in which "getScore" method – described in 4.1.9 has been tested. This method contains a number of branches and hence testing is more difficult. However, possible test cases have been derived to ensure exhaustive test coverage of this method. These are shown in table 13.

A/A	Test Case Description
<u>Testing "Unconcerned" answers as following (cases 1-7):</u>	
1	• All answers to questions are "Unconcerned" • Expected Result: The scores of all techniques are 100%
For each technique *t*:	
2	• Achieve full match with no "Unconcerned" answers • Expected Result: Score for *t* is 100%
3	• Achieve full match with one "Unconcerned" answer • Expected Result: Score for *t* is 100%
4	• Achieve full match with many "Unconcerned" answers • Expected Result: Score for *t* is 100%
5	• Achieve partial match with no "Unconcerned" answers by:

<table>
<tr><td></td><td>o Calculating score manually (expected score)
o Calling “getScore” to get actual score
• Expected Result: expected score = actual score</td></tr>
<tr><td><b>6</b></td><td>• Achieve partial match with one “Unconcerned” answer by:
o Calculating score manually (expected score)
o Calling “getScore” to get actual score
• Expected Result: expected score = actual score</td></tr>
<tr><td><b>7</b></td><td>• Achieve partial match with many “Unconcerned” answers by:
o Calculating score manually (expected score)
o Calling “getScore” to get actual score
• Expected Result: expected score = actual score</td></tr>
<tr><td colspan="2"><b><u>Testing “not applicable” values as following (cases 8-10):</u></b>
<b>For each technique <i>t</i> which has a “not applicable” value in any property:</b></td></tr>
<tr><td><b>8</b></td><td>• Give answer “Yes” to “not applicable value” question and give random answers to the rest of questions
• Call “getScore” to get the score for <i>t</i> (score one)
• Then give answer “No” to “not applicable value” question and give the same set of random answers to other questions.
• Call “getScore” to get the score for <i>t</i> again (score two)
• Expected Result: score one != score two</td></tr>
<tr><td><b>9</b></td><td>• Give answer “Yes” to “not applicable value” question and give random answers to the rest of questions
• Calculate the score manually while giving the random answers (expected score)
• Call getScore method to get the actual score
• Expected Result: actual score = expected score</td></tr>
<tr><td><b>10</b></td><td>• Give answer “No” to “not applicable value” question and give random answers to the rest of questions
• Calculate the score manually while giving the random answers (expected score)
• Call “getScore” method to get actual score
• Expected Result: actual score = expected score</td></tr>
</table>

Testing “Unaffected” values as following (case 11): **For each technique *t*, which has an “Unaffected” – that is “Don’t Care” – value in any property *p*:**	
11	• Give any answer to the question associated with *p* and random answers to the rest of questions • Call “getScore” method to get the score for *t* (score one) • Then, give a different answer to the question associated with *p* and the same set of random answers to the rest of questions • Call “getScore” method to get the score for *t* again (score two) • Expected Result: score one = score two

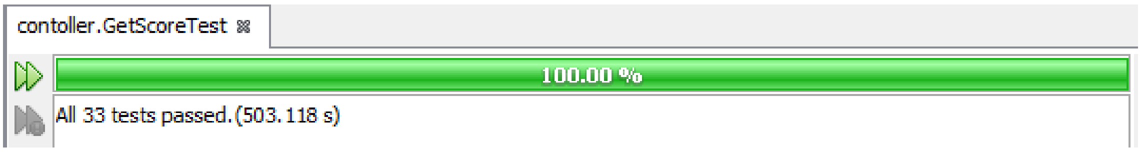
Table 13: Test Cases for Score Calculation

As table 13 suggests, there are eleven test cases for “getScore” method. However, in order to test this method exhaustively, these test cases were repeated for each type of credit fraud. Therefore 33 tests have actually been run since there are three different types of credit fraud. It is worth noting that all test cases – except 1 – are repeated for each possible detection technique. This is done by using loops inside each test case.

Notice that the answer for “not applicable value” question in cases 1-7 and 11 can be any – either “Yes” or “No”. Notice also that in order to achieve full or partial match – as required by test cases 2-7; the characteristics of each technique were being retrieved from the ontology and then answers to questions based on these characteristics were being given. For full match all the answers must have been satisfied by the current technique *t* whereas for partial match only some of the answers must have been satisfied by *t*. In addition to that, when a test case uses the word “many” for the “Unconcerned” answers, it really means that it can be tested with two or more “Unconcerned” answers.

Manual score calculation means that the test needed to keep track of the technique’s score while answering each question. This could simply be done by increasing the score counter whenever the given answer could be satisfied by the current technique *t*.

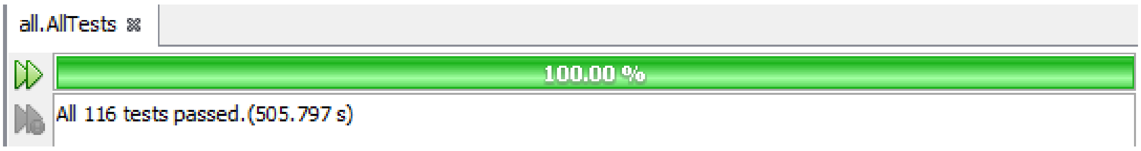
Figure 42 – which is taken from NetBeans – illustrates that all the tests for “getScore” method have eventually passed.



All 33 tests passed.(503.118 s)

Figure 42: Tests for Calculating Score Passed

Figure 43 illustrates that all the tests for the expert system have eventually passed.



All 116 tests passed.(505.797 s)

Figure 43: All Tests Passed

5.2 Conclusion

This chapter provided an indication regarding the way in which the expert system has been tested.

The next chapter evaluates the expert system.

6 Evaluation and Critical Analysis

This chapter demonstrates the evaluation of expert system using an online questionnaire.

6.1 Questionnaire

An online questionnaire has been created via Survey Monkey [111] to evaluate the usefulness of expert system. The questionnaire contained a link to the expert system which has been hosted by an online server. It is worth mentioning that the questionnaire has been approved by School Ethics Committee [112].

More information about the questionnaire is included in the next subsections.

6.1.1 Questions

Table 14 demonstrates the questions which were included in the questionnaire. All the questions – except question 10 – were multiple choice questions. Table 15 contains the possible options of these questions. For question 10, people could type in text.

A/A	Question	Options ID
1	“Are you a software developer and/or do you have some knowledge on building software?”	o1
2	“Given that the area of fraud detection involves a high amount of complexity; the system tries to minimize this complexity by asking concise and precise questions which contain just enough information to help you understand what they are talking about and avoid too much details”	o2
3	“The notes, questions and reported results of the system reduce the amount of research that a software developer needs to undertake in order to build a credit fraud detection tool”	o2
4	“The system is useful for software developers who wish to build a credit fraud detection tool”	o2
5	“If you wanted to build a tool for detecting any of the three	o3

	fraud types that are supported by the system; how likely would it be to trust this system”	
6	“The system is user-friendly and easy to use”	o2
7	“It is easy to understand both the questions and the information which are provided by the system even if you don’t have any prior knowledge on fraud detection techniques”	o2
8	“The reported scores are a good indication as to which technique is most suitable for the current situation”	o2
9	“The filtering facility complements the reported scores by allowing the users to perform more advanced search on the properties of each detection technique. This makes the users more capable to choose the technique which best suits their needs”	o2
10	“If you have any comments and/or you would like to suggest any improvements please do not hesitate to report them here”	-

Table 14: Evaluation Questions

Options ID	Possible Question Options
o1	a. Yes b. No
o2	a. Strongly Agree b. Agree c. Neither Agree nor Disagree d. Disagree e. Strongly Disagree
o3	a. Extremely Likely b. Likely c. Neither Likely nor Unlikely d. Unlikely e. Extremely Unlikely

Table 15: Possible Question Options

Notice that table 14 and 15 are linked using options' ID. For instance the possible options of question 1 are "Yes" and "No" since its option ID is "o1".

6.1.2 Participants

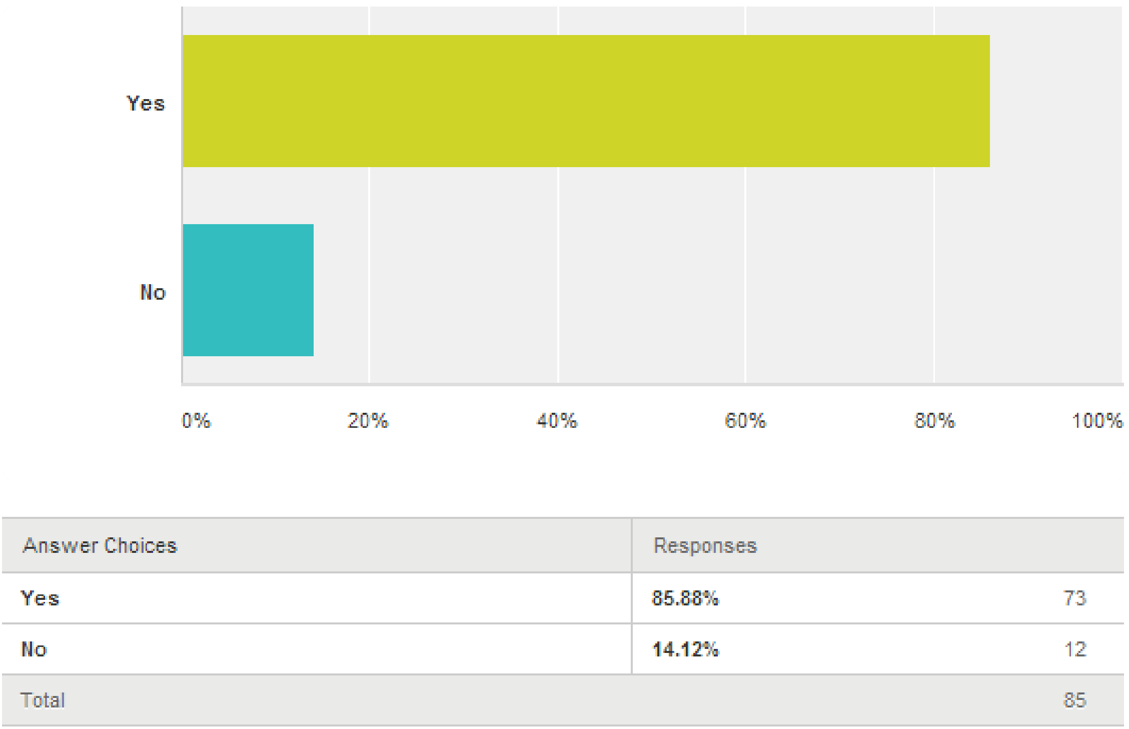
Only software developers or other people who have knowledge on building software could ever have evaluated the expert system. Therefore an email containing a link to both the expert system and questionnaire was initially sent to all students of this School. However, only 35 people responded to the request for questionnaire participation. For this reason, 50 additional responses were found by employing Survey Monkey Audience project [113]. With this project, one can pay Survey Monkey to find responses on behalf of questionnaire owner. The latter needs to simply choose the profile of participants – for example students, employers etc – he or she would like to have; and let Survey Monkey to find these participants. For this questionnaire, Survey Monkey could only guarantee that participants would be IT professionals. This did not necessarily mean that all participants would have knowledge on building software; but since there was not any better way to receive additional responses, Survey Money Audience project was eventually employed.

The total number of responses – including those received by Survey Monkey – was 85.

6.1.3 Results

This subsection demonstrates the results of each question as these were reported by Survey Monkey. Notice that the term "positive responses" – which is used below – refers to the sum of "Strongly Agree" and "Agree" responses. The term "negative responses" refers to the sum of "Strongly Disagree" and "Disagree" responses whereas the term "neutral responses" refers to "Neither Agree nor Disagree" responses.

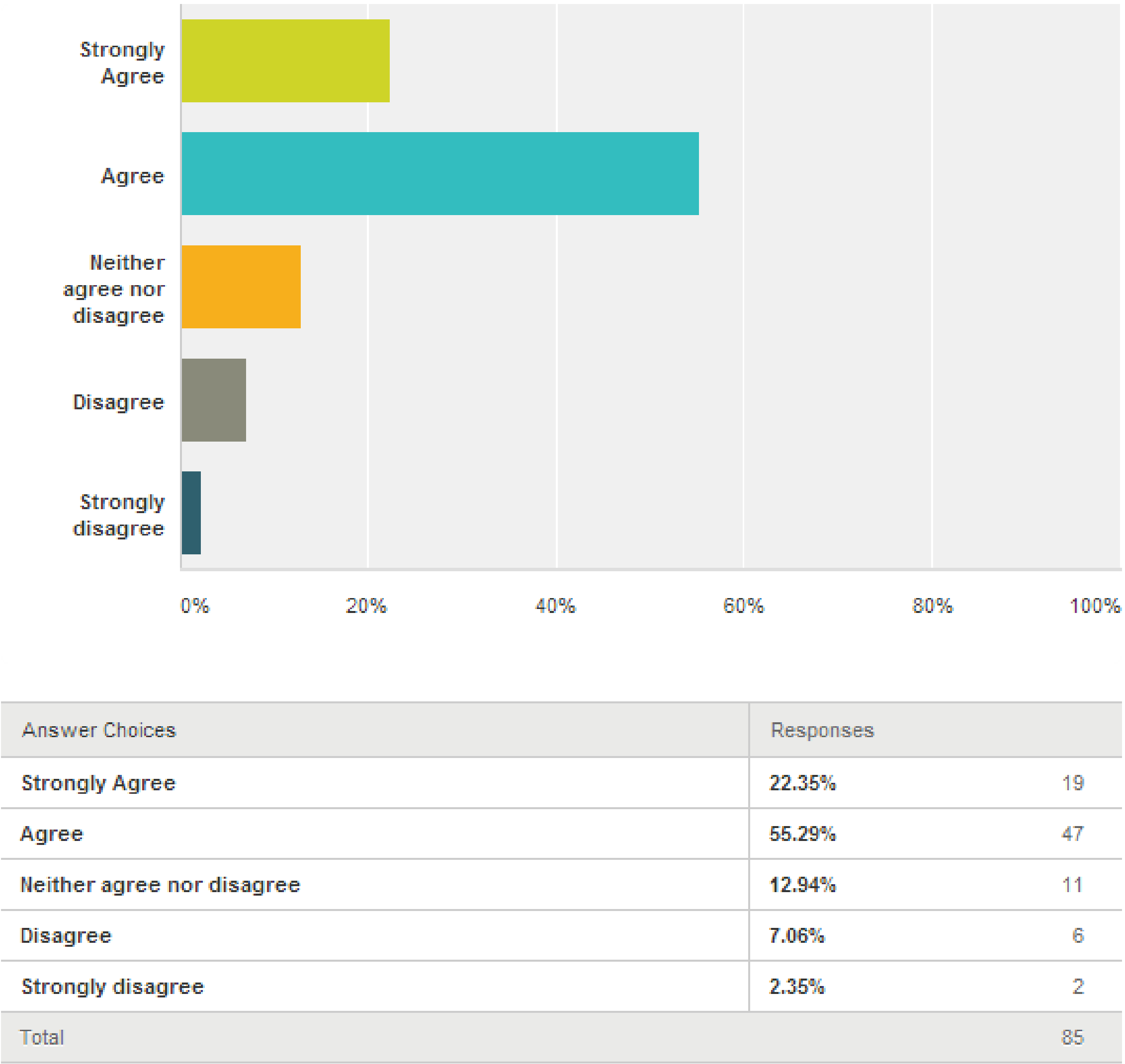
Figure 44 illustrates the results of question 1. Only 14.12% of all participants did not have knowledge on building software.



Answer Choices	Responses	
Yes	85.88%	73
No	14.12%	12
Total		85

Figure 44: Results of Question 1 [111]

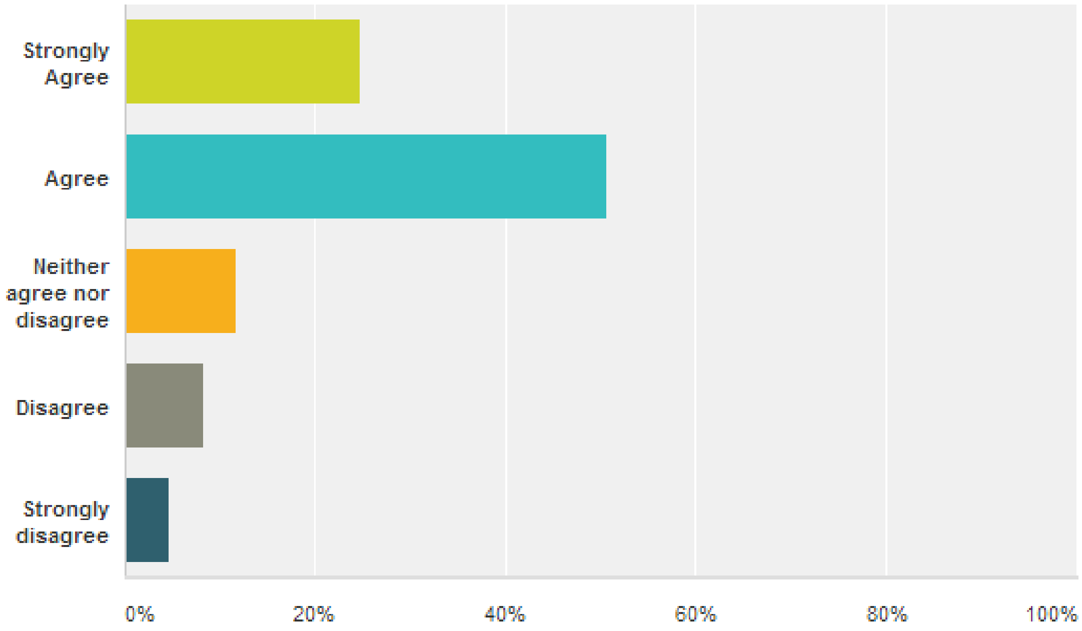
Figure 45 illustrates the results of question 2. There are 77.64% positive responses, 9.41% negative responses and 12.94% neutral responses.

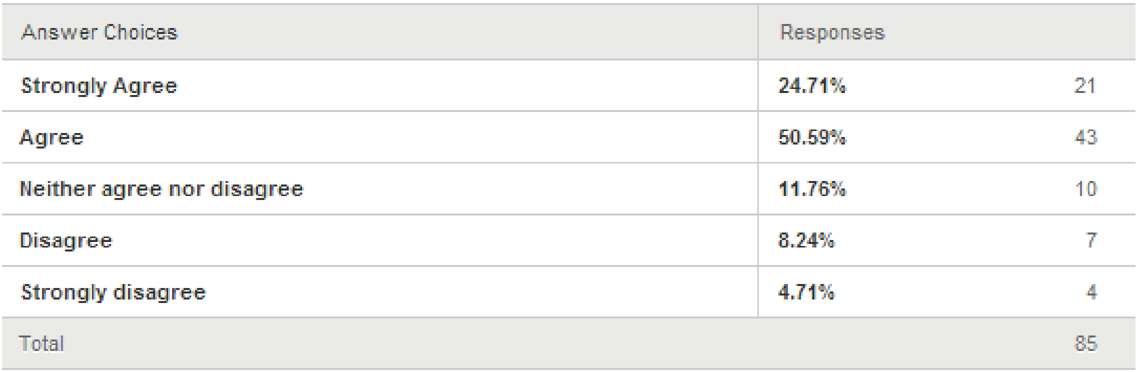


Answer Choices	Responses	
Strongly Agree	22.35%	19
Agree	55.29%	47
Neither agree nor disagree	12.94%	11
Disagree	7.06%	6
Strongly disagree	2.35%	2
Total		85

Figure 45: Results of Question 2 [111]

Figure 46 illustrates the results of question 3. There are 75.3% positive responses, 12.95% negative responses and 11.76% neutral responses. In other words, 75.3% of participants agree that the expert system reduces the amount of research that software developers need to undertake in order to construct a tool for detecting credit fraud. On the other hand, 24.71% of participants either disagree or cannot decide whether the expert system reduces the amount of research or not. This might happen because most software developers would prefer to get the actual code or at least a detailed documentation which would help them write that code. Clearly the expert system is at a higher level of abstraction than documentations and codes. Its purpose is to give software developers a good starting point and not the complete code. Software developers can refer to the literature article which describes the suggested technique in order to find more implementation details.

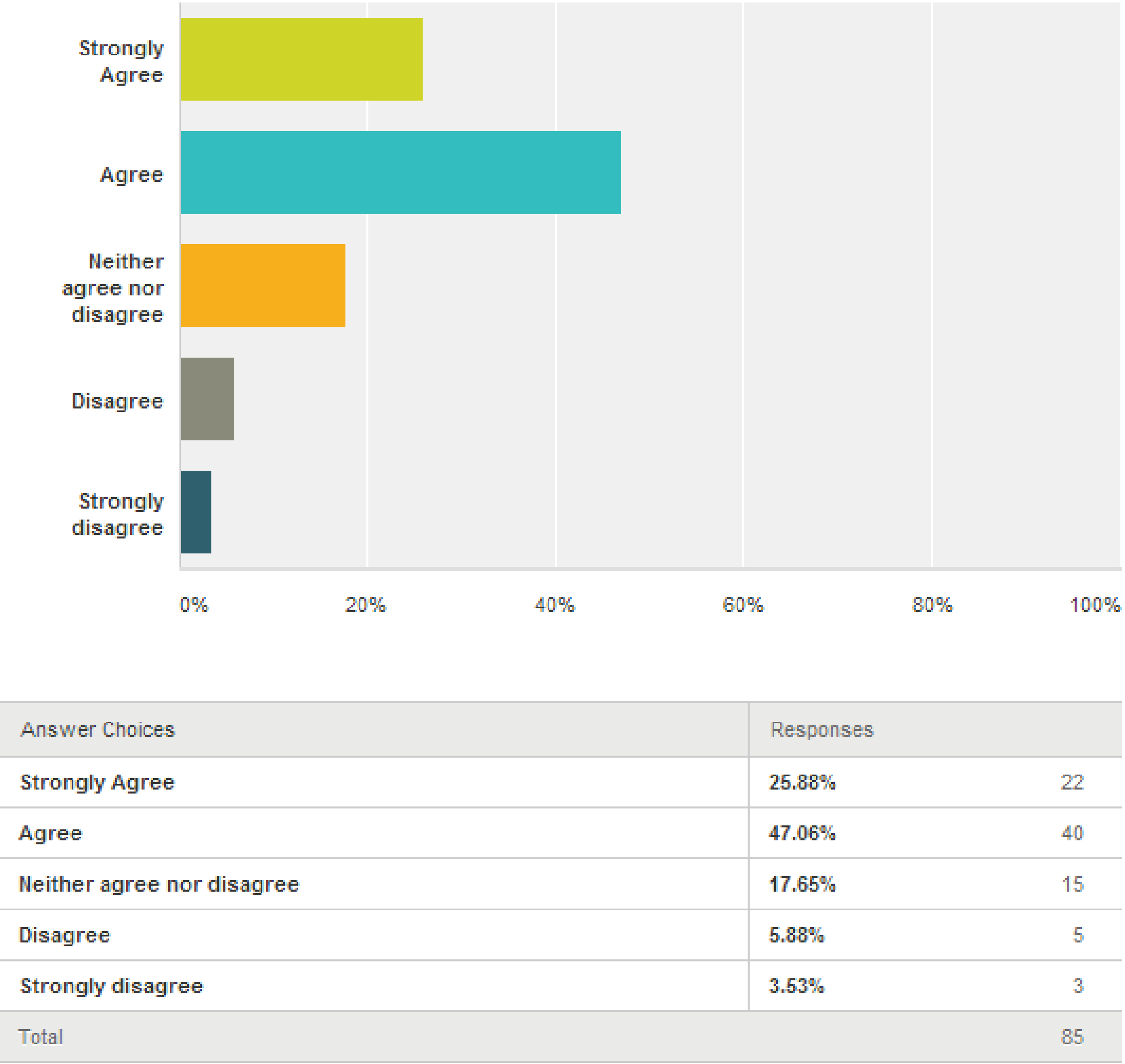




Answer Choices	Responses	
Strongly Agree	24.71%	21
Agree	50.59%	43
Neither agree nor disagree	11.76%	10
Disagree	8.24%	7
Strongly disagree	4.71%	4
Total		85

Figure 46: Results of Question 3 [111]

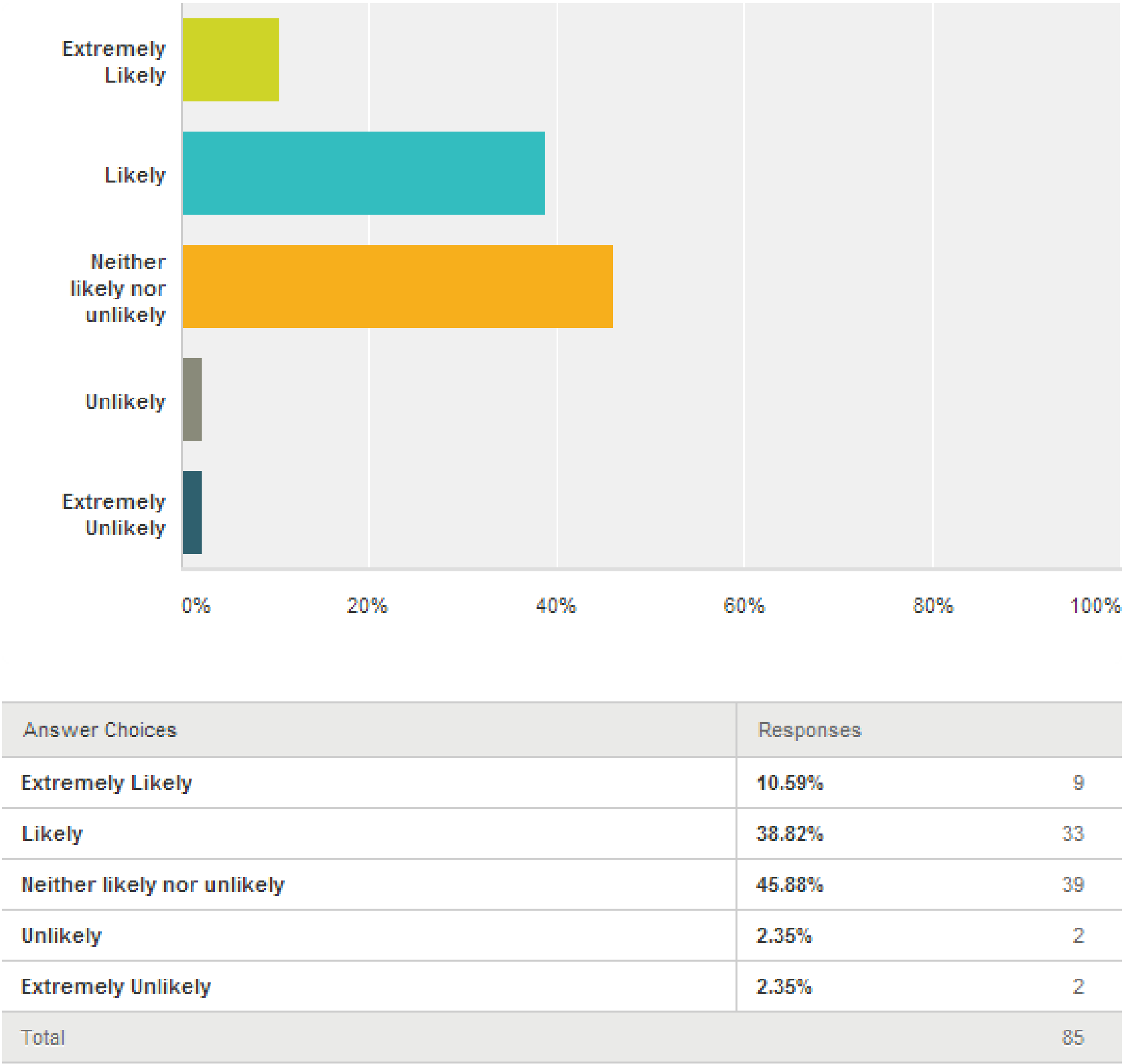
Figure 47 illustrates the results of question 4. There are 72.94% positive responses, 9.41% negative responses and 17.65% neutral responses. In other words, 72.94% of participants believe that the system is useful for software developers; whereas 27.06% either disagree or cannot decide whether it is useful or not. The reason for this disagreement or indecision is probably the same as for figure 46.



Answer Choices	Responses	
Strongly Agree	25.88%	22
Agree	47.06%	40
Neither agree nor disagree	17.65%	15
Disagree	5.88%	5
Strongly disagree	3.53%	3
Total		85

Figure 47: Results of Question 4 [111]

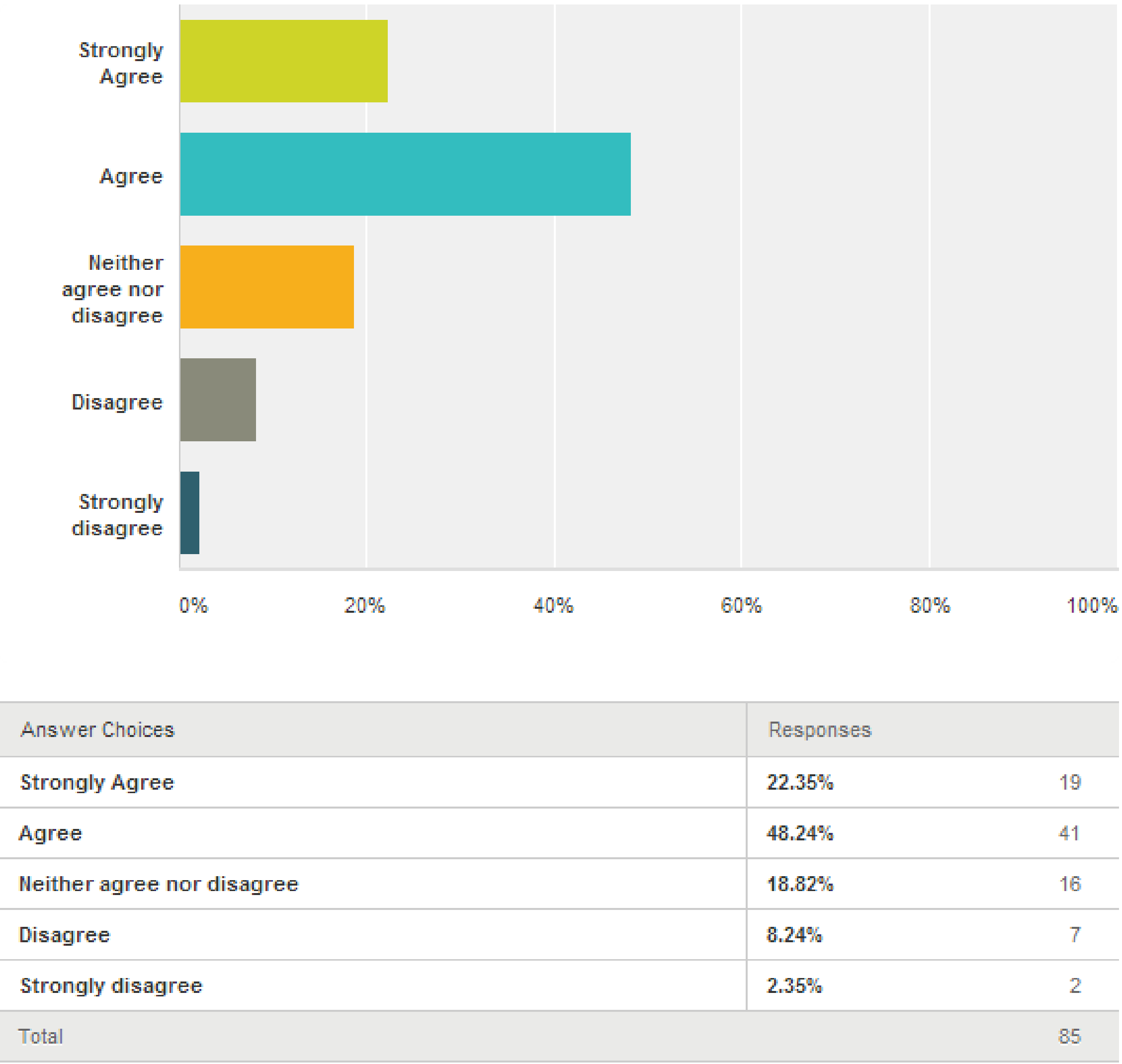
Figure 48 illustrates the results of question 5. There are 49.41% positive responses, 4.70% negative responses and 45.88% neutral responses. Here, 45.88% of participants cannot decide whether they would trust this system in case they wanted to build a credit fraud detection tool. This is a disappointing result. The reason for this indecision was probably the fact that the expert system has been constructed by an MSc student. This could cause participants to be sceptical about the quality and correctness of system's information. Another reason for this indecision might be the fact that most software developers would probably want to have the code instead of the literature article. In addition to that, participants might think that they would never be required to build such a tool and therefore they could not decide if they would ever trust it or not.



Answer Choices	Responses	
Extremely Likely	10.59%	9
Likely	38.82%	33
Neither likely nor unlikely	45.88%	39
Unlikely	2.35%	2
Extremely Unlikely	2.35%	2
Total		85

Figure 48: Results of Question 5 [111]

Figure 49 illustrates the results of question 6. There are 70.59% positive responses, 10.59% negative responses and 18.82% neutral responses. In other words, 70.59% of participants agree that the system was user-friendly and easy to use whereas 29.41% either disagree or cannot decide.

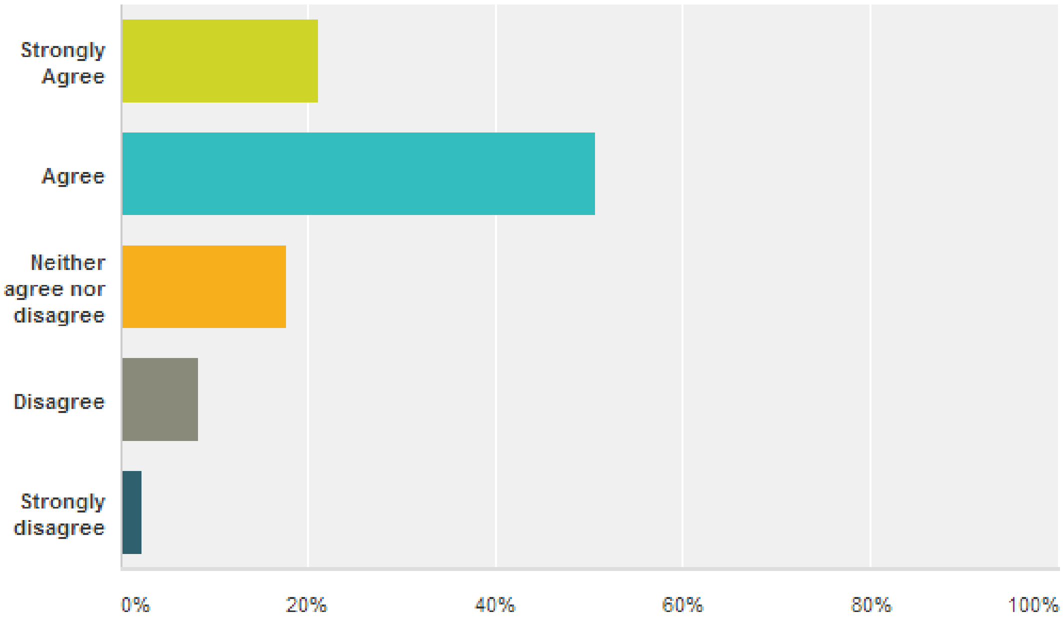


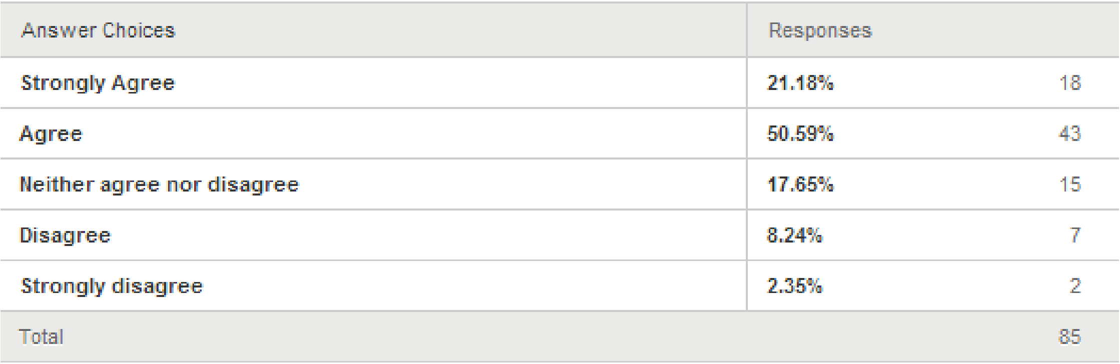
Answer Choices	Responses	
Strongly Agree	22.35%	19
Agree	48.24%	41
Neither agree nor disagree	18.82%	16
Disagree	8.24%	7
Strongly disagree	2.35%	2
Total		85

Figure 49: Results of Question 6 [111]

Figure 50 illustrates the results of question 7. There are 71.77% positive responses, 10.59% negative responses and 17.65% neutral responses. In other words, 71.77% of participants agree that it was easy to understand the questions and the other information contained in the system; whereas 28.24% either disagree or cannot decide.

Recall – from 4.1 – that the expert system contains notes to help users understand the questions. The writer of this report tried to express these notes as well as the questions of expert system in the simplest possible way. The results suggest that a significant proportion of participants – namely 28.24% – disagree or cannot decide whether the questions and notes were easy to understand. Given that the area of fraud detection involves a high level of complexity; the writer of this report still believes that the expert system could not be simpler.

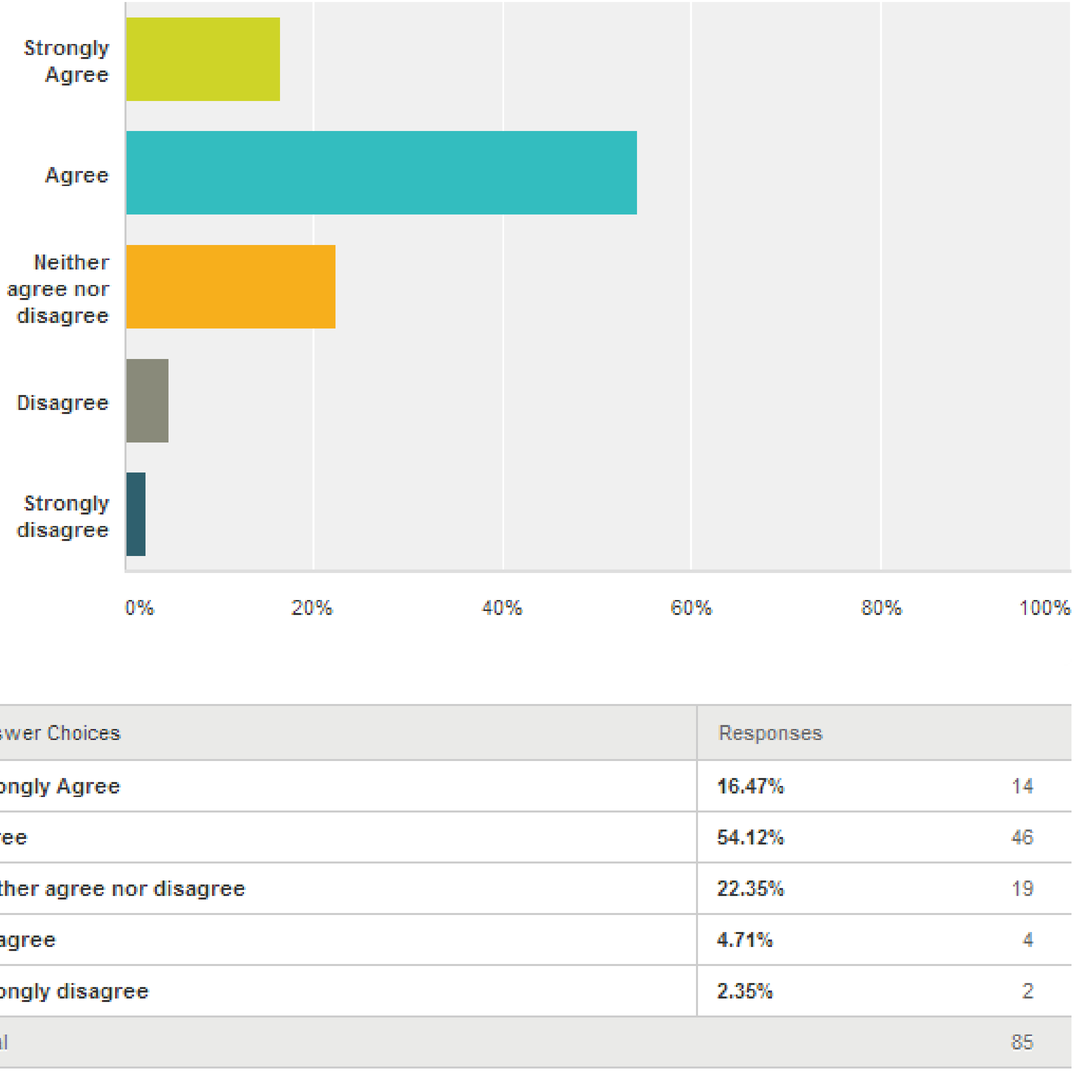




Answer Choices	Responses	
Strongly Agree	21.18%	18
Agree	50.59%	43
Neither agree nor disagree	17.65%	15
Disagree	8.24%	7
Strongly disagree	2.35%	2
Total		85

Figure 50: Results of Question 7 [111]

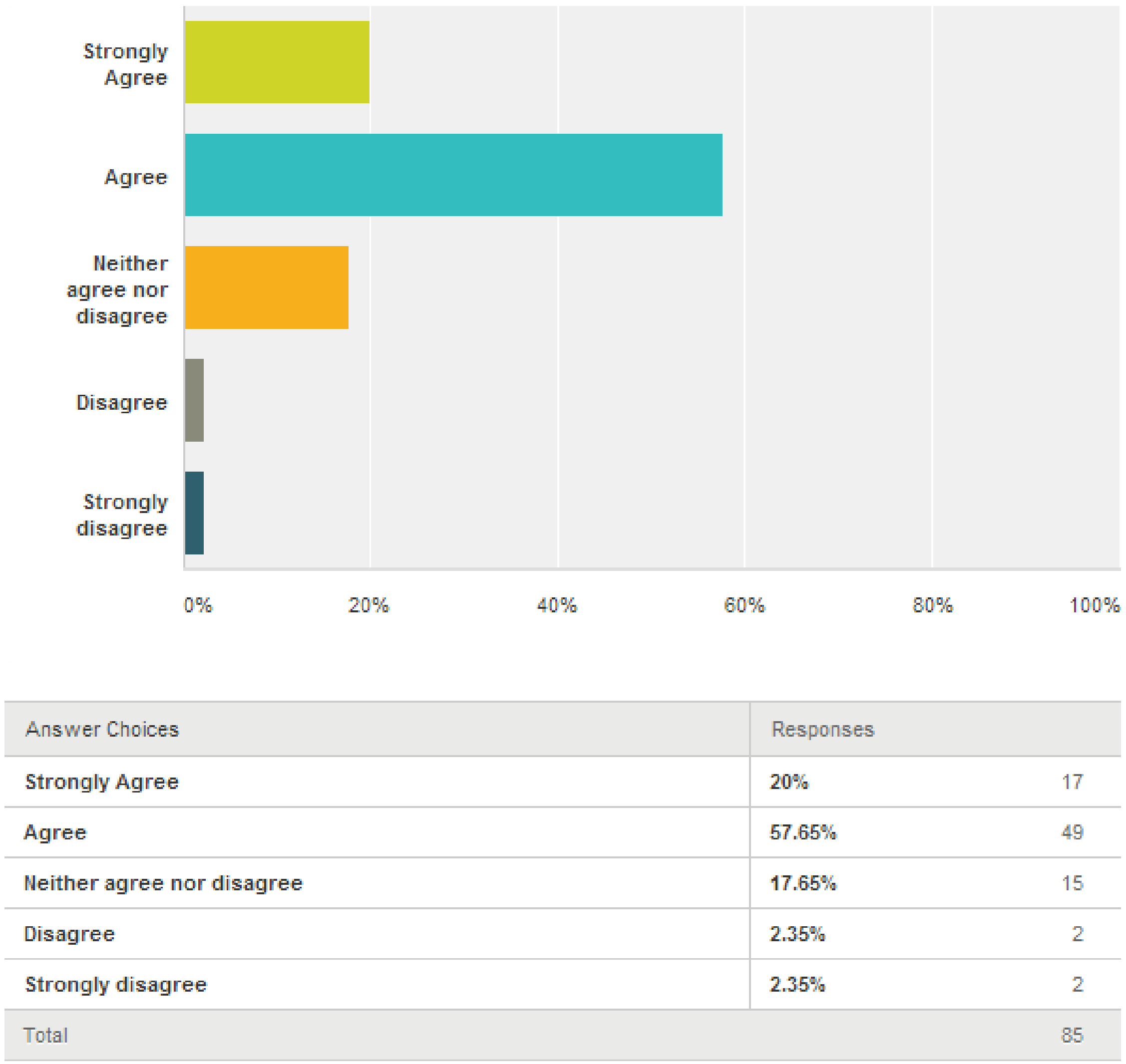
Figure 51 illustrates the results of question 8. There are 70.59% positive responses, 7.06% negative responses and 22.35% neutral responses. In other words, 70.59% of participants agree that the reported scores are a good indication as to which technique is most appropriate to the current user; whereas 29.49% either disagree or cannot decide.



Answer Choices	Responses	
Strongly Agree	16.47%	14
Agree	54.12%	46
Neither agree nor disagree	22.35%	19
Disagree	4.71%	4
Strongly disagree	2.35%	2
Total		85

Figure 51: Results of Question 8 [111]

Figure 52 illustrates the results of question 9. There are 77.65% positive responses, 4.70% negative responses and 17.65% neutral responses. In other words, 77.65% of participants agree that the filtering facility allows users to perform more advanced search on techniques' characteristics. This makes them more capable to choose the most appropriate technique for them. On the other hand, 22.35% of participants either disagree with the above statement or cannot decide at all.



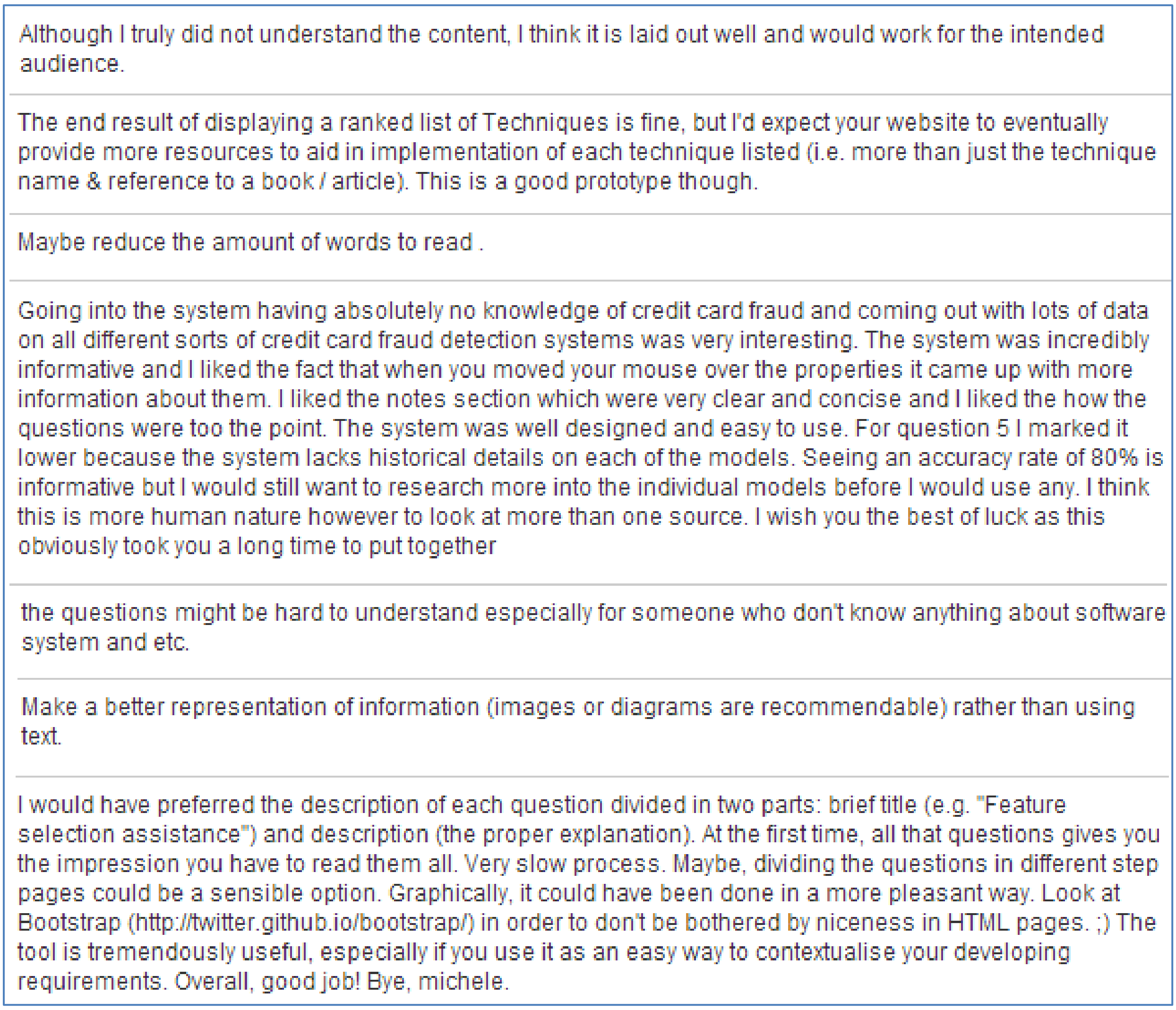
Answer Choices	Responses	
Strongly Agree	20%	17
Agree	57.65%	49
Neither agree nor disagree	17.65%	15
Disagree	2.35%	2
Strongly disagree	2.35%	2
Total		85

Figure 52: Results of Question 9 [111]

Figure 53 illustrates the exact comments provided by participants on question 10. Unfortunately, very few participants have provided a comment on the expert system.

The comments are generally positive; although some of the participants wanted to receive more information regarding technique's implementation. As already mentioned, software developers are looking for the actual code or at least a detailed documentation.

Notice that the first and fifth comments were probably provided by people who have no knowledge on building software.



Although I truly did not understand the content, I think it is laid out well and would work for the intended audience.

The end result of displaying a ranked list of Techniques is fine, but I'd expect your website to eventually provide more resources to aid in implementation of each technique listed (i.e. more than just the technique name & reference to a book / article). This is a good prototype though.

Maybe reduce the amount of words to read .

Going into the system having absolutely no knowledge of credit card fraud and coming out with lots of data on all different sorts of credit card fraud detection systems was very interesting. The system was incredibly informative and I liked the fact that when you moved your mouse over the properties it came up with more information about them. I liked the notes section which were very clear and concise and I liked the how the questions were too the point. The system was well designed and easy to use. For question 5 I marked it lower because the system lacks historical details on each of the models. Seeing an accuracy rate of 80% is informative but I would still want to research more into the individual models before I would use any. I think this is more human nature however to look at more than one source. I wish you the best of luck as this obviously took you a long time to put together

the questions might be hard to understand especially for someone who don't know anything about software system and etc.

Make a better representation of information (images or diagrams are recommendable) rather than using text.

I would have preferred the description of each question divided in two parts: brief title (e.g. "Feature selection assistance") and description (the proper explanation). At the first time, all that questions gives you the impression you have to read them all. Very slow process. Maybe, dividing the questions in different step pages could be a sensible option. Graphically, it could have been done in a more pleasant way. Look at Bootstrap (http://twitter.github.io/bootstrap/) in order to don't be bothered by niceness in HTML pages. ;) The tool is tremendously useful, especially if you use it as an easy way to contextualise your developing requirements. Overall, good job! Bye, michele.

Figure 53: Comments on Question 10 [111]

6.1.4 Hypothesis Acceptance

Recall that the term “positive responses” refers to the sum of “Strongly Agree” and “Agree” responses.

The average percentage of positive responses, which were given to questions 2-9, is 70.7%. The writer of this report believes that this is a good result in general and that the project’s hypothesis can be considered as acceptable. Although the expert system does not contain implementation details; it can reduce the amount of research that software developers need to make if they wish to implement a credit fraud detection tool. The expert system can indicate, to software developers, the right direction of research.

6.2 Conclusion

This chapter evaluated the expert system by presenting the results of the online questionnaire.

The next chapter concludes this dissertation.

7 Conclusion and Future Work

This chapter concludes the dissertation. It starts with a summary of achievements and moves to future work.

7.1 Summary of Achievements

During this project the characteristics of techniques for detecting credit fraud were derived and encapsulated in an ontology – the credit fraud detection ontology. Using this ontology, an expert system was constructed. This can advise software developers as to which technique they should implement in order to detect credit fraud. The expert system asks some questions to the user and then calculates and reports the matching scores of techniques based on user answers. In addition to that, the expert system allows users to filter the detection techniques by their characteristics. The hypothesis of expert system was that it reduces the amount of research that software developers need to make in case they want to implement a credit fraud detection tool. This was validated using an online questionnaire.

It is worth mentioning that the credit fraud detection ontology and hence the expert system support 3 different types of credit fraud as well as 25 different detection techniques in total.

An additional contribution was achieved during this project. This was the construction of another ontology – called generic fraud ontology – which encapsulates the characteristics of 32 different types of fraud and crime including credit fraud. This could be used as the basis of a system capable to inform users for the type of fraud which they have been victimized.

The two ontologies can easily be expanded due to the way in which they have been constructed; that is starting from generic classes and moving to more specific subclasses. People can simply add new subclasses in case they want to encapsulate additional semantics in these ontologies.

It is worth mentioning that this project helped me develop my research and analytical skills due to the significant amount of research which needed to be undertaken. It also helped me improve my understanding on data mining techniques as well as broaden my knowledge on various frauds and crimes including credit fraud.

7.2 Future Work

Few improvements could be done on project's deliverables in the future. The credit fraud detection ontology and expert system could be expanded to support advice on more types of fraud. This would require the finding of techniques which could detect the new fraud types as well as the encapsulation of their characteristics in credit fraud detection ontology. It would be fascinating if the expert system could provide some code or at least a detailed documentation to facilitate the implementation of various detection techniques. Obviously, this is a complicated task and cannot be achieved during a five-month MSc project; given that the number of different detection techniques is large.

Concerning the generic fraud ontology, an associated system could be constructed to advise users for the type of fraud which they have been victimized. In addition to that, the generic fraud ontology could be expanded to encapsulate more types of fraud and crime.

References

[1] P. Gosset and M. Hyland, “Classification, Detection and Prosecution of Fraud on Mobile Networks,” *Proceedings of ACTS Mobile Summit,* 1999.

[2] P. Alexopoulos and K. Kafentzis, “TOWARDS A GENERIC FRAUD ONTOLOGY IN E-GOVERNMENT,” *ICE-B,* pp. 269-276, 2007.

[3] D. O. Friedrichs, Trusted Criminals: White Collar Crime in Contemporary Society, Wadsworth Publishing Co Inc; 4th edition, 2009.

[4] Linda Delamaire, Hussein Addou, John Pointon, “Credit card fraud and detection techniques: a review,” *Banks and Bank Systems,* vol. 4, no. 2, pp. 57-68, 2009.

[5] D. J. Hand, “Fraud Detection in Telecommunications and Banking: Discussion of Becker, Volinsky, and Wilks (2010) and Sudjianto et al. (2010),” *Technometrics,* vol. 52, no. 1, pp. 34-38, 2010.

[6] R. J. Bolton and D. J. Hand, “Unsupervised Profiling Methods for Fraud Detection,” London, 2002.

[7] “Business Wire,” 21 November 2011. [Online]. Available: http://www.businesswire.com/news/home/20111121005121/en/U.S.-Leads-World-Credit-Card-Fraud-states. [Accessed 25 March 2013].

[8] J. T. Quah and M. Sriganesh, “Real-time credit card fraud detection using computational intelligence,” *Expert Systems with Applications,* pp. 1721-1732, 2008.

[9] N. Laleh and A. M. Azgomi, “A Taxonomy of Frauds and Fraud Detection Techniques,” *ICISTM,* vol. 31, pp. 256-267, 2009.

[10] E. Aleskerov, B. Fieisleben and R. Bharat, “CARDWATCH: A Neural Network Based Database Mining System for Credit Card Fraud Detection,” *Department of Electrical Engineering and Computer Science, University of Siegen,* pp. 220-226, 1997.

[11] N. F. R. Centre, “Bank card and cheque fraud,” National Fraud Authority, UK.

[12] R. Patidar and L. Sharma, “Credit Card Fraud Detection Using Neural Network,” *International Journal of Soft Computing and Engineering (IJSCE),* vol. 1, no. NCAI2011, pp. 2231-2307, 2011.

[13] T. Xiong, S. Wang, A. Mayers and E. Monga, “Personal bankruptcy prediction by mining credit card data,” *Expert Systems with Applications,* pp. 665-676, 2013.

[14] J. Whittaker, C. Whitehead and M. Somers, “The neglog transformation and quantile regression for the analysis of a large credit scoring database,” *Royal*

Statistical Society, vol. 54, no. 5, pp. 863-878, 2005.

[15] S. B. Hoar, “Identity Theft: The Crime of the New Millennium,” *Or. L.,* pp. 1423-1448, 2001.

[16] B.-J. Koops and R. Leenes, “Identity theft, identity fraud and/or identity-related crime,” *Datenschutz und Datensicherheit-DuD,* vol. 30, no. 9, pp. 553-556, 2006.

[17] “Identity fraud and identity theft,” Action Fraud, [Online]. Available: www.actionfraud.police.uk/fraud_protection/identity_fraud. [Accessed 22 June 2013].

[18] C. Phua, R. Gayler, V. Lee and K. Smith-Miles, “On the communal analysis suspicion scoring for identity crime in streaming credit applications,” *European Journal of Operational Research,* vol. 195, pp. 595-612, 2009.

[19] R. Bose, “Intelligent Technologies for Managing Fraud and Identity Theft,” *IEEE Computer Society,* 2006.

[20] A. Abdelhalim and I. Traore, “Identity Application Fraud Detection using Web Mining and Rule-based Decision Tree,” *International Journal of Computer and Network Security (IJCNS),* vol. 1, no. 1, pp. 31-44, 2009.

[21] M.-S. Chen, J. Han and P. S. Yu, “Data mining: An Overview from a Database Perspective,” *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGIJEERING,* vol. 8, no. 6, pp. 866-883, 1996.

[22] J. Frand, “Data Mining: What is Data Mining?,” [Online]. Available: http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm. [Accessed 29 March 2013].

[23] A. M. Hormozi and S. Giles, “Data mining: A competitive weapon for banking and retail industries,” *Information Systems Management,* pp. 62-71, 2004.

[24] A. Srivastava, A. Kundu, S. Sural and A. K. Majumdar, “Credit Card Fraud Detection Using Hidden Markov Model,” *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING,* vol. 5, no. 1, pp. 37-48, 2008.

[25] S. Maes, K. Tuyls, B. Vanschoenwinkel and B. Manderick, “Credit Card Fraud Detection Using Bayesian and Neural Networks,” Vrije Universiteit Brussel - Department of Computer Science, Belgium, 2002.

[26] X. Zhu, “Semi-Supervised Learning Literature Survey,” University of Wisconsin, Madison, 2008.

[27] E. Ngai, Y. Hu, Y. Wong, Y. Chen and X. Sun, “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,” *Decision Support Systems,* pp. 559-569, 2011.

[28] C. Gershenson, “Artificial Neural Networks for Beginners,” 2003.

[29] S. R. Gunn, “Support Vector Machines for Classification and Regression,” UNIVERSITY OF SOUTHAMPTON, UK, 1998.

[30] D. Meyer, “Support Vector Machines,” Technische Universit¨at Wien,, Austria, 2012.

[31] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand and D. Steinberg, “Top 10 algorithms in data mining,” *Springer,* pp. 1-37, 2007.

[32] R. P. Ang and D. H. Goh, “Predicting Juvenile Offending: A Comparison of Data Mining Methods,” *International Journal of Offender Therapy and Comparative Criminology,* vol. 57, no. 2, pp. 191-207, 2013.

[33] J. Cheng and R. Greiner, “Learning Bayesian Belief Network Classifiers: Algorithms and System,” Department of Computing Science, University of Alberta, Canada, 2001.

[34] “Decision Trees - What are they?,” Decision Trees for Business Intelligence and Data Mining: Using SAS Enterprise Miner.

[35] T. M. Mitchell and H. McGraw, *Decision Tree Learning,* 1997.

[36] J. Bastos, “Credit scoring with boosted decision trees,” CEMAPRE, School of Economics and Management (ISEG), Technical University of Lisbon, 2007.

[37] Y. Sahin and E. Duman, “Detecting Credit Card Fraud by Decision Trees and Support Vector Machines,” *Proceedings of the International MultiConference of Engineers and Computer Scientists,* vol. I, pp. 1-6, 2011.

[38] S. Hawkins, H. He, G. Williams and R. Baxter, “Outlier Detection Using Replicator Neural Networks,” CSIRO Mathematical and Information Sciences, Australia, 2002.

[39] W.-F. YU and N. Wang, “Research on Credit Card Fraud Detection Model Based on Distance Sum,” *International Joint Conference on Artificial Intelligence,* pp. 353-356, 2009.

[40] “Outlier -- from Wolfram MathWorld,” Wolfram , [Online]. Available: http://mathworld.wolfram.com/Outlier.html. [Accessed 1 April 2013].

[41] I. Ben-Gal, “OUTLIER DETECTION,” in *Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, Israel, Kluwer Academic Publishers, 2005, pp. 1-16.

[42] D. J. Weston, D. J. Hand, N. M. Adams, C. Whitrow and P. Juszczak, “Plastic card fraud detection using peer group analysis,” *Springer-Verlag,* pp. 45-62,

2008.

[43] Z. Ghahramani, “An Introduction to Hidden Markov Models and Bayesian Networks,” *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 15, no. 1, 2001.

[44] P. Blunsom, “Hidden Markov Models,” 2004.

[45] A. Brabazon, J. Cahill, P. Keenan and D. Walsh, “Identifying Online Credit Card Fraud using Artificial Immune,” in *IEEE Congress on Evolutionary Computation (CEC)*, Dublin, 2010.

[46] J. Tuo, S. Red, W. Lid, X. Li, B. Li and L. Lei, “Artificial Immune System for Fraud Detection,” in *IEEE International Conference on Systems, Man and Cybernetics*, China, 2004.

[47] C. Elkan, “Nearest Neighbor Classification,” 2011.

[48] T. Dingsoyr and E. M. Lidal, “An Evaluation of Data Mining Methods and Tools,” Norwegian University of Science and Technology (NTNU), Norway, 1997.

[49] M. F. A. Gadi, X. Wang and A. Pereira do Lago, “Credit Card Fraud Detection with Artificial Immune System,” *Springer-Verlag Berlin Heidelberg,* pp. 119-131, 2008.

[50] B. Wiese and C. Omlin, “Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks,” *Innovations in Neural Infor. Paradigms & Appli.,* pp. 231-268, 2009.

[51] T. GUO and G.-Y. LI, “NEURAL DATA MINING FOR CREDIT CARD FRAUD DETECTION,” in *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, Kunming, 2008.

[52] D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing,* vol. 4, pp. 68-85, 1994.

[53] R.-C. Chen, T.-S. Chen and C.-C. Lin, “A new binary support vector system for increasing detection rate of credit card fraud,” *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 20, no. 2, pp. 227-239, 2006.

[54] R.-C. Chen, M.-L. Chiu, Y.-L. Huang and L.-T. Chen, “Detecting Credit Card Fraud by Using Questionnaire-Responded Transaction Model Based on Support Vector Machines,” *Springer-Verlag Berlin Heidelberg,* pp. 800-806, 2004.

[55] H. Li, J. Sun and J. Wu, “Predicting business failure using classification and regression tree: An empirical comparison with popular classical statistical methods and top classification mining methods,” *Expert Systems with Applications,* pp. 5895-5904, 2010.

[56] K. Yamanishi and J.-I. Takeuchi, “On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms,” *Data Mining and Knowledge Discovery,* pp. 275-300, 2004.

[57] R. L. Wilson and R. Sharda, “Bankruptcy prediction using neural networks,” *Decision Support Systems,* pp. 545-557, 1994.

[58] P. C. Pendharkar, “A threshold-varying artificial neural network approach for classification and its application to bankruptcy prediction problem,” *Computers & Operations Research,* pp. 2561-2582, 2005.

[59] J. H. Min and Y.-C. Lee, “Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters,” *Expert Systems with Applications,* pp. 603-614, 2005.

[60] C.-H. Wu, G.-H. Tzeng, Y.-J. Goo and W.-C. Fang, “A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy,” *Expert Systems with Applications,* pp. 397-408, 2007.

[61] H.-L. Chen, B. Yang, G. Wang, J. Liu, X. Xu, S.-J. Wang and D.-Y. Liu, “A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method,” *Knowledge-Based Systems,* pp. 1348-1359, 2011.

[62] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” Purdue School of Engineering and Technology, Washington, 1995.

[63] L. Breiman, J. Friedman, C. J. Stone and R. Olshen, Classification and Regression Trees, Chapman and Hall, 1984.

[64] I. A. F. D. u. W. M. a. R.-b. D. Tree, “Abdelhalim, Amany; Traore, Issa,” *(IJCNS) International Journal of Computer and Network Security,* vol. 1, no. 1, pp. 31-44, 2009.

[65] T. Gruber, “What is an Ontology?,” Stanford University, [Online]. Available: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html. [Accessed 6 April 2013].

[66] C. Martinez-Cruz, I. J. Blanco and M. A. Vila, “Ontologies versus relational databases: are they so different? A comparison,” *Artificial Intelligence Review,* pp. 271-290, 2012.

[67] M. Horridge, “A Practical Guide To Building OWL Ontologies Using Protègè 4 and CO-ODE Tools, Edition 1.3,” The University of Manchester, 2011.

[68] F. Baader, I. Horrocks and U. Sattler, “Description Logics,” Elsevier, 2007.

[69] S. C. f. B. I. Research, “Protégé,” Stanford University School of Medicine, [Online]. Available: http://protege.stanford.edu/. [Accessed 8 April 2013].

[70] C. S. Hilas and P. A. Mastorocostas, “An application of supervised and unsupervised learning approaches to telecommunications fraud detection,” *Knowledge-Based Systems,* vol. 21, no. 7, pp. 721-726, 2008.

[71] P. Hoath, “Telecoms fraud, the gory details,” *Computer Fraud & Security,* vol. 1, pp. 10-14, 1998.

[72] S. Rosset, U. Murad, E. Neumann, Y. Idan and G. Pinkas, “Discovery of Fraud Rules for Telecommunications - Challenges and Solutions,” *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining,* pp. 409-413, 1999.

[73] C. Tressler, “Is Your Mobile Bill a Cram Sandwich?,” National Consumer Protection Week, [Online]. Available: http://www.ncpw.gov/blog/your-mobile-bill-cram-sandwich. [Accessed 19 June 2013].

[74] “Mystery Charges on Your Phone Bill,” FEDERAL TRADE COMMISSION, [Online]. Available: http://www.consumer.ftc.gov/articles/0183-mystery-charges-your-phone-bill. [Accessed 19 June 2013].

[75] “What Is Securities Fraud?,” [Online]. Available: http://www.whistleblowingprotection.org/?q=node/32. [Accessed 21 June 2013].

[76] “Securities Fraud,” The Sheena Law Firm, [Online]. Available: http://sheenalawfirm.com/securitiesfraud.htm. [Accessed 21 June 2013].

[77] S. Dolgopolov, “Insider Trading,” Library of Economics and Liberty, [Online]. Available: http://www.econlib.org/library/Enc/InsiderTrading.html. [Accessed 21 June 2013].

[78] J. Clark, “How Insider Trading Works,” [Online]. Available: http://money.howstuffworks.com/insider-trading1.htm. [Accessed 21 June 2013].

[79] M. A. McDermott, “Ponzi Schemes and the Law of Fraudulent and Preferential Transfers,” *Am. Bankr.,* 1998.

[80] “"Pump-and-Dumps" and Market Manipulations,” U.S. Securities and Exchange Commission, [Online]. Available: http://www.sec.gov/answers/pumpdump.htm. [Accessed 21 June 2013].

[81] “Pump & Dump: Tips for Avoiding Stock Scams on the Internet,” U.S. Securities and Exchange Commission. [Online]. [Accessed 21 June 2013].

[82] M. Clarke, “Insurance Fraud,” *The British Journal of Criminology,* vol. 29, no. 1, pp. 1-20, 1989.

[83] S. Viaene and G. Dedene, “Insurance fraud: issues and challenges,” *The Geneva Papers on Risk and Insurance-Issues and Practice ,* vol. 29, no. 2, pp. 313-333, 2004.

[84] “Insurance Fraud,” Insurance Information Institute, [Online]. Available: http://www.iii.org/facts_statistics/fraud.html. [Accessed 23 June 2013].

[85] T. H. Nguyen and H. N. Pontell, “Mortgage origination fraud and the global economic crisis,” *Criminology & Public Policy,* vol. 9, no. 3, pp. 591-612, 2010.

[86] J. Smith, “The Structural Causes of Mortgage Fraud,” *Syracuse Law Review,* vol. 60, 2010.

[87] D. Masciandaro, “Money laundering: the economics of regulation,” *European Journal of Law and Economics,* vol. 7, no. 3, pp. 225-240, 1999.

[88] M. Levi, “Money laundering and its regulation,” *The Annals of the American Academy of Political and Social Science,* vol. 582, no. 1, pp. 181-194, 2002.

[89] P. J. Quirk, “Money laundering: muddying the macroeconomy,” *Finance and Development,* vol. 34, pp. 7-9, 1997.

[90] J. Walker, “How big is global money laundering?,” *Journal of Money Laundering Control,* vol. 3, no. 1, pp. 25-37, 1999.

[91] W. Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall; 5 edition, 2010.

[92] F. Cohen, “Computer viruses: theory and experiments,” *Computers & security,* vol. 6, no. 1, pp. 22-35, 1987.

[93] B. K. Mishra and N. Jha, “Fixed period of temporary immunity after run of anti-malicious software on computer nodes,” *Applied Mathematics and Computation,* vol. 190, no. 2, pp. 1207-1212, 2007.

[94] C. M. Kahn and W. Roberds, “Credit and identity theft,” *Journal of Monetary Economics,* vol. 55, no. 2, pp. 251-264, 2008.

[95] “Fighting Friendly Fraud Chargebacks,” Risk Payments, [Online]. Available: http://www.riskpayments.com/knowledge-base/articles/fighting-friendly-fraud-chargebacks/. [Accessed 24 June 2013].

[96] “Friendly Fraud - A Very Real Problem,” Friendly Fraud Prevent, [Online]. Available: http://www.friendlyfraudprevent.com/FriendlyFraud.aspx. [Accessed 24 June 2013].

[97] K.-S. Shin, T. S. Lee and H.-j. Kim, “An application of support vector machines in bankruptcy prediction model,” *Expert Systems with Applications,* pp. 127-135, 2005.

[98] S. K.K and R. Nedunchezhian, “Boat Adaptive Credit Card Fraud Detection System,” Department of Information Technology, Toc H Institute of science & Technology, Ernakulam, India, 2010.

[99] P. Cunningham and S. J. Delany, “k-Nearest Neighbour Classifiers,” University College Dublin, 2007.

[100] S. Lawrence, C. L. Giles and A. C. Tsoi, “Lessons in Neural Network Training: Overfitting May be Harder than Expected,” *Proceedings of the Fourteenth National Conference on Artificial Intelligence,* pp. 540-545, 1997.

[101] N. Friedman and M. Goldszmidt, “Discretizing Continuous Attributes While Learning Bayesian Networks,” 1996.

[102] J. Cheng, D. A. Bell and W. Liu, “An Algorithm for Bayesian Belief Network Construction from Data,” School of Information and Software Engineering, University of Ulster at Jordanstown, Northern Ireland, 1997.

[103] A. Bharadwaj and S. Minz, “Discretization based Support Vector Machines for Classification,” *Journal of the Indian Society of Agricultural Statistics,* vol. 63, no. 2, pp. 189-197, 2009.

[104] A. Varga and R. Moore, “Hidden Markov Model Decomposition of Speech And Noise,” Speech Research Unit, Royal Signals and Radar Establishment, Great Britain, 1990.

[105] J. Vlissides, R. Johnson, R. Helm and E. Gamma, Design patterns: Elements of reusable object-oriented software, Addison-Wesley, 1995.

[106] “The OWL API,” [Online]. Available: http://owlapi.sourceforge.net/. [Accessed 31 July 2013].

[107] “Java Advantages and Disadvantages,” [Online]. Available: http://www.webdotdev.com/nvd/articles-reviews/java/java-advantages-and-disadvantages-1042.html. [Accessed 2 August 2013].

[108] “JavaServer Faces (JSF) Tutorial,” [Online]. Available: http://www.tutorialspoint.com/jsf/. [Accessed 2 August 2013].

[109] “JavaServer Faces Technology Overview,” [Online]. Available: http://www.oracle.com/technetwork/java/javaee/overview-140548.html. [Accessed 2 August 2013].

[110] C. Larman and V. R. Basili, “Iterative and incremental developments: a brief history,” *Computer,* vol. 36, no. 6, pp. 47-56, 2003.

[111] Survey Monkey, [Online]. Available: http://www.surveymonkey.com/. [Accessed 6 August 2013].

[112] “Computer Science Ethics,” School of Computer Science, [Online]. Available: http://ethics.cs.manchester.ac.uk/. [Accessed 6 August 2013].

[113] “Survey Monkey Audience,” [Online]. Available:

http://www.surveymonkey.com/mp/audience/. [Accessed 6 August 2013].

[114] J. Baek and S. Cho, "Bankruptcy Prediction for Credit Risk Using an Auto-Associative Neural Network in Korean firms," Department of Industrial Engineering, Seoul National University, Korea, 2003.

[115] O. Sutton, "Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction," 2012.

[116] A. Oniśko, M. J. Druzdzel and H. Wasyluk, "Learning Bayesian network parameters from small data sets: application of Noisy-OR gates," *International Journal of Approximate Reasoning,* vol. 27, pp. 165-182, 2001.

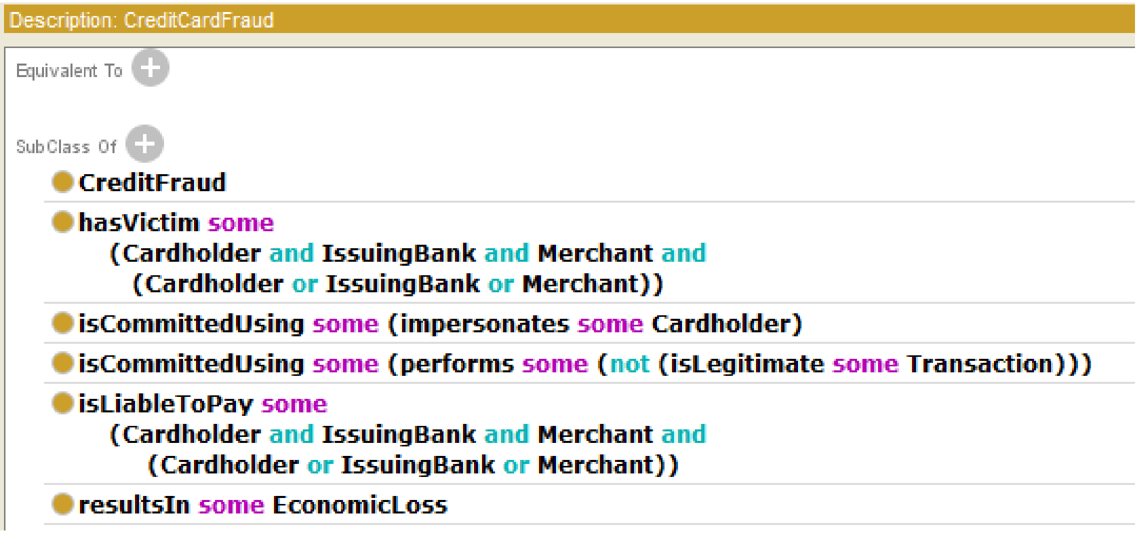
[117] M. C. van Wezel, J. N. Kok and K. Sere, "Determining the Number of Dimensions Underlying Customer-choices with a Competitive Neural Network," Centre for Mathematics and Computer Science (CWI), Amsterdam, 1996.

[118] A. Mittal and L.-F. Cheong, "Addressing the Problems of Bayesian Network Classification of Video Using High-Dimensional Features," *IEEE Transactions on Knowledge and Data Engineering,* vol. 16, no. 2, pp. 230-244, 2004.

Appendix

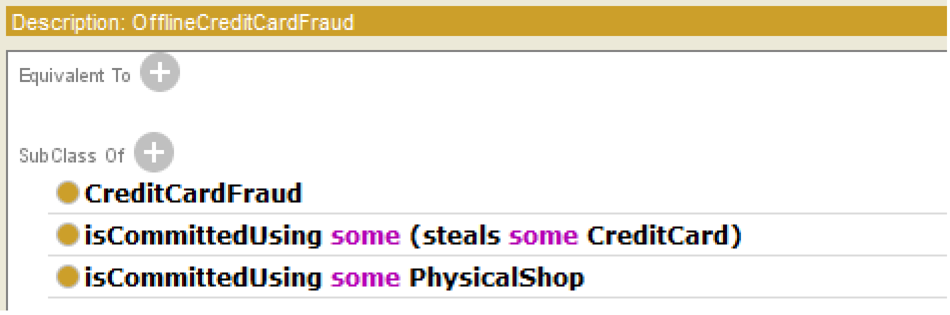
This appendix contains screenshots of all the frauds and crimes which are encapsulated by generic fraud ontology but are not included in subsection 3.5.3.

Credit Card Fraud



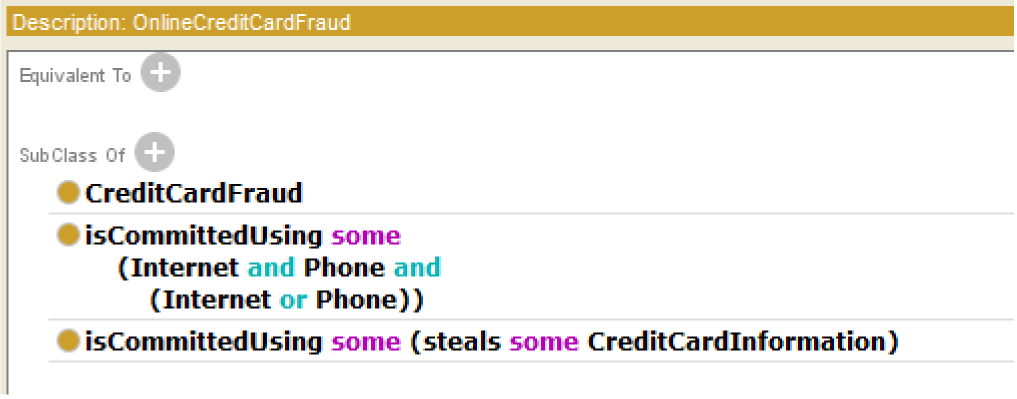
```
Description: CreditCardFraud

Equivalent To

SubClass Of
  ● CreditFraud
  ● hasVictim some
        (Cardholder and IssuingBank and Merchant and
          (Cardholder or IssuingBank or Merchant))
  ● isCommittedUsing some (impersonates some Cardholder)
  ● isCommittedUsing some (performs some (not (isLegitimate some Transaction)))
  ● isLiableToPay some
        (Cardholder and IssuingBank and Merchant and
           (Cardholder or IssuingBank or Merchant))
  ● resultsIn some EconomicLoss
```

Figure 54: Semantics of Credit Card Fraud



```
Description: OfflineCreditCardFraud

Equivalent To

SubClass Of
  ● CreditCardFraud
  ● isCommittedUsing some (steals some CreditCard)
  ● isCommittedUsing some PhysicalShop
```

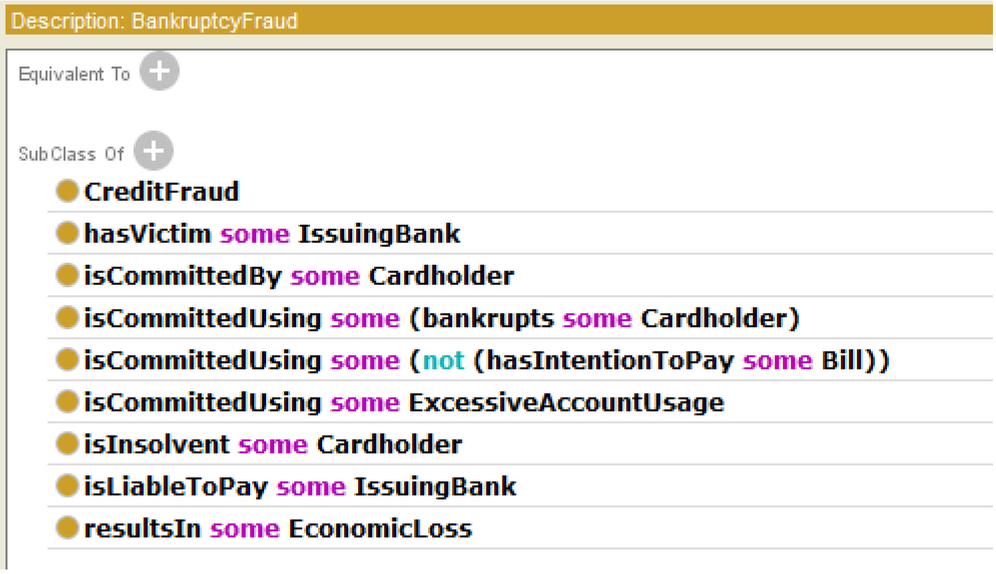
Figure 55: Semantics of Offline Credit Card Fraud



```
Description: OnlineCreditCardFraud

Equivalent To

SubClass Of
  ● CreditCardFraud
  ● isCommittedUsing some
        (Internet and Phone and
           (Internet or Phone))
  ● isCommittedUsing some (steals some CreditCardInformation)
```

Figure 56: Semantics of Online Credit Card Fraud

Bankruptcy Fraud



Description: BankruptcyFraud

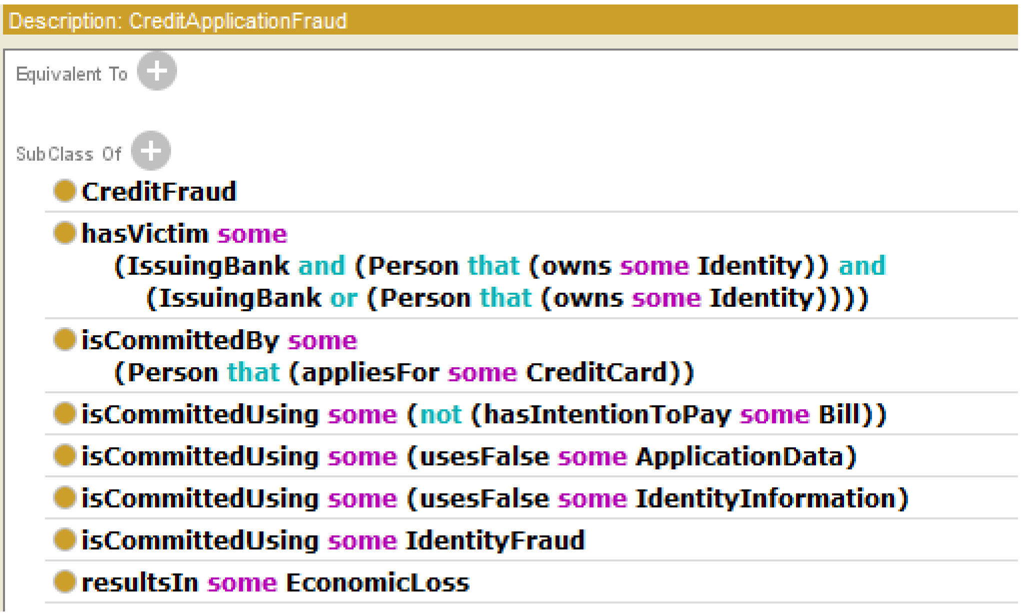
Equivalent To

SubClass Of

- CreditFraud
- hasVictim some IssuingBank
- isCommittedBy some Cardholder
- isCommittedUsing some (bankrupts some Cardholder)
- isCommittedUsing some (not (hasIntentionToPay some Bill))
- isCommittedUsing some ExcessiveAccountUsage
- isInsolvent some Cardholder
- isLiableToPay some IssuingBank
- resultsIn some EconomicLoss

Figure 57: Semantics of Bankruptcy Fraud

Credit Application Fraud



Description: CreditApplicationFraud

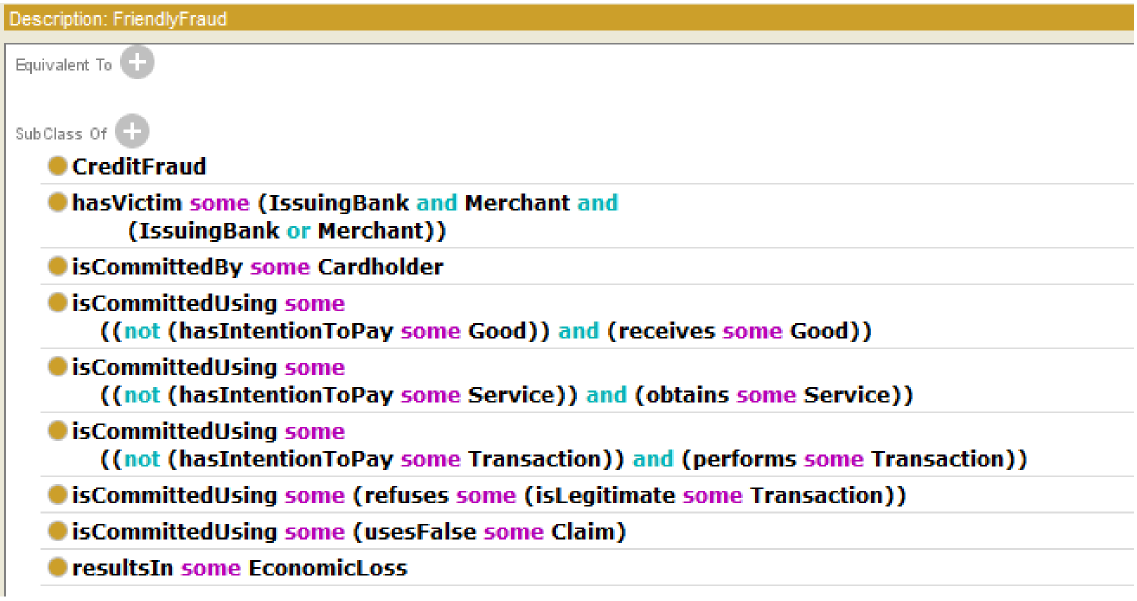
Equivalent To

SubClass Of

- CreditFraud
- hasVictim some
  (IssuingBank and (Person that (owns some Identity)) and
  (IssuingBank or (Person that (owns some Identity))))
- isCommittedBy some
  (Person that (appliesFor some CreditCard))
- isCommittedUsing some (not (hasIntentionToPay some Bill))
- isCommittedUsing some (usesFalse some ApplicationData)
- isCommittedUsing some (usesFalse some IdentityInformation)
- isCommittedUsing some IdentityFraud
- resultsIn some EconomicLoss

Figure 58: Semantics of Credit Application Fraud

Friendly Fraud



Description: FriendlyFraud

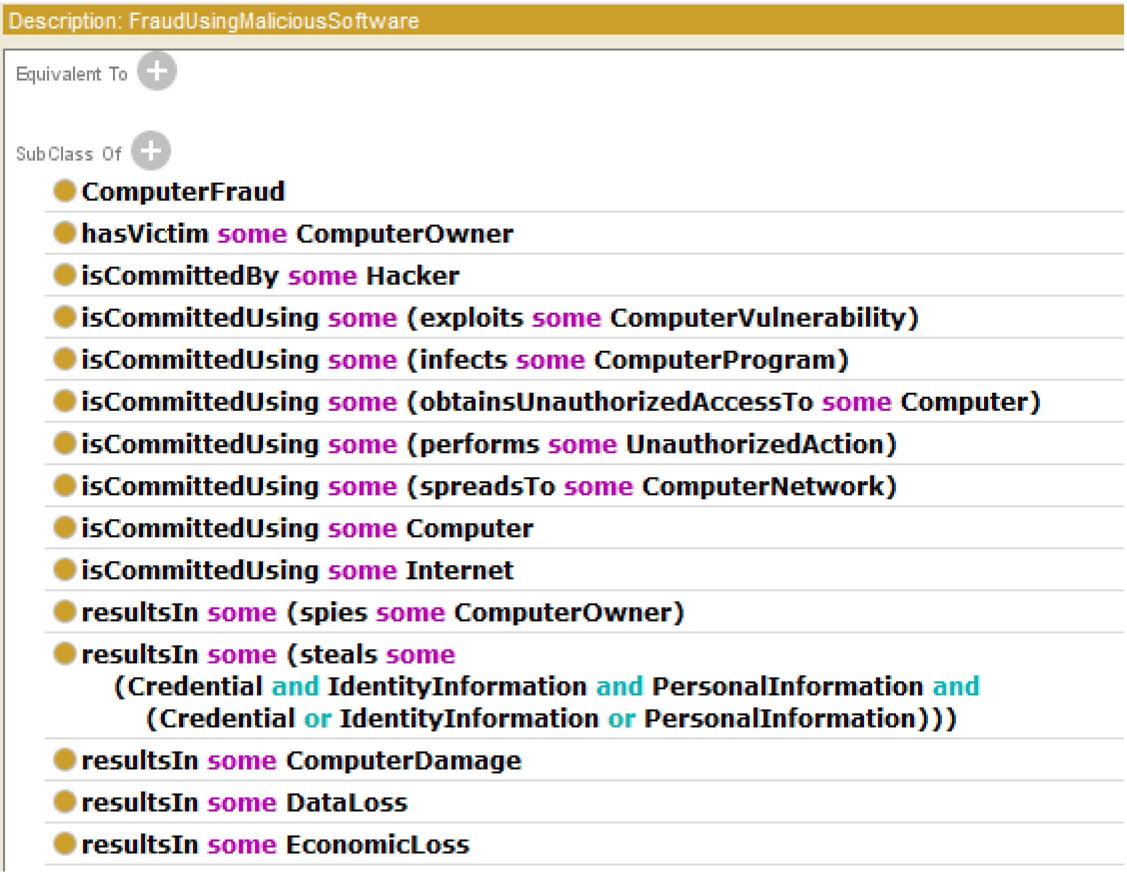
Equivalent To

SubClass Of

- CreditFraud
- hasVictim some (IssuingBank and Merchant and (IssuingBank or Merchant))
- isCommittedBy some Cardholder
- isCommittedUsing some ((not (hasIntentionToPay some Good)) and (receives some Good))
- isCommittedUsing some ((not (hasIntentionToPay some Service)) and (obtains some Service))
- isCommittedUsing some ((not (hasIntentionToPay some Transaction)) and (performs some Transaction))
- isCommittedUsing some (refuses some (isLegitimate some Transaction))
- isCommittedUsing some (usesFalse some Claim)
- resultsIn some EconomicLoss

Figure 59: Semantics of Friendly Fraud

Fraud using Malicious Software



Description: FraudUsingMaliciousSoftware

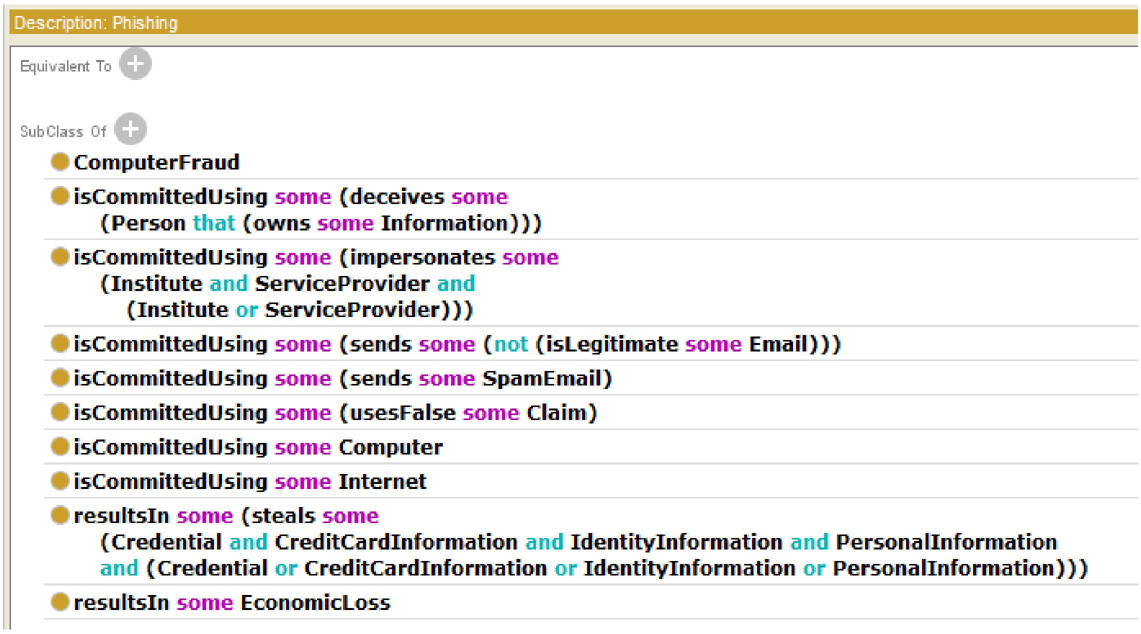
Equivalent To

SubClass Of

- ComputerFraud
- hasVictim some ComputerOwner
- isCommittedBy some Hacker
- isCommittedUsing some (exploits some ComputerVulnerability)
- isCommittedUsing some (infects some ComputerProgram)
- isCommittedUsing some (obtainsUnauthorizedAccessTo some Computer)
- isCommittedUsing some (performs some UnauthorizedAction)
- isCommittedUsing some (spreadsTo some ComputerNetwork)
- isCommittedUsing some Computer
- isCommittedUsing some Internet
- resultsIn some (spies some ComputerOwner)
- resultsIn some (steals some (Credential and IdentityInformation and PersonalInformation and (Credential or IdentityInformation or PersonalInformation)))
- resultsIn some ComputerDamage
- resultsIn some DataLoss
- resultsIn some EconomicLoss

Figure 60: Semantics of Fraud using Malicious Software

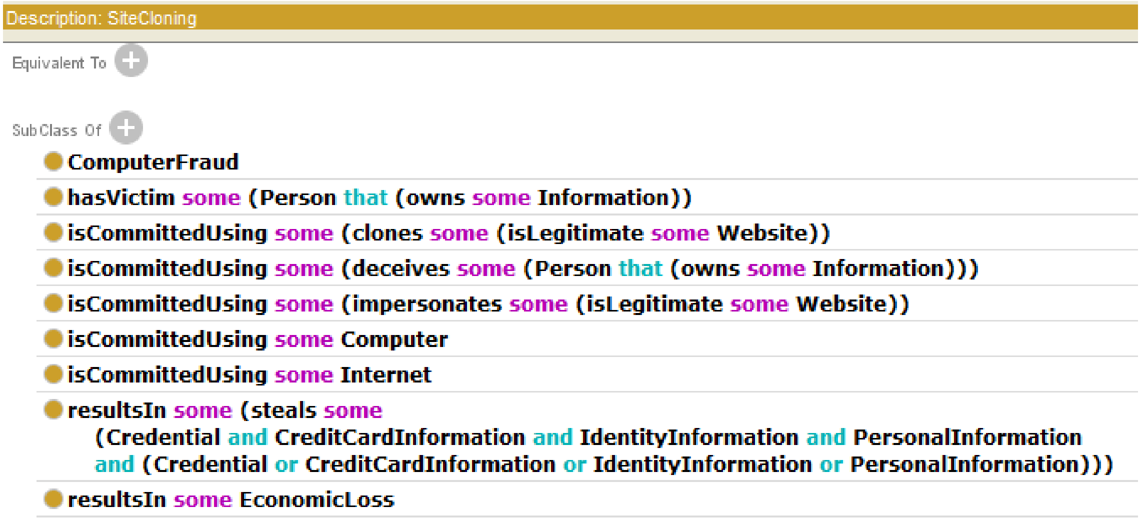
Phishing



```
Description: Phishing

Equivalent To

SubClass Of
● ComputerFraud
● isCommittedUsing some (deceives some
     (Person that (owns some Information)))
● isCommittedUsing some (impersonates some
     (Institute and ServiceProvider and
        (Institute or ServiceProvider)))
● isCommittedUsing some (sends some (not (isLegitimate some Email)))
● isCommittedUsing some (sends some SpamEmail)
● isCommittedUsing some (usesFalse some Claim)
● isCommittedUsing some Computer
● isCommittedUsing some Internet
● resultsIn some (steals some
     (Credential and CreditCardInformation and IdentityInformation and PersonalInformation
     and (Credential or CreditCardInformation or IdentityInformation or PersonalInformation)))
● resultsIn some EconomicLoss
```

Figure 61: Semantics of Phishing

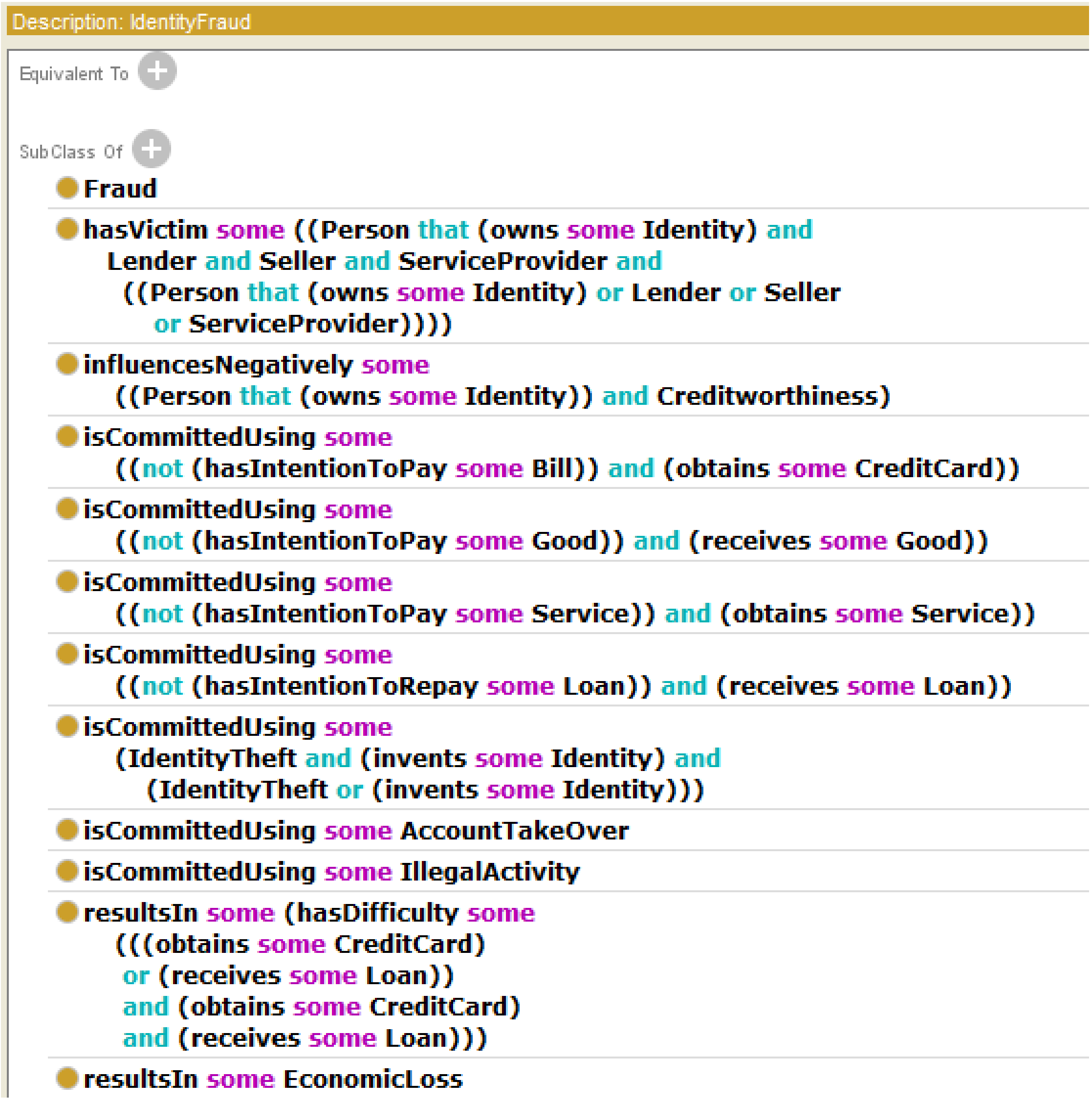
Site Cloning



```
Description: SiteCloning

Equivalent To

SubClass Of
● ComputerFraud
● hasVictim some (Person that (owns some Information))
● isCommittedUsing some (clones some (isLegitimate some Website))
● isCommittedUsing some (deceives some (Person that (owns some Information)))
● isCommittedUsing some (impersonates some (isLegitimate some Website))
● isCommittedUsing some Computer
● isCommittedUsing some Internet
● resultsIn some (steals some
     (Credential and CreditCardInformation and IdentityInformation and PersonalInformation
     and (Credential or CreditCardInformation or IdentityInformation or PersonalInformation)))
● resultsIn some EconomicLoss
```

Figure 62: Semantics of Site Cloning

Identity Fraud



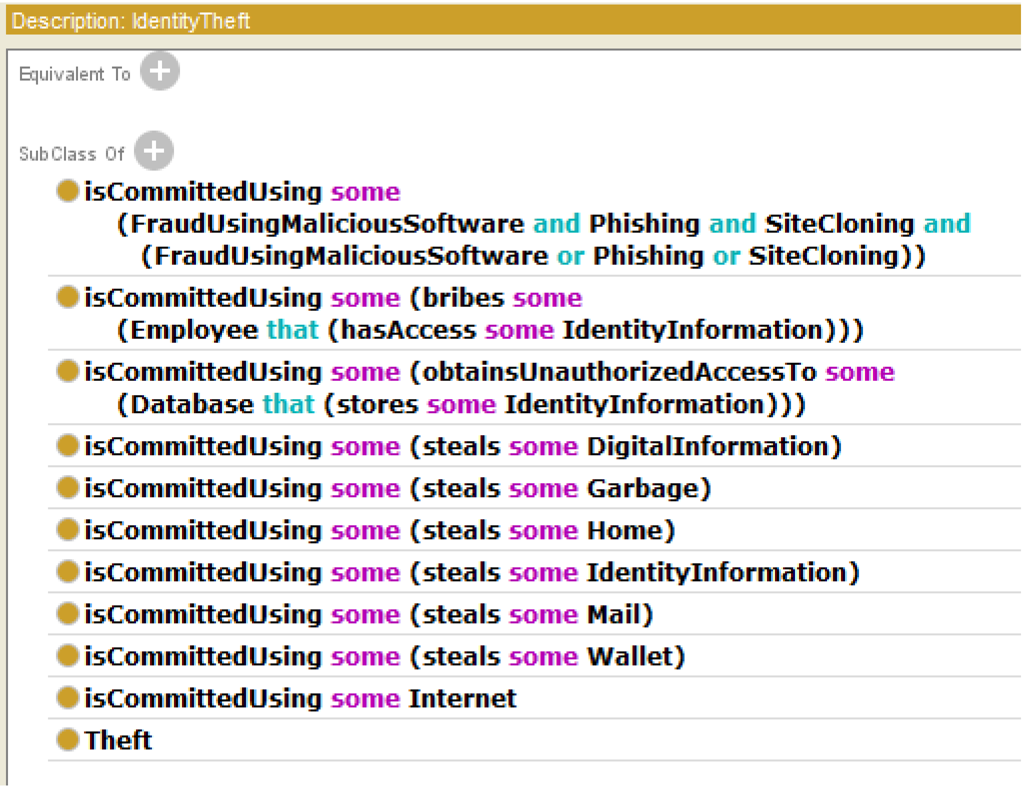
Description: IdentityFraud

Equivalent To ⊕

SubClass Of ⊕

```
● Fraud
● hasVictim some ((Person that (owns some Identity) and
    Lender and Seller and ServiceProvider and
      ((Person that (owns some Identity) or Lender or Seller
          or ServiceProvider))))
● influencesNegatively some
     ((Person that (owns some Identity)) and Creditworthiness)
● isCommittedUsing some
     ((not (hasIntentionToPay some Bill)) and (obtains some CreditCard))
● isCommittedUsing some
     ((not (hasIntentionToPay some Good)) and (receives some Good))
● isCommittedUsing some
     ((not (hasIntentionToPay some Service)) and (obtains some Service))
● isCommittedUsing some
     ((not (hasIntentionToRepay some Loan)) and (receives some Loan))
● isCommittedUsing some
     (IdentityTheft and (invents some Identity) and
         (IdentityTheft or (invents some Identity)))
● isCommittedUsing some AccountTakeOver
● isCommittedUsing some IllegalActivity
● resultsIn some (hasDifficulty some
     (((obtains some CreditCard)
      or (receives some Loan))
      and (obtains some CreditCard)
      and (receives some Loan)))
● resultsIn some EconomicLoss
```

Figure 63: Semantics of Identity Fraud



Description: IdentityTheft

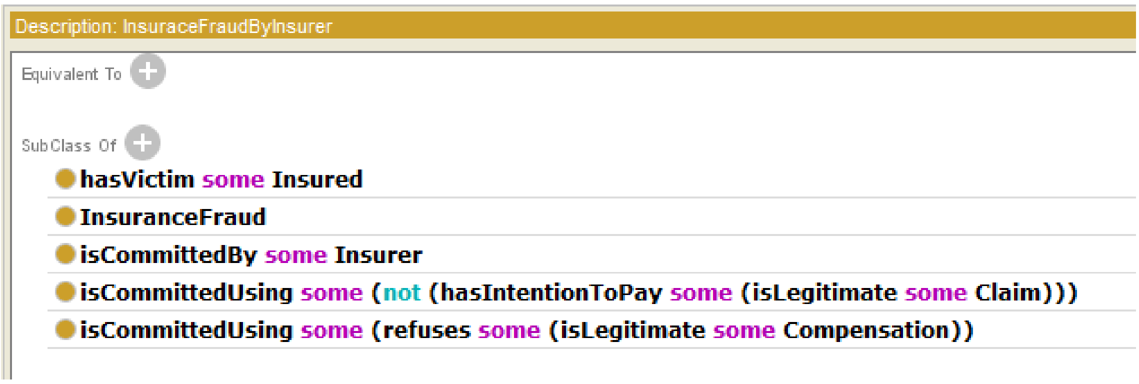
Equivalent To ⊕

SubClass Of ⊕

- ● isCommittedUsing some
  (FraudUsingMaliciousSoftware and Phishing and SiteCloning and
  (FraudUsingMaliciousSoftware or Phishing or SiteCloning))
- ● isCommittedUsing some (bribes some
  (Employee that (hasAccess some IdentityInformation)))
- ● isCommittedUsing some (obtainsUnauthorizedAccessTo some
  (Database that (stores some IdentityInformation)))
- ● isCommittedUsing some (steals some DigitalInformation)
- ● isCommittedUsing some (steals some Garbage)
- ● isCommittedUsing some (steals some Home)
- ● isCommittedUsing some (steals some IdentityInformation)
- ● isCommittedUsing some (steals some Mail)
- ● isCommittedUsing some (steals some Wallet)
- ● isCommittedUsing some Internet
- ● Theft

Figure 64: Semantics of Identity Theft

Insurance Fraud by Insurer



Description: InsuraceFraudByInsurer

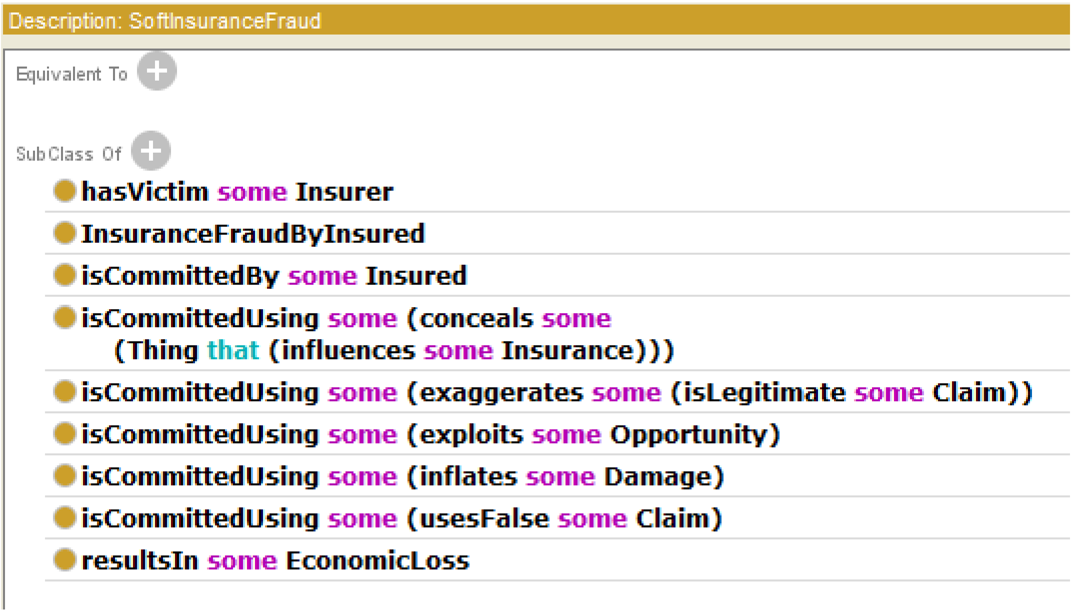
Equivalent To ⊕

SubClass Of ⊕

- ● hasVictim some Insured
- ● InsuranceFraud
- ● isCommittedBy some Insurer
- ● isCommittedUsing some (not (hasIntentionToPay some (isLegitimate some Claim)))
- ● isCommittedUsing some (refuses some (isLegitimate some Compensation))

Figure 65: Semantics of Insurance Fraud by Insurer

Insurance Fraud by Insured



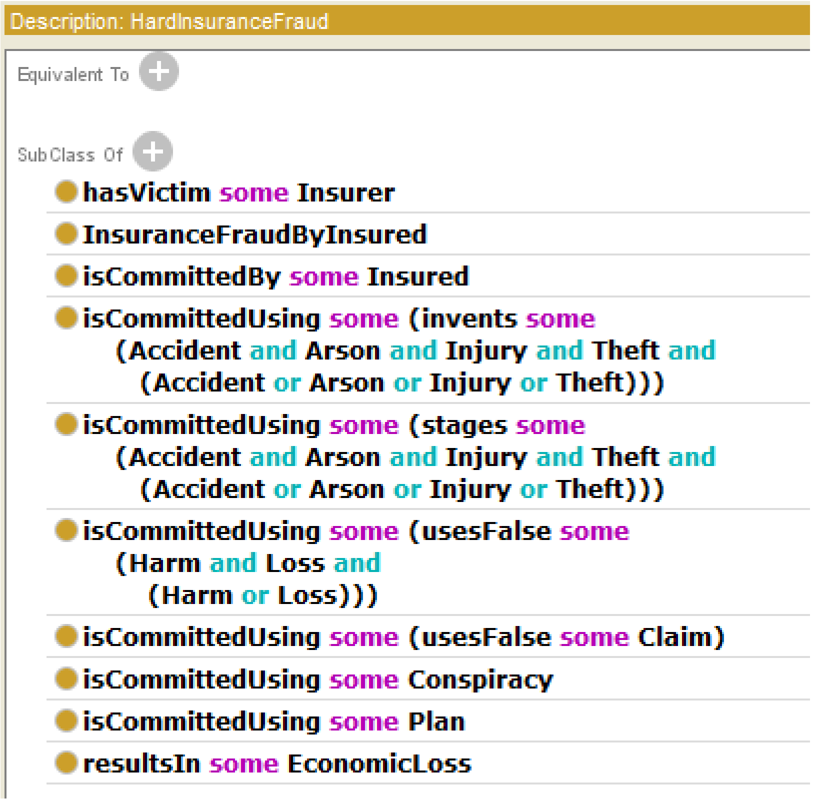
Description: SoftInsuranceFraud

Equivalent To

SubClass Of

- hasVictim some Insurer
- InsuranceFraudByInsured
- isCommittedBy some Insured
- isCommittedUsing some (conceals some (Thing that (influences some Insurance)))
- isCommittedUsing some (exaggerates some (isLegitimate some Claim))
- isCommittedUsing some (exploits some Opportunity)
- isCommittedUsing some (inflates some Damage)
- isCommittedUsing some (usesFalse some Claim)
- resultsIn some EconomicLoss

Figure 66: Semantics of Soft Insurance Fraud



Description: HardInsuranceFraud

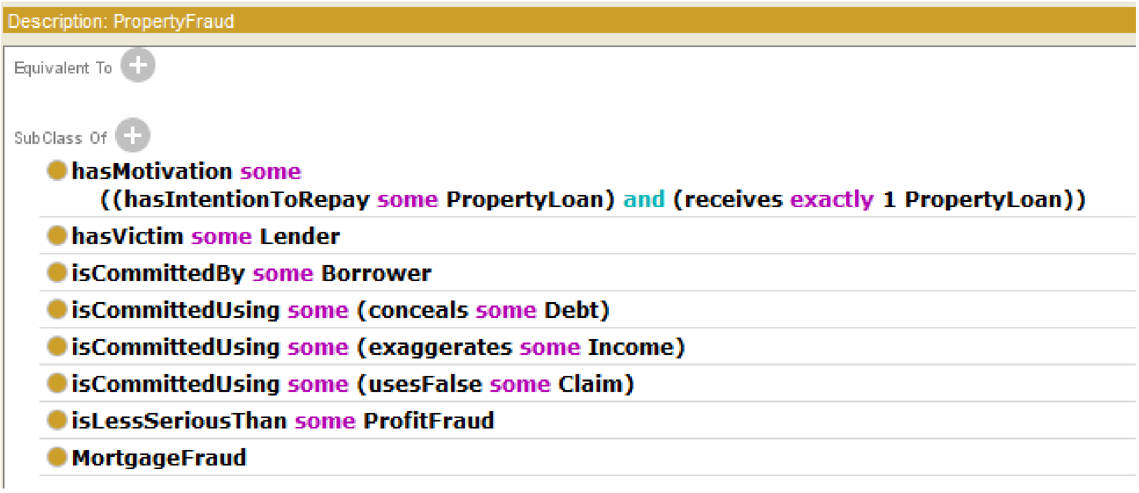
Equivalent To

SubClass Of

- hasVictim some Insurer
- InsuranceFraudByInsured
- isCommittedBy some Insured
- isCommittedUsing some (invents some (Accident and Arson and Injury and Theft and (Accident or Arson or Injury or Theft)))
- isCommittedUsing some (stages some (Accident and Arson and Injury and Theft and (Accident or Arson or Injury or Theft)))
- isCommittedUsing some (usesFalse some (Harm and Loss and (Harm or Loss)))
- isCommittedUsing some (usesFalse some Claim)
- isCommittedUsing some Conspiracy
- isCommittedUsing some Plan
- resultsIn some EconomicLoss

Figure 67: Semantics of Hard Insurance Fraud

Property Fraud



Description: PropertyFraud

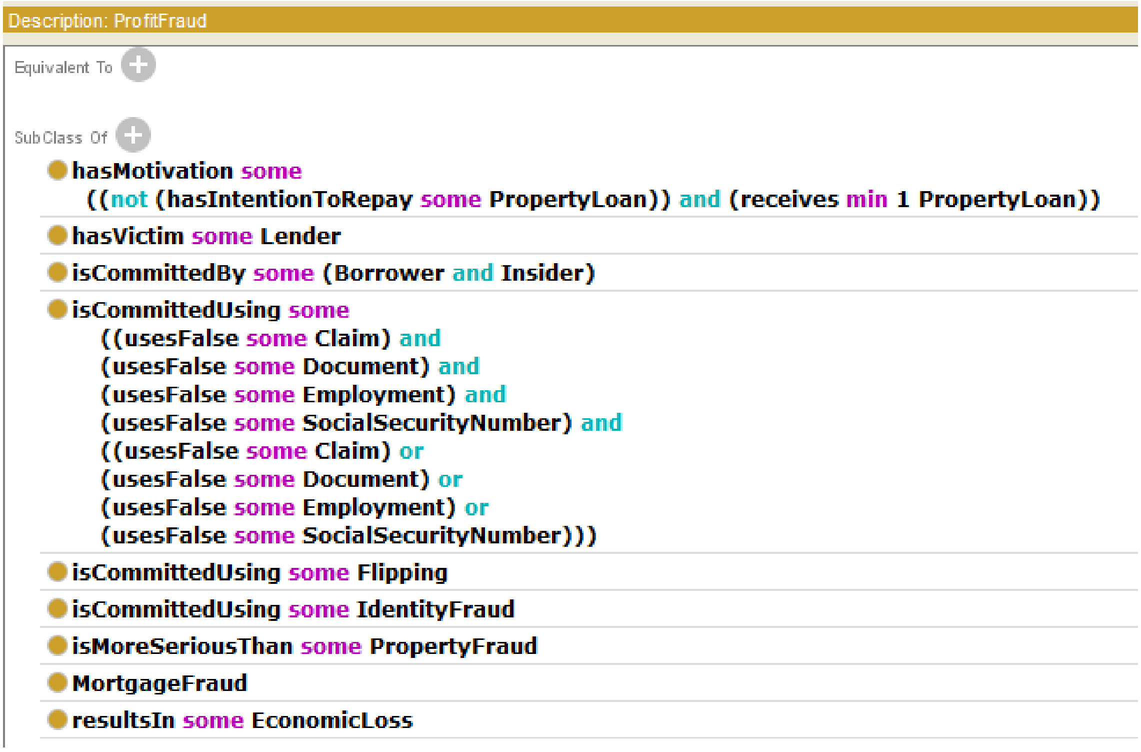
Equivalent To

SubClass Of

- hasMotivation some
  ((hasIntentionToRepay some PropertyLoan) and (receives exactly 1 PropertyLoan))
- hasVictim some Lender
- isCommittedBy some Borrower
- isCommittedUsing some (conceals some Debt)
- isCommittedUsing some (exaggerates some Income)
- isCommittedUsing some (usesFalse some Claim)
- isLessSeriousThan some ProfitFraud
- MortgageFraud

Figure 68: Semantics of Property Fraud

Profit Fraud



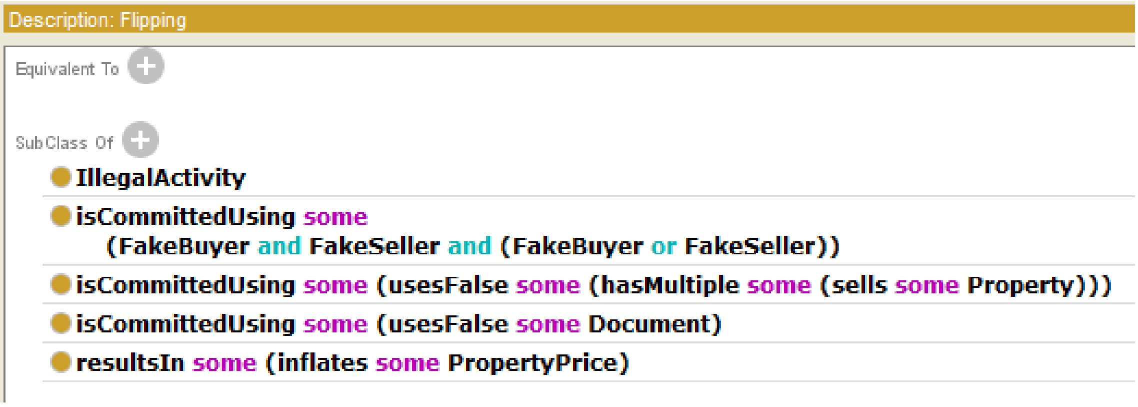
Description: ProfitFraud

Equivalent To

SubClass Of

- hasMotivation some
  ((not (hasIntentionToRepay some PropertyLoan)) and (receives min 1 PropertyLoan))
- hasVictim some Lender
- isCommittedBy some (Borrower and Insider)
- isCommittedUsing some
  ((usesFalse some Claim) and
  (usesFalse some Document) and
  (usesFalse some Employment) and
  (usesFalse some SocialSecurityNumber) and
  ((usesFalse some Claim) or
  (usesFalse some Document) or
  (usesFalse some Employment) or
  (usesFalse some SocialSecurityNumber)))
- isCommittedUsing some Flipping
- isCommittedUsing some IdentityFraud
- isMoreSeriousThan some PropertyFraud
- MortgageFraud
- resultsIn some EconomicLoss

Figure 69: Semantics of Profit Fraud



Description: Flipping

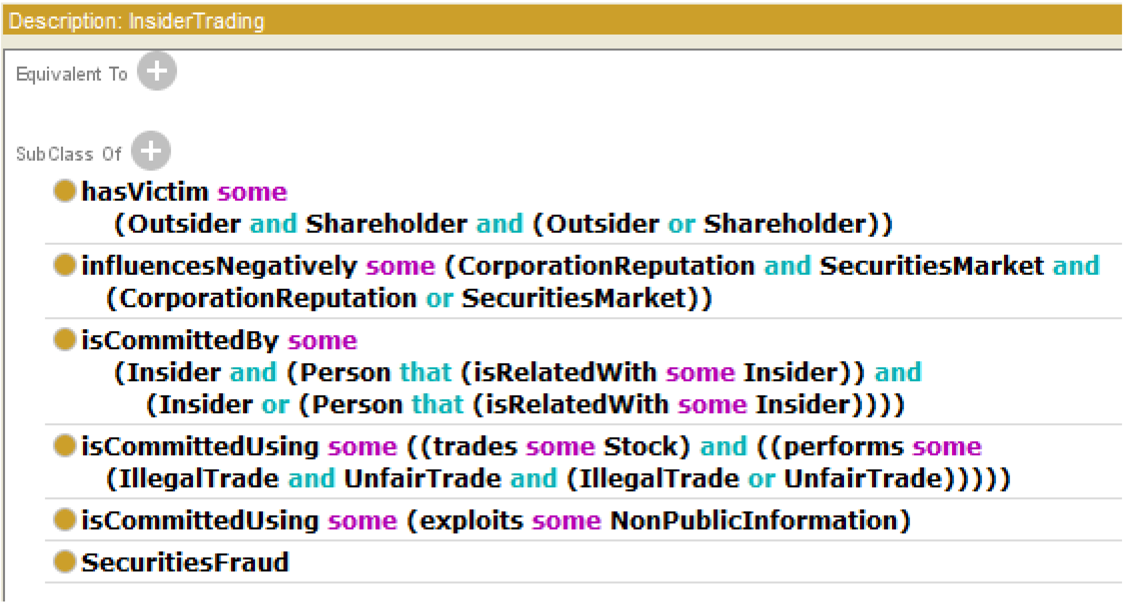
Equivalent To

SubClass Of

- IllegalActivity
- isCommittedUsing some (FakeBuyer and FakeSeller and (FakeBuyer or FakeSeller))
- isCommittedUsing some (usesFalse some (hasMultiple some (sells some Property)))
- isCommittedUsing some (usesFalse some Document)
- resultsIn some (inflates some PropertyPrice)

Figure 70: Semantics of Flipping

Insider Trading



Description: InsiderTrading

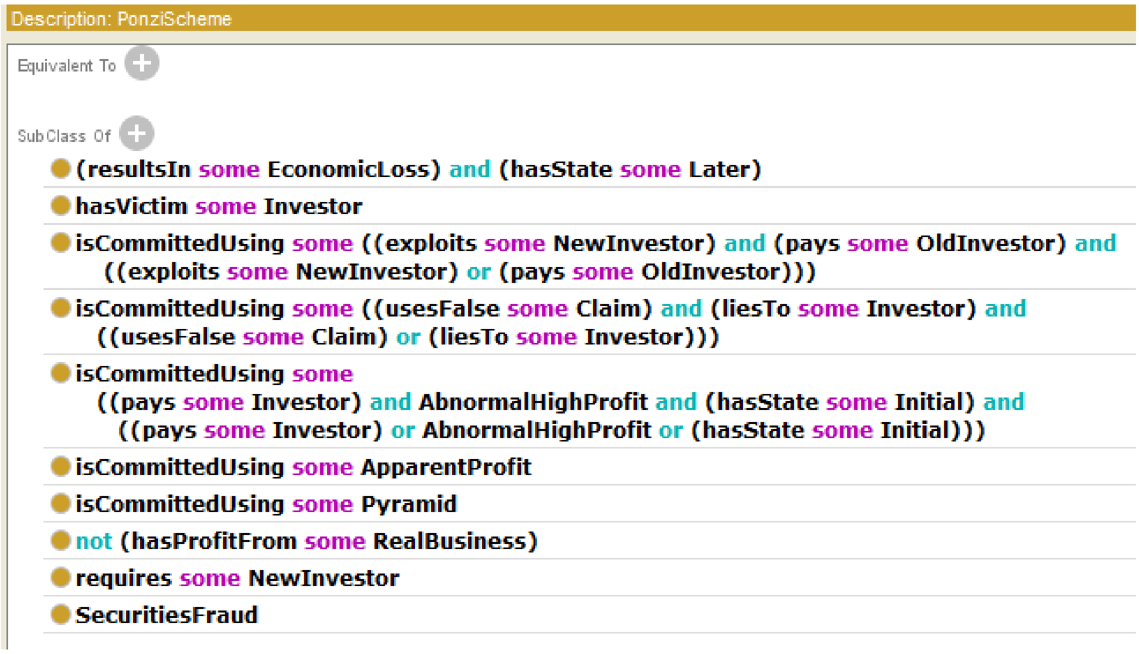
Equivalent To

SubClass Of

- hasVictim some (Outsider and Shareholder and (Outsider or Shareholder))
- influencesNegatively some (CorporationReputation and SecuritiesMarket and (CorporationReputation or SecuritiesMarket))
- isCommittedBy some (Insider and (Person that (isRelatedWith some Insider)) and (Insider or (Person that (isRelatedWith some Insider))))
- isCommittedUsing some ((trades some Stock) and ((performs some (IllegalTrade and UnfairTrade and (IllegalTrade or UnfairTrade)))))
- isCommittedUsing some (exploits some NonPublicInformation)
- SecuritiesFraud

Figure 71: Semantics of Insider Trading

Ponzi Scheme



Description: PonziScheme

Equivalent To +

SubClass Of +

● (resultsIn some EconomicLoss) and (hasState some Later)

● hasVictim some Investor

● isCommittedUsing some ((exploits some NewInvestor) and (pays some OldInvestor) and ((exploits some NewInvestor) or (pays some OldInvestor)))

● isCommittedUsing some ((usesFalse some Claim) and (liesTo some Investor) and ((usesFalse some Claim) or (liesTo some Investor)))

● isCommittedUsing some ((pays some Investor) and AbnormalHighProfit and (hasState some Initial) and ((pays some Investor) or AbnormalHighProfit or (hasState some Initial)))

● isCommittedUsing some ApparentProfit

● isCommittedUsing some Pyramid

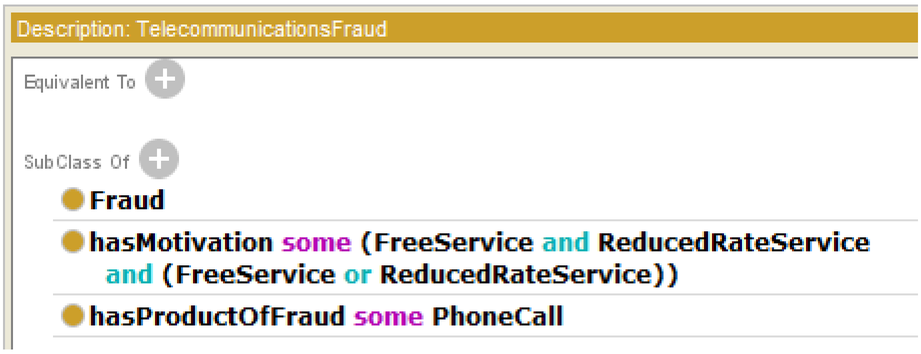
● not (hasProfitFrom some RealBusiness)

● requires some NewInvestor

● SecuritiesFraud

Figure 72: Semantics of Ponzi Scheme

Telecommunications Fraud



Description: TelecommunicationsFraud

Equivalent To +

SubClass Of +

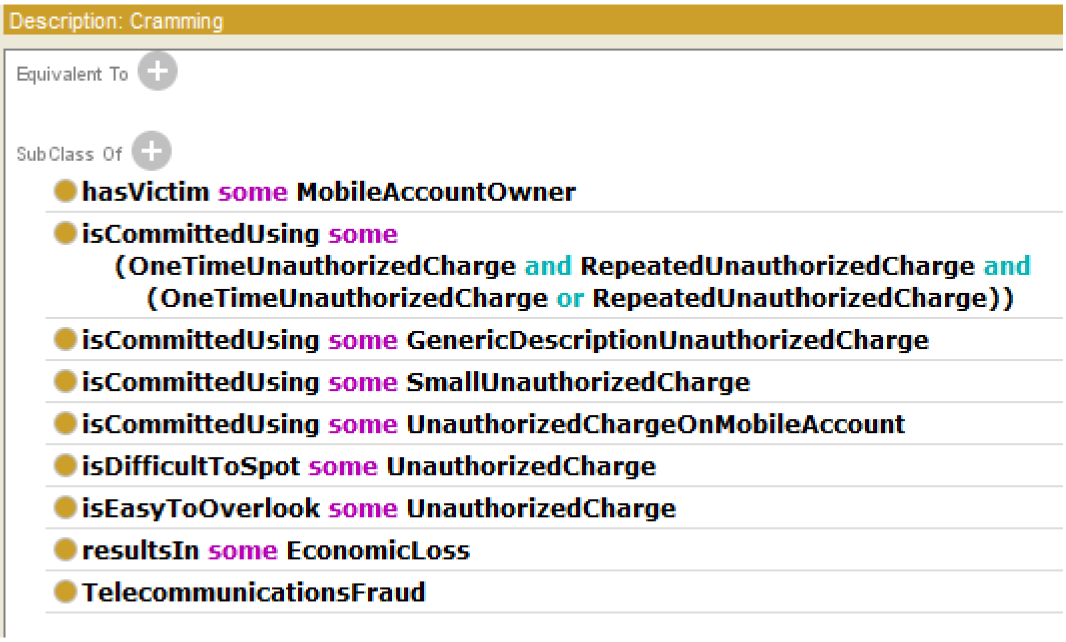
● Fraud

● hasMotivation some (FreeService and ReducedRateService and (FreeService or ReducedRateService))

● hasProductOfFraud some PhoneCall

Figure 73: Semantics of Telecommunications Fraud

Cramming



Description: Cramming

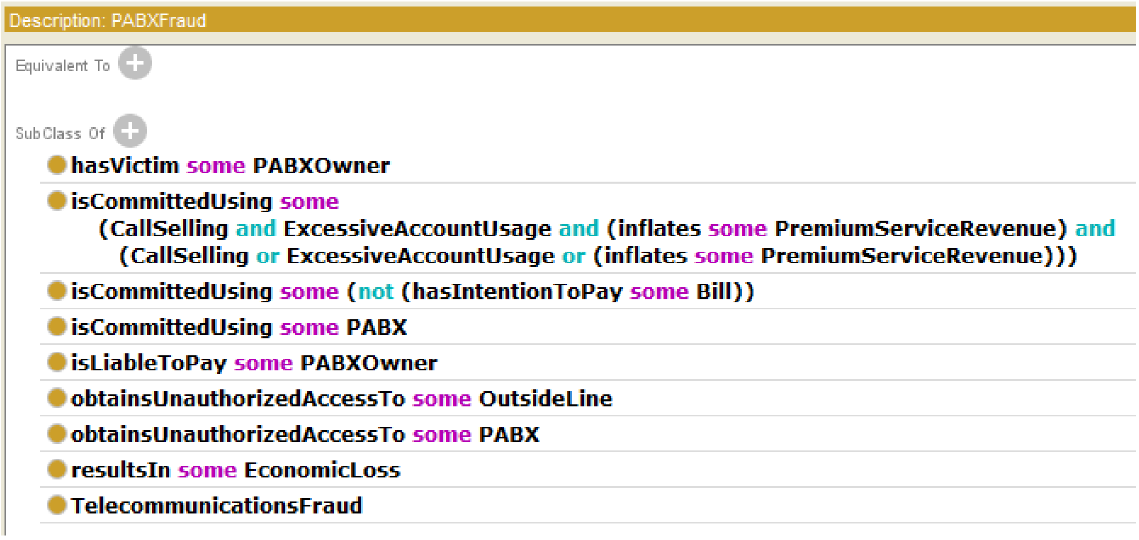
Equivalent To

SubClass Of

- hasVictim some MobileAccountOwner
- isCommittedUsing some
  (OneTimeUnauthorizedCharge and RepeatedUnauthorizedCharge and
  (OneTimeUnauthorizedCharge or RepeatedUnauthorizedCharge))
- isCommittedUsing some GenericDescriptionUnauthorizedCharge
- isCommittedUsing some SmallUnauthorizedCharge
- isCommittedUsing some UnauthorizedChargeOnMobileAccount
- isDifficultToSpot some UnauthorizedCharge
- isEasyToOverlook some UnauthorizedCharge
- resultsIn some EconomicLoss
- TelecommunicationsFraud

Figure 74: Semantics of Cramming

PABX Fraud



Description: PABXFraud

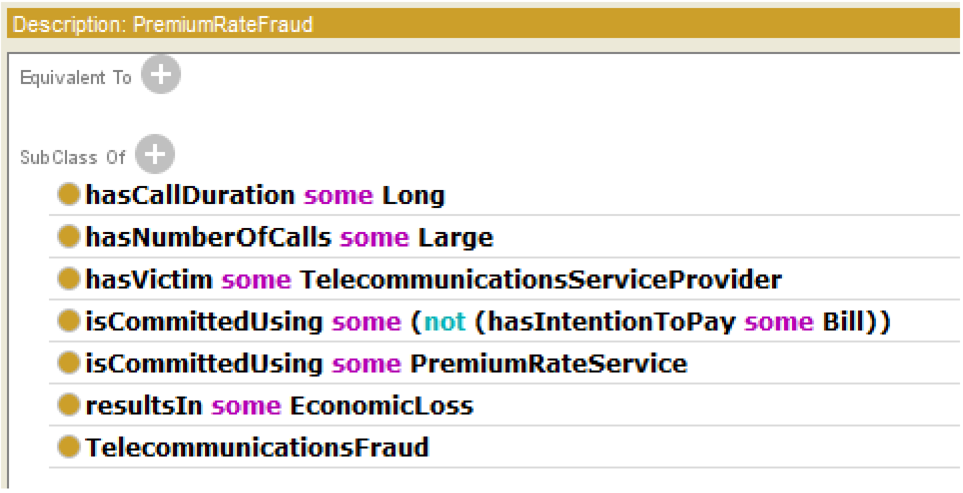
Equivalent To

SubClass Of

- hasVictim some PABXOwner
- isCommittedUsing some
  (CallSelling and ExcessiveAccountUsage and (inflates some PremiumServiceRevenue) and
  (CallSelling or ExcessiveAccountUsage or (inflates some PremiumServiceRevenue)))
- isCommittedUsing some (not (hasIntentionToPay some Bill))
- isCommittedUsing some PABX
- isLiableToPay some PABXOwner
- obtainsUnauthorizedAccessTo some OutsideLine
- obtainsUnauthorizedAccessTo some PABX
- resultsIn some EconomicLoss
- TelecommunicationsFraud

Figure 75: Semantics of PABX Fraud

Premium Rate Fraud



Description: PremiumRateFraud

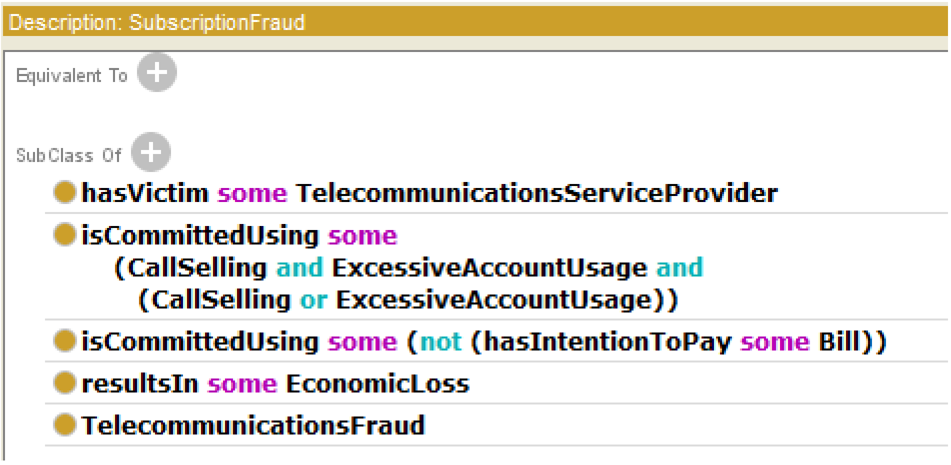
Equivalent To ⊕

SubClass Of ⊕

- hasCallDuration some Long
- hasNumberOfCalls some Large
- hasVictim some TelecommunicationsServiceProvider
- isCommittedUsing some (not (hasIntentionToPay some Bill))
- isCommittedUsing some PremiumRateService
- resultsIn some EconomicLoss
- TelecommunicationsFraud

Figure 76: Semantics of Premium Rate Fraud

Subscription Fraud



Description: SubscriptionFraud

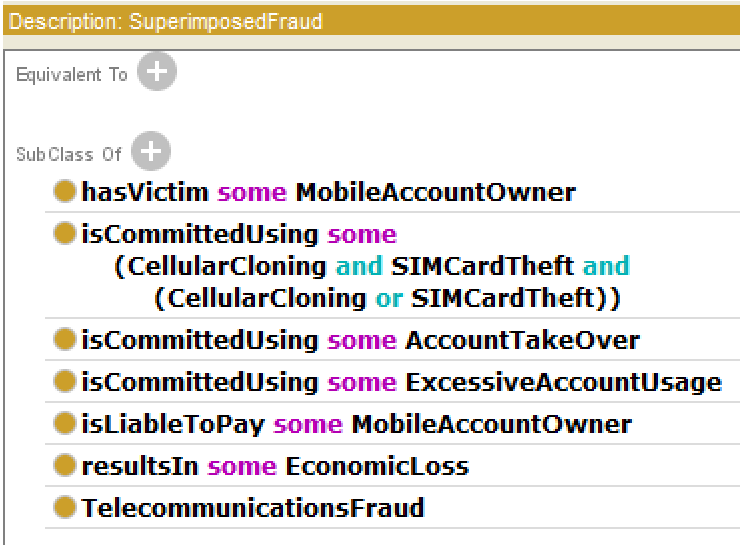
Equivalent To ⊕

SubClass Of ⊕

- hasVictim some TelecommunicationsServiceProvider
- isCommittedUsing some
  (CallSelling and ExcessiveAccountUsage and
  (CallSelling or ExcessiveAccountUsage))
- isCommittedUsing some (not (hasIntentionToPay some Bill))
- resultsIn some EconomicLoss
- TelecommunicationsFraud

Figure 77: Semantics of Subscription Fraud

Superimposed Fraud



Description: SuperimposedFraud

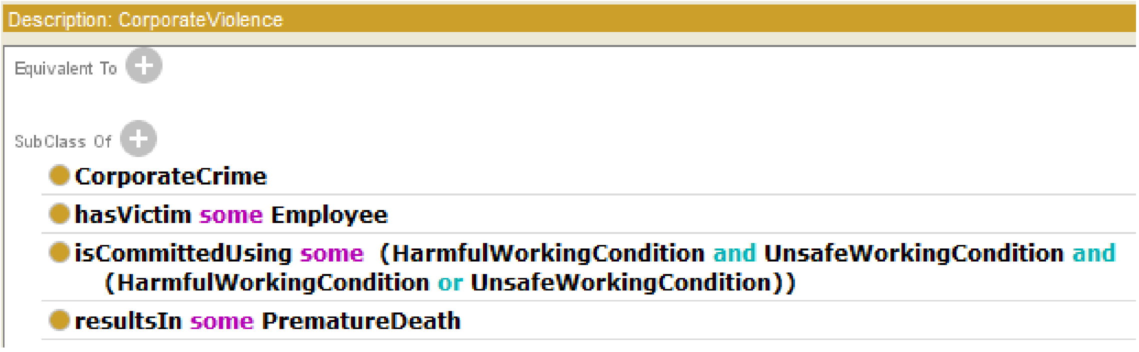
Equivalent To ⊕

SubClass Of ⊕

- ● hasVictim some MobileAccountOwner
- ● isCommittedUsing some (CellularCloning and SIMCardTheft and (CellularCloning or SIMCardTheft))
- ● isCommittedUsing some AccountTakeOver
- ● isCommittedUsing some ExcessiveAccountUsage
- ● isLiableToPay some MobileAccountOwner
- ● resultsIn some EconomicLoss
- ● TelecommunicationsFraud

Figure 78: Semantics of Superimposed Fraud

Corporate Violence



Description: CorporateViolence

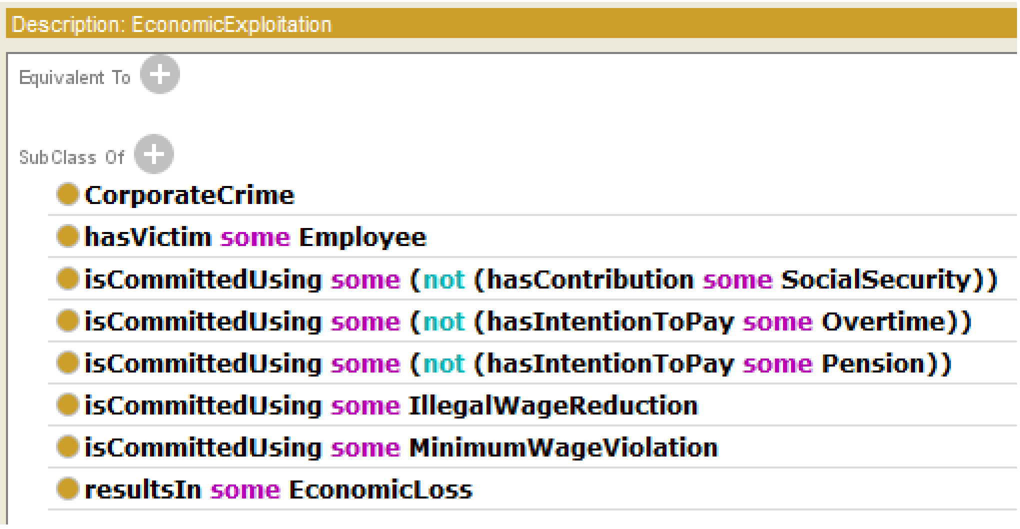
Equivalent To ⊕

SubClass Of ⊕

- ● CorporateCrime
- ● hasVictim some Employee
- ● isCommittedUsing some (HarmfulWorkingCondition and UnsafeWorkingCondition and (HarmfulWorkingCondition or UnsafeWorkingCondition))
- ● resultsIn some PrematureDeath

Figure 79: Semantics of Corporate Violence

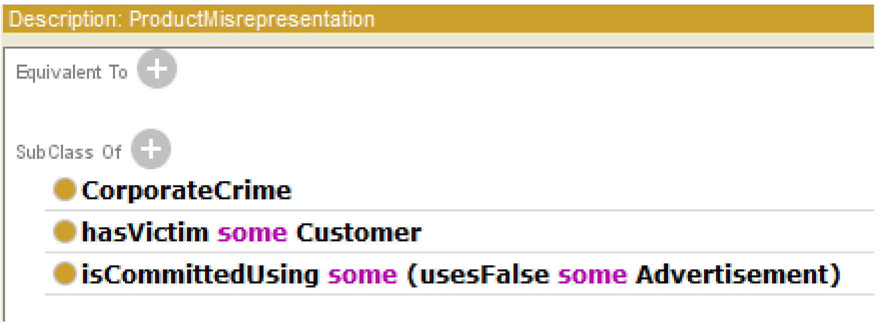
Economic Exploitation



```
Description: EconomicExploitation

Equivalent To +

SubClass Of +
    ● CorporateCrime
    ● hasVictim some Employee
    ● isCommittedUsing some (not (hasContribution some SocialSecurity))
    ● isCommittedUsing some (not (hasIntentionToPay some Overtime))
    ● isCommittedUsing some (not (hasIntentionToPay some Pension))
    ● isCommittedUsing some IllegalWageReduction
    ● isCommittedUsing some MinimumWageViolation
    ● resultsIn some EconomicLoss
```

Figure 80: Semantics of Economic Exploitation

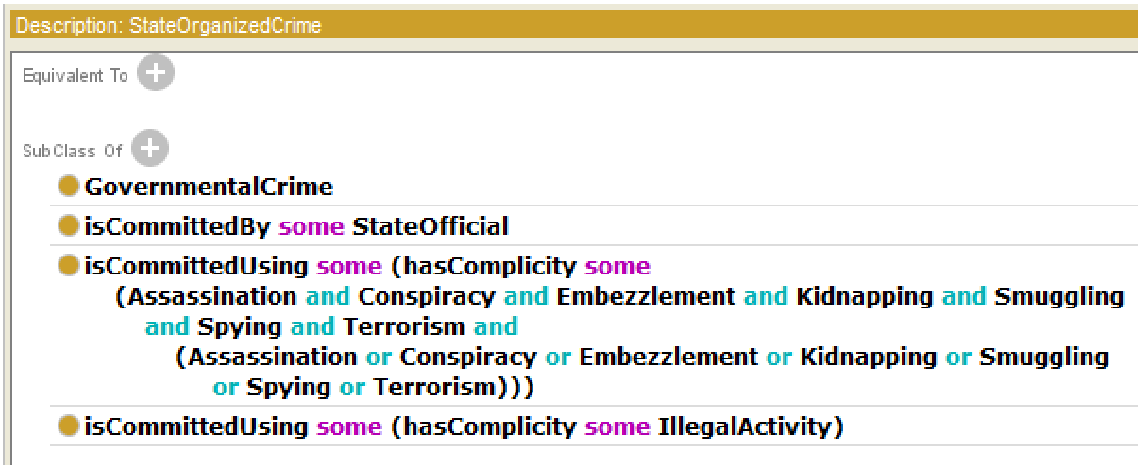
Product Misrepresentation



```
Description: ProductMisrepresentation

Equivalent To +

SubClass Of +
    ● CorporateCrime
    ● hasVictim some Customer
    ● isCommittedUsing some (usesFalse some Advertisement)
```

Figure 81: Semantics of Product Misrepresentation

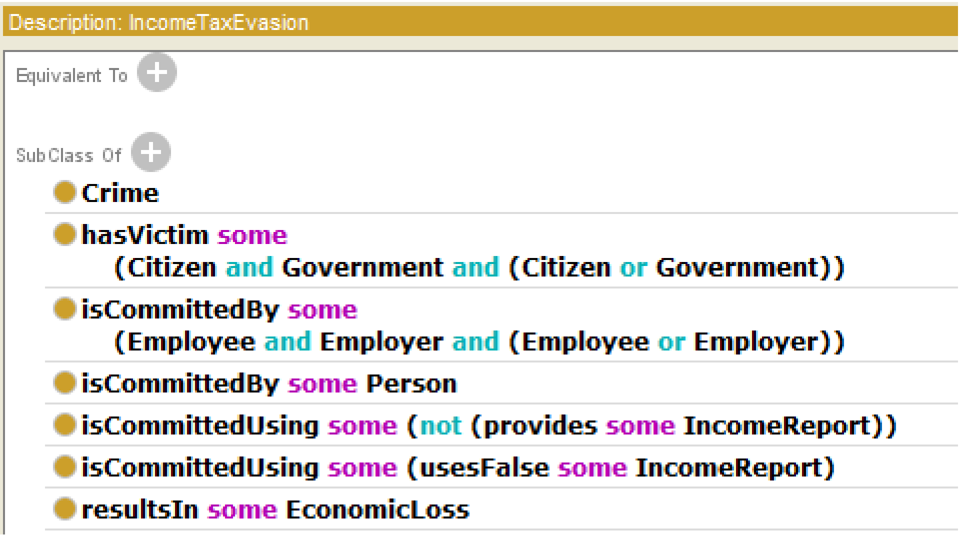
State-organized Crime



```
Description: StateOrganizedCrime

Equivalent To +

SubClass Of +
    ● GovernmentalCrime
    ● isCommittedBy some StateOfficial
    ● isCommittedUsing some (hasComplicity some
          (Assassination and Conspiracy and Embezzlement and Kidnapping and Smuggling
            and Spying and Terrorism and
              (Assassination or Conspiracy or Embezzlement or Kidnapping or Smuggling
               or Spying or Terrorism)))
    ● isCommittedUsing some (hasComplicity some IllegalActivity)
```

Figure 82: Semantics of State-organized Crime

Income Tax Evasion



Description: IncomeTaxEvasion

Equivalent To ⊕

SubClass Of ⊕

● Crime

● hasVictim some
(Citizen and Government and (Citizen or Government))

● isCommittedBy some
(Employee and Employer and (Employee or Employer))

● isCommittedBy some Person

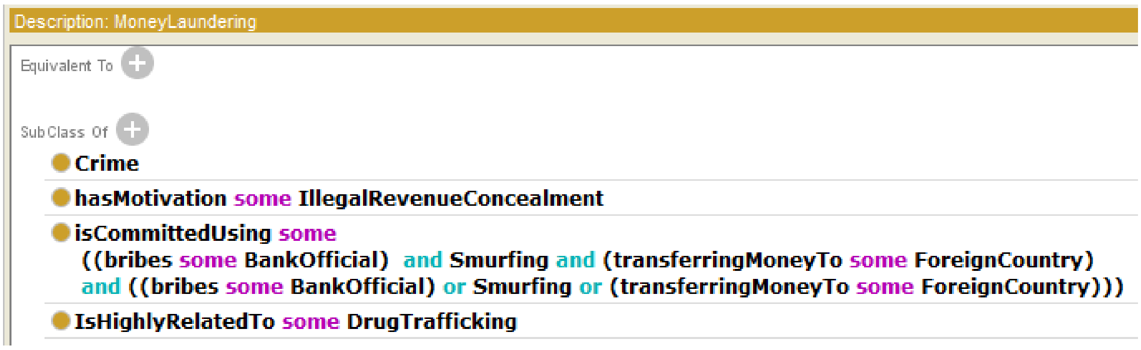
● isCommittedUsing some (not (provides some IncomeReport))

● isCommittedUsing some (usesFalse some IncomeReport)

● resultsIn some EconomicLoss

Figure 83: Semantics of Income Tax Evasion

Money Laundering



Description: MoneyLaundering

Equivalent To ⊕

SubClass Of ⊕

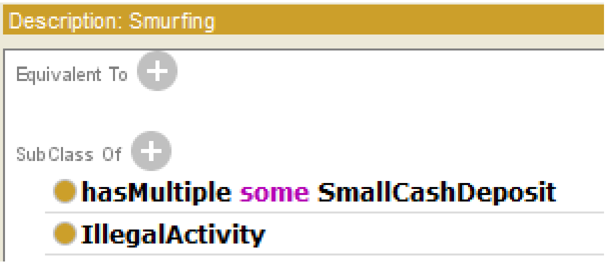
● Crime

● hasMotivation some IllegalRevenueConcealment

● isCommittedUsing some
((bribes some BankOfficial) and Smurfing and (transferringMoneyTo some ForeignCountry)
and ((bribes some BankOfficial) or Smurfing or (transferringMoneyTo some ForeignCountry)))

● IsHighlyRelatedTo some DrugTrafficking

Figure 84: Semantics of Money Laundering



Description: Smurfing

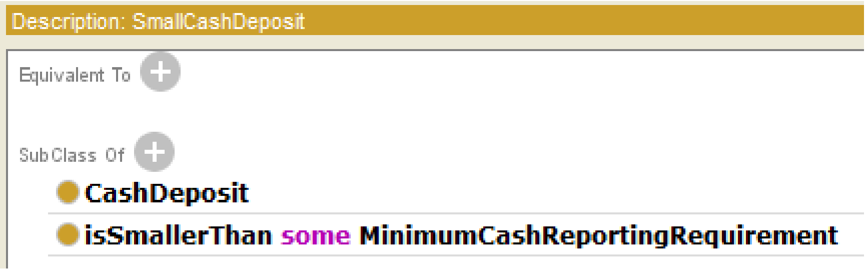
Equivalent To ⊕

SubClass Of ⊕

● hasMultiple some SmallCashDeposit

● IllegalActivity

Figure 85: Semantics of Smurfing



Description: SmallCashDeposit

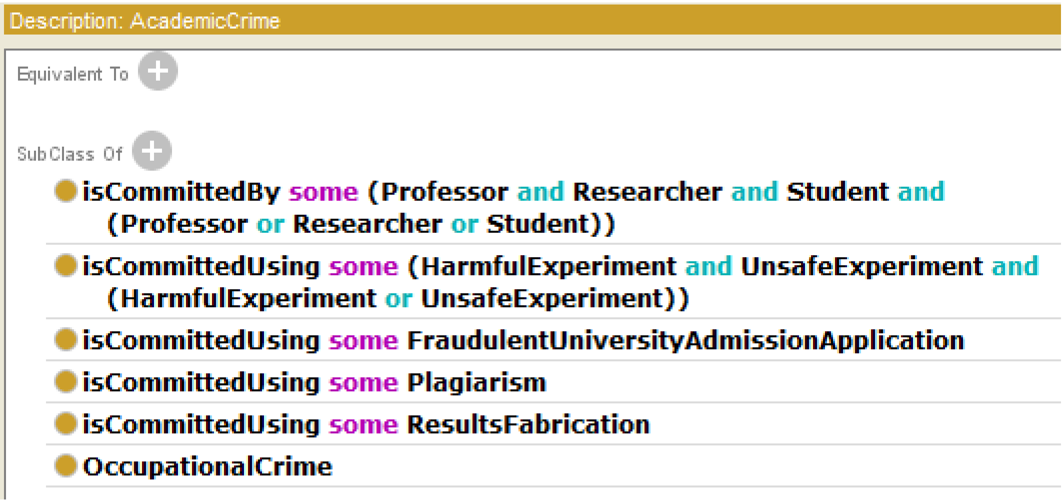
Equivalent To

SubClass Of

- CashDeposit
- isSmallerThan some MinimumCashReportingRequirement

Figure 86: Semantics of Small Cash Deposit

Academic Crime



Description: AcademicCrime

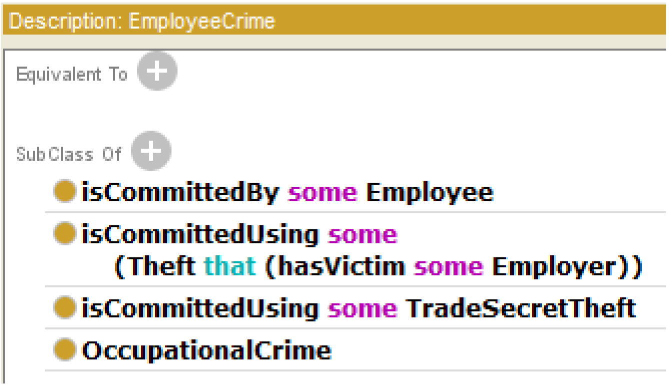
Equivalent To

SubClass Of

- isCommittedBy some (Professor and Researcher and Student and (Professor or Researcher or Student))
- isCommittedUsing some (HarmfulExperiment and UnsafeExperiment and (HarmfulExperiment or UnsafeExperiment))
- isCommittedUsing some FraudulentUniversityAdmissionApplication
- isCommittedUsing some Plagiarism
- isCommittedUsing some ResultsFabrication
- OccupationalCrime

Figure 87: Semantics of Academic Crime

Employee Crime



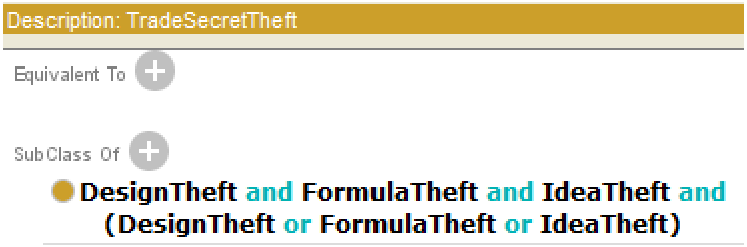
Description: EmployeeCrime

Equivalent To

SubClass Of

- isCommittedBy some Employee
- isCommittedUsing some (Theft that (hasVictim some Employer))
- isCommittedUsing some TradeSecretTheft
- OccupationalCrime

Figure 88: Semantics of Employee Crime



Description: TradeSecretTheft

Equivalent To

SubClass Of

● DesignTheft and FormulaTheft and IdeaTheft and (DesignTheft or FormulaTheft or IdeaTheft)

Figure 89: Semantics of Trade Secret Theft