Master Computer Science

An Exploration of Multiple Unsupervised Algorithms in a Financial Auditing Context

Name:	Alban Bastiaan
Student ID:	s1695495
Date:	01/10/2019
Specialisation:	Data Science
1st supervisor:	Dr. Wojtek Kowalczyk
2nd supervisor:	Dr. Bas van Stein

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

An Exploration of Multiple Unsupervised Algorithms in a Financial Auditing Context

October 30, 2019

Abstract

This paper concerns an exploration of the combination of two distinct fields: accounting and data science. The focus of this paper is the usage of unsupervised outlier detection for frauds and errors within an accounting context. Unique synthetic data are used from a digital simulation which simulates a company with many business transactions. The simulation creates the possibility to verify the functioning of the unsupervised techniques with a ground truth target variable containing the real error and fraud cases inserted in the simulation. Two distinct situations are used: a simulation case where a relatively lower number of errors or fraud are present, and a simulation case where there are relatively more instances of error and/or fraud. Multiple unsupervised algorithms are explored. The results show the Vanilla Autoencoder and Contractive Autoencoder obtain the best performance over the different datasets and imbalance settings. Additional work is identified, among other aspects, in finding methods to overcome imbalancing problems with unsupervised learning.

1 Introduction

Providing assurance to the correctness and reliability of a companies financial statements is a central aspect of the work of a financial auditor. The goal of a financial auditor is to provide a judgement whether the financial statements are free of *material* errors or frauds. The financial auditor is, in a sense, the human form of an anomaly detector. The goal of this paper is to do a first exploration into multiple unsupervised algorithms within a financial auditing context.

The current procedures withing the auditing context used so far have had relatively lower amounts of innovation in the past 100 years. It is characterized by using basic sampling methods and manual work. Besides these observations it is a trivial fact that the bookkeeping which makes up the financial statements is characterized by highly structured data and larger numbers of records. Transactions of companies are labeled within a general ledger. This provides a rich number of categories to be exploited for algorithms. With the rising possibilities of more advanced data science possibilities, combined with all of the preceding observations, the usage of algorithms gives the opportunity to explore the usage of more modern innovations of unsupervised algorithms within a financial auditing context.

The problem which largely exists with exploring algorithms within a financial auditing context is that the confirmation of the performance is hard to do. This is due to two problems: (1) the unavailability of ground truths concerning the existence of frauds or not in real life data, (2) the unavailability of public data in general. To elaborate on (1): if real data are collected from a company there is no ground truth available on which transactions are true frauds or errors. These are almost never available for *all* transactions and/or it takes a large effort to confirm the anomalies (note: frauds and errors) found. To elaborate on (2) another problem which exists is that public data are not available, since businesses don't want to provide this data for privacy reasons. This blocks the possibility of doing strong research. For the purposes of this paper a unique dataset is provided, which provides synthetic data from an audit simulator. This dataset simulates a real life company, and also provides ground truths for which transactions are frauds and errors. Related research is conducted in financially related fields such as credit card fraud or is conducted with supervised models[7]. This provides a unique opportunity to apply algorithms within a financial auditing context.

1.1 Research Questions and contributions

Using unsupervised algorithms within a financial auditing environment is challenging. This is mainly related to the intersection of regulation between for instance the "International Auditing Standards" (IAS), external reporting such as "International Financial Reporting Standards" (IFRS) and multiple fiscal considerations. Knowledge about these regulations is necessary to (1) be able to know what transaction are considered fraudulous or errors, (2) but also sets up the limitations of the financial auditor in what kind of natural authorizations the auditor has to perform his work. Exploring the usage of unsupervised algorithms must take the previous aspects into consideration. The main research question of this paper is:

"To what extent can unsupervised algorithms detect financial frauds or errors in a financial auditing context?"

The sub questions of this paper are:

1. Which algorithms can be taken into consideration to use, and why?

2. How do the algorithms perform across multiple evaluation measures?

3. Can a robustness of results be found per algorithm using different fraud and error contexts? Note that the third question pertains to have more errors or frauds in the *same* type of digital simulation or case and different dataset sizes.

The contribution of this paper is threefold:

1. To do an exploration of applying unsupervised algorithms in a financial auditing context. In order to do so strong domain knowledge is required of financial auditing, as well as the understanding of unsupervised algorithms.
2. To provide additional benchmarking of the performance the different types of algorithms different datasets across multiple evaluation metrics. The emerging literature on anomaly detection, combined with new types of algorithms, will give additional insight in how new types of algorithms perform.
3. To formulate research directions which have relevance for the data science community as well as the auditing community. The set of problems encountered for further implementation within the audit profession is formulated.

1.2 Research Setup

This is a paper which is an exploration of multiple unsupervised algorithms within a financial auditing context. The consequence of this explorative nature is that no central hypothesis is formulated. The steps within this research are the following:

1. Understanding the different characteristics of the Financial Auditing Problem.

2. Reframing the auditing problem within a data scientific approach.

3. Literature review to identify different types of relevant algorithms.

4. Feature Construction from the data using Financial Auditing knowledge.

5. Evaluate the performance of different algorithms using multiple evaluation metrics.

6. Discussion of findings for further suggestions.

The focus of this paper is to do a strong exploration of the usage of algorithms within an auditing context. In order to perform research a special audit simulation is used. This audit simulation is provided by Nyenrode Business University and simulates a real life client who trades in laptops on a large scale. Multiple frauds and errors are inserted into the dataset. It is possible to change certain aspects of this audit case and thus insert more errors and more frauds. Two main settings will be explored using this simulation as input data for the algorithms. One dataset where there are relatively lower number of errors in the auditing simulation and one dataset where there are around three times as much as the lower one. In addition a smaller dataset is also explored. All the settings and datasets are realistic representations of real situations which might occur. The performance of the algorithms will be explored for all settings.

1.3 Structure Paper

The structure of this paper is set out as follows: In section 2 a description is given of the auditor process and where the data scientific approach is relevant. It explains the context and the problem to be solved from a data scientific point of view. Section 3 is dedicated to a literature study and finding relevant algorithms. Section 4 describes the (context of) the datasets, evaluation measures and feature construction. Section 5 contains the reporting of the experiments and discussing the results. Lastly section 6 contains the conclusions and the summary and some discussion for the relevance to the auditing profession. In addition a discussion is added for further suggested work.

2 Context Auditor and Problem Statement

Part of this paper is to explore how an effective usage of unsupervised algorithms might be used. This section is used to explain the professional context of the financial auditor. Figure 8 in the Appendices shows the context of the regular audit process on a higher abstraction level. On the right of Figure 8 contains the new procedures which are executed within this paper.

2.1 Regular Audit Process

The (yearly) financial statements of a company show the financial consequences which the activities of that year have had. The financial statements are an important document, showing for instance the profits of that particular year, for stakeholders such as stockholders and banks. These stakeholders rely on the reliability of the figures presented in the financial statements. For example, it may not become the case that the profits shown in the financial statements are 40 million euro's, when in reality the profits of the company are 30 million euro's. The task of the auditor is to audit the bookkeeping of a company to make sure that all the figures in the financial statements are an accurate description of the financial reality. In order to provide context to a highly regularized profession as the auditor Figure 8 in the appendix shows the "Regular Audit Process" on a higher abstraction level. Each step in the process is described here below.

The steps refer to Figure 8 in the Appendix

1. Step 1. The auditor performs a qualitative analysis for each balance sheet item and profit-and-loss item. Using the business algorithm, type of company and context the greatness of the risk for errors or frauds is determined for each type of data stream within the company.
2. Step 2. In order to perform the task the auditor obtains the bookkeeping which contains multiple datasets which are related to each other. A basic bookkeeping contains a data structure for recordings sales, costs, stock and bank transactions. These transactions are almost always fully stored digitally. All these together make up the bookkeeping which underlies the financial statements.
3. Step 3.The auditor uses sampling methods to randomly sample transactions from the bookkeeping in Step 2, from the datastream area's which are considered a possible risk for misstatements as specified in Step 1 (within most procedures of the auditor). Possible datastreams are for example sales, personnel costs, stock transactions. These transactions are then, within the sample, ordered and made ready for evaluation. Usually tools such as Excel or other general data softwares are used.
4. Step 4. Evaluation of every transaction using invoices, bank statements, orders, receipts etc. to determine if frauds or errors have occurred. If the sample contains too many errors or frauds then it is decided to repeat step 3 and obtain more samples to evaluate more transactions.
5. Step 5. A conclusion is formulated for the subsection of the data for the financial statements. Are there real material misstatements? Then a negative conclusion is formulated on the financial statements. If no (large) misstatements due to fraud or error are present, then a positive formulation is used.

2.2 Data Driven Auditing

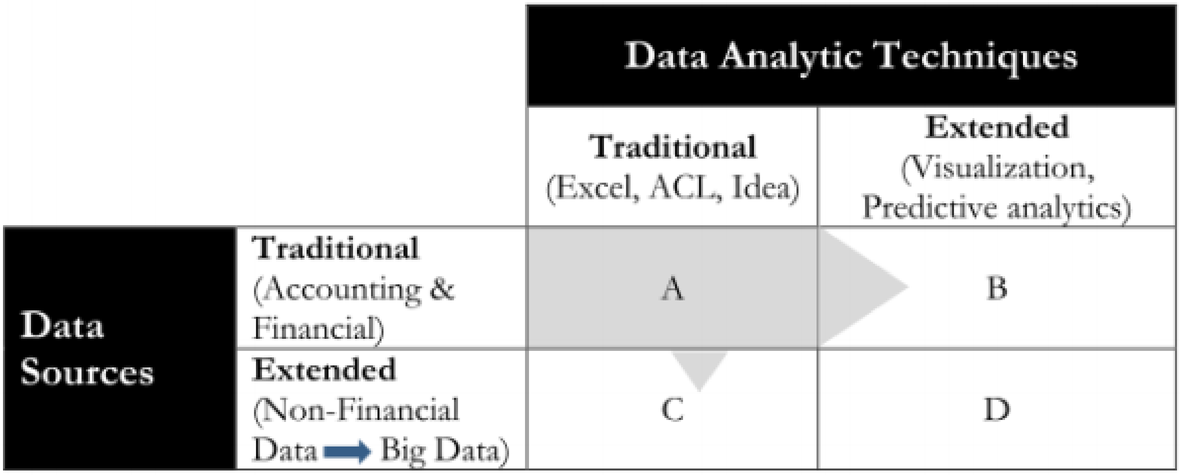
This section explains in a more abstract fashion which new steps are generated for increasing the chances of obtaining an effective procedure detecting errors and frauds. The right side of Figure 2.1 in the appendices is explained.
The steps refer to Figure 2.1.

1. Step 1. The auditor performs a qualitative analysis for each balance sheet item and profit-and-loss item. The auditor determines which types of datastreams are viable for using unsupervised algorithms, and which are not. All assertions and risks are determined for those datastreams.
2. Step 2. In order to perform the task the auditor obtains the bookkeeping which contains multiple datasets which are related to each other. The auditor identifies the risks relevant for the current datastream using domain knowledge. These risks require that certain features should be created (the combination between multiple general ledger accounts). A general and very simple example might be

to combine the existing feature ”invoice date” with ”shipping date” to ensure that no shipping date is *after* the invoice date (which in a financial auditing context would lead to incorrect recognition of financial item statements). The features should be (1) related to a specific risk, (2) using a qualitative judgement giving enough discriminative strength to be used within unsupervised algorithms. See section 4.2 for more information.

3. Step 3. Select unsupervised algorithms which are suitable for the auditing task and are suitable to the types of features constructed. Then use the entire dataset to train and test the unsupervised algorithm and thus detect frauds or errors for each transaction in the dataset. Multiple parameter settings for the same algorithm might be applied. See Section 3 for information on the types of algorithms used.
4. Step 4. The anomalies found by the algorithm are evaluated manually (if they truly are errors). No resampling is performed since the entire dataset is already included in Step 3.
5. Step 5. A conclusion is formulated for the subsection of the data for the financial statements. Are there real material misstatements? Then a negative conclusion is formulated on the financial statements. If no (large) misstatements due to fraud or error are present, then a positive formulation is used.

Figure 1 from Brandas et al. [5] shows the different type of data sources which can be used and what kind of algorithmic methods are used. The proposed shift and explored methods from section 2.1 and section 2.2 are a movement from, in Figure 2, A to B. This paper will focus on using traditional data sources of financial auditing from the auditing environment. So the traditional data sources like cost transactions and sales transactions are used for an unsupervised learning context.



		Data Analytic Techniques	
		Traditional (Excel, ACL, Idea)	**Extended** (Visualization, Predictive analytics)
Data Sources	**Traditional** (Accounting & Financial)	A	B
	Extended (Non-Financial Data ➡ Big Data)	C	D

Figure 1: Table Audit Approach Frameworks

2.3 Global Comparison Between Audit Approaches

The previous two sections, section 2.1 and section 2.2, provide a general overview of the regular auditing approach and data driven auditing approach. Table 1 contains the different characteristics which are relevant for Financial Auditing Purposes. A short description is given for each of the audit approaches explained in section 2.1 and in section 2.2. A small explanation for each of the left column of Table 1 is given here.

1. Qualification of work refers to the way the audit approach is executed. From a regular auditing perspective a lot of the work is done manually where the Data Driven Auditing is done with algorithms.
2. Data Selection refers to in which way the data are selected. The Regular Auditing Approach uses sampling methods to sample small sets of an entire datastream and to check for anomalies. Whereas Data Driven Auditing uses the entire dataset to generate anomalies.
3. Type of Judgement refers to in what manner the anomalies are determined. With the regular auditing process this is done subjectively, going through each transaction manually and subjectively decide if a

transaction is fraudulous or an error. For Data Driven Auditing this process is done objectively by the unsupervised algorithm.

4. Height of Auditing Effort refers to the amount of resources which are necessary to finalize an audit. For Regular Auditing this requires more individuals to perform the work, whereas for Data Driven Auditing this requires a relatively smaller amount (if compared to the amount of data which is verified and looked at).

5. Formatting of data refers to in what manner the data should be structured to adequately perform the audit process. The Regular Auditing Process is able to take on multiple formats, whereas the Data Driven Auditing Approach requires the data are formatted in a precise way.

6. Speed of Auditing refers to the time needed to complete the audit of a transaction stream.

Table 1 gives an overview of the comparison between the two audit approaches. Note the final results of this paper will indicate the general performance (for instance accuracy and recall) of the algorithms and is therefore not included as one of the characteristics compared in this table.

Overview: Global Comparison Audit Approaches

Important Aspects of Auditing	Regular Auditing (See section 2.1)	Data Driven Auditing (See section 2.2)
1. Qualification of Work	Largely Manual	Largely Automatic
2. Data Selection	Sampling of Data	Full Dataset
3. Type of Judgement	(More) Subjective	(More) Objective
4. Height of Auditing Effort	Relatively Higher	Relatively Lower
5. Formatting of data	Can take multiple forms	Enforces correct formatting (for Algorithm input)
6. Speed of Auditing	Slow	Fast

Table 1: Global Comparison of the Regular Audit Approach and the Data Driven Audit Approach

3 Theoretical Related Work

This section is to provide an overview of the related theoretical work on data science and auditing. Section 3.1 discusses the existing literature on data scientific techniques in a financial auditing context and what current progress is made within science and practice. Relevant theoretical considerations, in relation to the financial auditing environment, are discussed in the following sections. In section 3.2 the considerations for unsupervised and supervised learning are discussed. In section 3.3 the types of anomalies relevant to the audit problem are discussed. Then in section 3.4 the algorithms are selected which meet the criteria in the previous sections and the application to the dataset.

3.1 Existing Literature on Financial Auditing and Data Science

As mentioned in the introduction of this paper, this paper is a new exploration of using unsupervised algorithms within a financial auditing context. Some relevant research have been done on the subject of financial accounting. Such as [4], which demonstrates that a lot of work in the financial domain is performed. Credit card frauds and money laundering type of contexts has been researched. This research, as shown in table 6 of [4], uses supervised models. No unsupervised models are used within a financial auditing context.

More recently some limited amount of research is performed on unsupervised context. More basic clustering approaches, such as K-Means, are explored in [16] and [5]. The methods are promising and signal a first movement into using unsupervised learning. The methods are however quite rudimentary. The relatively small amount of research in (financial) auditing has been noted by Geppa et al. stating that the *" the use of big data techniques in auditing, and (...) the practice is not as widespread as it is in other related fields.* [1]. It is observed that many scholars have noted the lack of the usage of big data in auditing [1]. Reasons for this are also introduced in the introduction of this paper.

3.2 Unsupervised or Supervised Learning

The nature of the financial auditing context is such that the auditor never knows beforehand what types of errors or frauds might be present in the data. This paper assumes the auditor has only a single year to audit, and uses the approach in section 2.2 for the entire approach. The consequence is no supervised or semi-supervised methods are possible (see also section 6.3 for further suggestions on future work) since there are no pre-labeled target variables present. Therefore only unsupervised algorithms are used.

3.3 Type of Anomalies

There are multiple types of anomalies: point anomalies, contextual anomalies and collective anomalies [3]. An explanation of all types of anomalies in relation to financial auditing is listed below.

1. Point anomalies. Point anomalies are single instances in the data which are far off from the normal data and therefore irregular. These type of anomalies might be present within a financial context. Wrongly recorded invoices or irregular orders might easily be captured being a single irregularity.

2. Contextual Anomalies. Instances which appear as anomalies when made conditional on other features of the data. Some of the frauds or errors might be contextual. A common example within a financial auditing context is that a company has a seasonal pattern, a simple company like an ice cream company might suffice. Many products for this type of company are sold in summer (between May and September). A high sales amount in for instance winter time, such as December, might be considered anomalous. From a financial auditing point of view, and the content of the audit simulation (see also section 4.1), these type of anomalies are not considered.

3. Collective anomalies are anomalies which for a single instance are not directly anomalies (such as point anomalies or contextual ones) but taken together form a different group of instances. These type of errors or frauds are also quite common in the financial auditing context. Common error cases are employees who fraudulously extract multiple products (which have *per* product a small impact, but in

total a higher one) or an administrative system used for bookkeeping which doesn't work properly and therefore larger portions of invoices which are given wrong dates could be such an error. These dates are important within a financial auditing context.

In summary: in the exploration of unsupervised algorithms, the algorithms must be able to both detect point anomalies and collective anomalies since these types of anomalies are most common within a financial auditing context.

3.4 Algorithms Selected and Formalized Descriptions

This section shortly describes the selected algorithms used for exploration. Most algorithms explored are neural network algorithms. The general strength of neural network algorithms are explained in section 3.4.1. Then section 3.4.2 explores different types of Autoencoders used in this paper. In addition in section Restricted Boltzmann Machines are formalized in section 3.4.3. Lastly the Isolation Forest is described in section 3.4.4.

3.4.1 Neural Network Architecture Algorithms

Unsupervised neural network algorithms are effective algorithms for the financial auditing environment. The financial auditing has no ground truth and it uses non-sequential data, which have a regular point anomalies but also collective anomalies. neural network algorithms are effective tools for this setting, since unsupervised neural network algorithms learn inherent characteristics of the data. The assumptions the neural network algorithms make for anomaly detection are: (1) the regular sections in the dataset or compressed variables are different from the anomaly, (2) the normal data has a (much) larger presence than the anomalies, (3) the neural network algorithms are able to produce an anomaly score which are the result of learning intrinsic properties of the dataset [14].

Assumptions (1) and (2) are both relevant within the financial auditing context. Frauds, besides general errors, are a special case since these are transactions which are meant to be "normal". However with enough domain knowledge it is possible to detect also frauds.

3.4.2 Autoencoders

Simple Vanilla Autoencoder
A Simple Vanilla Autoencoder is a special type of (unsupervised) neural network algorithm where the input of the network is the same as the output. The goal of the Autoencoder is to encode the input features through a (usually) smaller amount of hidden nodes. From these hidden nodes a decoding process occurs to the output layer. The basic idea is that the output layer tries to reconstruct the inputs from the compressed representation. The compressed representation is generally a non-linear representation of the input layer. In summary the Autoencoder tries to learn the identity function. See Figure 2 for a visual representation.

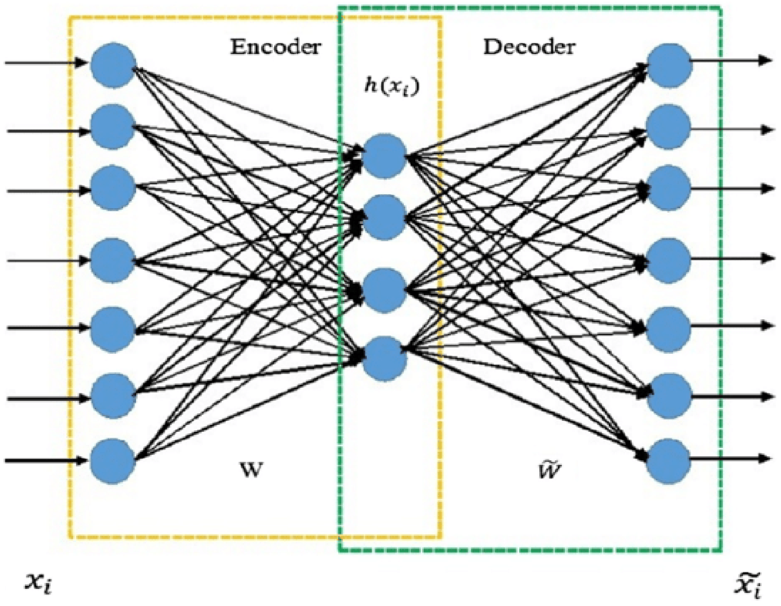


Figure 2: Simple Autoencoder

As displayed in Figure 2 the left side represents the input layer or the features which the dataset contains. The input vector is described by x_i and contains n features (n corresponds with the count of input nodes on the left side of Figure 2). This input is put through a set of weights (described by W in Figure 2 called the Encoder (or E) and results in a compressed representation described by the hidden nodes $h(x_i)$. The set of nodes described by $h(x_i)$ in Figure 2 is the latent representation in a lower dimension than the input nodes. From latent representation the decoder (or D) decodes the hidden layer to the output layer. As noted the input layer has the same number of n nodes as the output layer. The output layer on the right side of Figure 2 is described by $\tilde{x}_i$ and contains n nodes. $\tilde{x}_i$ is a reconstruction of x_i. $\tilde{x}_i$ can be described by $\tilde{x}_i = D(E(x_i))$.

The function which is optimized is given in Equation 1. The optimization function demonstrates the Autoencoder tries to learn the input data with the least amount of distortion.

$$\text{Reconstruction Error} = (\sum \tilde{x}_i - \sum x_i)^2 \tag{1}$$

The Reconstruction Error in Equation 1 is a way to measure the so called reconstruction error. The general idea is that the Autoencoder learns the deeper structure in the dataset from the latent features and then is able to use this to detect the anomalies. The compressed representation describes the normal transactions within the dataset well and will be less prone to learning the anomalies. Transactions which are fraudulent transactions would have a significant larger reconstruction error for that particular set of transactions, this is caused due to the fact that anomalies do not contribute to the latent hidden layer as much. The anomaly transaction(s) will have a higher reconstruction error which is the basis for the anomaly detection.

Exploration Advantage for Financial Auditing
The Vanilla Autoencoder is a basic structure from which other variations of Autoencoders are built (see also Contractive Autoencoder and Variational Autoencoder in this paper). The combination of the basic Autoencoder structure combined with the different variations might provide insight into the workings of these specific types of neural networks in a financial auditing context.

Contractive Autoencoder
Multiple Regularized Autoencoders, which add some form of regularization on top of the Contractive Autoencoder (CAE). The central characteristic of the CAE is that the learned representation is less sensitive to minor variations of the training records. The main difference with the Vanilla Autoencoder is that the CAE has an added penalty term which make sure there is a higher robustness to the training of the CAE. The CAE uses the Frobenius norm of the Jacobian matrix of the mapping of the encoding as a way for regulation of the sensitivity [15].

$$L = ||\tilde{X} - X||_2^2 + \lambda||J_h(X)||_F^2 \tag{2}$$

$$||J_h(X)||_F^2 = \sum_{ij}(\frac{\partial h_j(X)}{\partial X_i^2}) \tag{3}$$

The loss function in the CAE is similar to the Vanilla Autoencoder but a penalty term is added as shown in Equation 1. The penalty term is the Frobenius Norm of the Jacobian Matrix (as shown in Equation 3). The Jacobian Matrix contains the first-order partial derivatives for the hidden layer (h) with j nodes and is computed with respect to the input layer nodes (X) with i nodes of the CAE. The Frobenius norm of the Jacobian Matrix is calculated by summing up all the elements in the Jacobian Matrix. Where the rows of the Jacobian Matrix are i and represented by the input nodes of CAE and the columns of the Jacobian Matrix are j.

The consequence of this penalty term is that there appears a trade-off between the reconstruction error and the Frobenius norm of the Jacobian Matrix. More simply stated, it means that the CAE (compared to a Vanilla Autoencoder), will learn the larger variations the dataset contains and are less prone to changes to smaller variations. The Equation 3 is added to the loss function in Equation 1.

Exploration Advantage for Financial Auditing
Because frauds and errors in Financial auditing can be of different qualitative kinds, and the amount of the frauds and errors can differ the CAE is an algorithm which can provide robust results to differing variations in the data across multiple features.

Variational Autoencoder

The Variational Autoencoder is a generative algorithm. Variational Autoencoder (VAE) is described by Kingma et al. [12]. Just like the Vanilla Autoencoder it has an encoder, decoder and a loss function. The input of the VAE is compressed into a (hidden) latent representation (noted by z) which reduces the dimensionality of the input layer. A main difference with the Vanilla Autoencoder is that the VAE has a lower dimensional space which is of a stochastic nature. The encoder computes a set of parameters of $q_\theta(z|x)$ where x is the input of the VAE and z the latent representation. This represents a Gaussian probability density. The z is a stochastic variable. The VAE therefore converts the input to a continuous distribution with a mean and a standard deviation. The decoder of the VAE converts the latent representation to an output. The decoder is described by $p_\phi(x|z)$. Where ϕ represents the weights of the decoder from the latent representation z to the output layer. Figure 3 shows the general structure of a Variational Autoencoder as described in this paragraph.

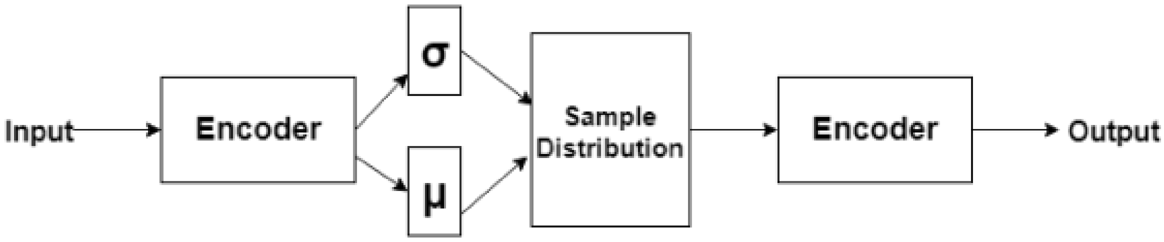


Figure 3: Variational Autoencoder

The loss function of this model is shown in Equation 4. It has two components: a negative log likelihood of the reconstruction error using the aforementioned decoder and encoder and also a Kullback-Leibler Divergence term. The loss function of a VAE does not contain general representation which are relevant for all the contents of the dataset. The first term can be described as the true reconstruction error. The $\mathbb{E}$ is computed in relation to the distribution of the encoder over the input. The first term basically states that if the decoder isn't able to create a good parametrization of the distribution of the input data then a larger loss will occur. Since there are no general representations for all datapoints the loss function is using single

data points l_i for the calculation of the total loss. The total loss is then computed by summing all N instances. The loss function for each input datapoint is then described by Equation 4. As mentioned above: θ and ϕ are respectively the weights and biases of the encoder and the decoder of the VAE.
The second part of Equation 4 is the Kullback-Leibler divergence between the two mappings of encoder and decoder. It gives the opportunity to measure how well the term q represents p. In a VAE, the term p is described with a normal distribution of the latent representation with a mean of 0 and a standard deviation of 1. This penalty term is necessary to make sure that the latent space z is kept a decent uniqueness. The smaller differences between common datapoints are then kept in a certain place within a Eucledian space of a Gaussian distribution.

$$l_i(\theta, \phi) = -\mathbb{E}_{z \sim q_\theta(z|x_i)}(\log p_\phi(x_i \mid z)) + \mathbb{KL}(q_\theta(z \mid x_i) \mid\mid p(z)) \tag{4}$$

An important aspect of a VAE is the usage of the reparametrization trick. Autoencoders use a backpropagation structure to optimize the identity function. However the latent representation z is a fixed representation of the distribution of q. Causing the derivative to be zero (but this derivative should of course not be zero). The reparametrization trick formulates the calculation of $z = \mu + \sigma \circledcirc \epsilon$ provides the opportunity to allocate the stochasticity of a Gaussian latent variable z to the ϵ. Then ϵ is calculated using a Gaussian normal distribution. Making it possible to calculate the derivatives of functions taking z as input. Figure 4 shows the difference, where the circled symbols represent the parameters which are stochastic, and the squared symbols which are deterministic. The deterministic quality allows backpropagation to function normally.

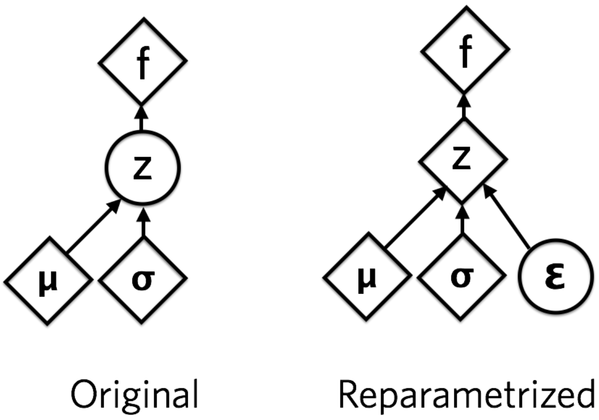


Figure 4: Reparametrization Trick for Variational Autoencoder

Exploration Advantage for Financial Auditing

The main advantage this type of algorithm has is that the output of the algorithm provides disentanglement. Disentanglement means that the factors try to differentiate themselves as strongly as possible from each other. [12] [10]. Being able to produce (in reduced dimensionality) the new variables ensure that one gets more independent latent variables. Each latent factor is then able to algorithm an independent underlying structure. It can be stated on a higher conceptual level that this is similar to the Varimax Rotation in Principled Component Analysis (which can be considered an Autoencoder with linear activation functions), where variables are loaded as much as possible on a single factor. Obtaining larger independence between the different factors [2].

Important consideration is that the dataset, see section 4.3, has mixed types of data. Categorical as well as continuous. Categorical data provide some complication in using variational autoencoders, since the autoencoder is trying to learn a distribution with a mean and a standard deviation for categorical data. In order to correct this a reparametrization is applied to categorical data called the Gumbal-Softmax transformation [11]. This results in data being more qualified to be effectively used by the VAE.

Transactions within a company have different groupings and underlying reasons. Each of the transaction streams have a different underlying structure (type of products, type of clients, discounts etc.). Multiple features can work together in a certain way, in light of the different type of transactions. It might provide a large advantage to learn independent latent features which are used for anomaly detection, taking into consideration these different independent distributions. The VAE provides this advantage over the normal Vanilla Autoencoder.

3.4.3 Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) [10] is a certain form of a (stochastic) neural network. The main architectural difference with Autoencoders is that the RBM lacks an output layer, and only has an input layer (usually called the visible layer) and one hidden layer, see Figure 5.

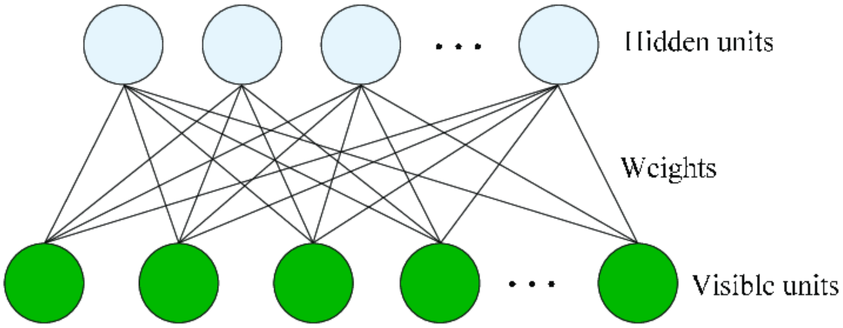


Figure 5: Restricted Boltzmann Machines

The RBM has weighted undirected edges between the different sets of nodes. RBM is a bipartite graph, so no edges exist between a single layer (a single layer is either the hidden nodes or the input nodes). The hidden nodes only takes on the binary positions of zero and one. The RBM uses uses an underlying joint probability of input and hidden nodes. The joint probability tries to match a strongly as possible the data such that the nodes match the input data. The joint probability is described by the following formula:

$$p(v) = \frac{1}{Z}\sum_h e^{-E(v,h)} \tag{5}$$

where Z is a partition function ensuring that $\sum p(v) = 1$,

$$Z = \sum_{v,h} e^{-E(v,h)}, \tag{6}$$

And E is the energy function,

$$E(v,h) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} b_j hj - \sum_{i,j} w_{ij} v_i h_j, \tag{7}$$

The equation in 8 is a probability distribution and can be called a Boltzmann distribution. When optimizing the RBM the parameters are found by maximizing the product of probabilities using the input data of a training set V .

$$\arg\max_{a_i,b_i,w_{ij}} \prod_V p(v)\ (8)$$

Optimization is usually performed using the Contrastive Divergence algorithm. When the algorithm converges the output of the RBM algorithm is a Boltzmann distribution calculated using the features as input. This probability distribution can then be ordered from low to high on a record level. Transposing this gives a value for outlier detection.

Advantage for Exploration for the Financial Auditing Context
RBM's have a similar architecture to Autoencoders, but have a different approach. RBM's have a probabilistic approach and are energy based algorithms. It may be expected, due to the different architecture and the undirected graph structure that the RBM is able to learn more freely the patterns in the dataset. At least learning other types of structures.

3.4.4 Isolation Forests

Isolation forest is, in contrast to the other models in this paper, a non-neural network algorithm to identify anomalies. The Isolation Forest has a similar structure as the random forest algorithm [13]. Whereas the general structure is to combine several decision trees and uses an averaging mechanism to decide on the outliers in the dataset. Instead of trying to learn the general structure of the dataset, Isolation forests however are focused on detecting anomalies. The general idea of Isolation Forests is shown in Figure 6. Anomalies have a different structure, or different values for each of the features in the dataset. As Figure 6 shows, the anomalies will be on a less deeper level when calculated from the root compared to very common records. Many trees are constructed with a random amount of features, and every transaction is averaged on which level the transaction is a leaf in the tree. The anomalies will most likely be more often at a more shallow level (so in Figure 6 the red leaves in the tree) than the more common ones.

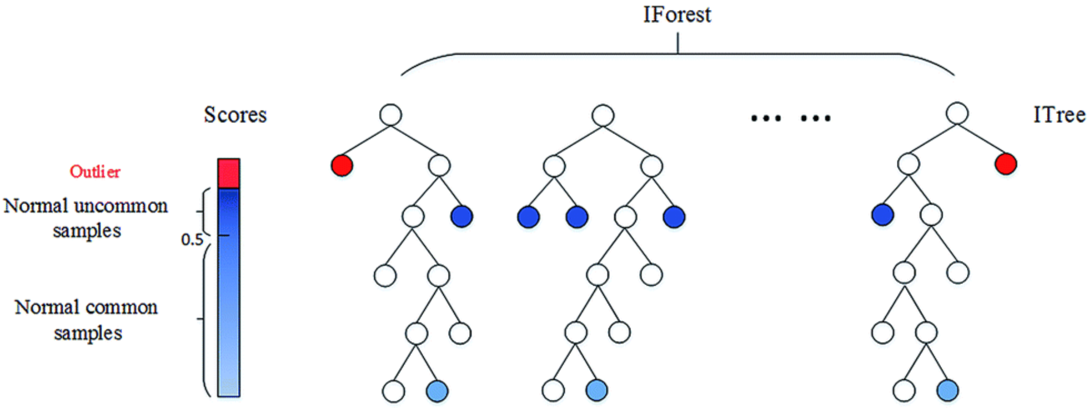


Figure 6: Isolation Forest

Advantage for Exploration for the Financial Auditing Context
Isolation Forest is a different type of architecture than the neural network algorithms formalized in section 3.4.1. The architecture of Isolation Forest might be stronger for detecting point anomalies since the uniqueness of these type of anomalies might be better captured by the leaves of the trees in the ensembling.

3.4.5 Ensembling Algorithms

Ensembling is a way to combine the results of trained algorithms to obtain more effective results [6]. Ensembling has taken more interest in the context of unsupervised learning [17]. Ensembling combines the results of single algorithms in multiple ways. Most common ways are ranking methods, stacking and boosting [17].

Ensembling is a cheap method in which one can combine the outlier detection strength of multiple methods. The different algorithms described in section 3.4 have different characteristics and as a consequence other types of anomalies can be detected. The method used for ensembling in this paper is the "rank-aggregation" method which is a common and effective method [17]. This method normalizes the outlier scores for each algorithm, ranks every record for every independent model. This ranking is then averaged for each transaction across all used algorithms. This new average is the new ensembled anomaly score.

4 Research Approach

This section is dedicated to the research approach. Section 4.1 provides some background to the financial auditing case which is used to run the algorithms on. Section 4.3 gives a description of the datasets used. Section 4.2 provides some guidance on the general framework used to construct features for fraud and error detection. Section 4.6 describes the programming library's which are used to perform the experimentation. Finally Section 4.7 discusses the evaluation metrics used.

4.1 The Audit Client: Digital Simulation

The experiments for the algorithms are performed within an audit simulator. The Audit Simulator is used in many accounting eduactational contexts, mainly Nyenrode Business University. The general website of the simulator is: https://www.auditgaming.nl/. As mentioned, the data are owned by Nyenrode Business University.

The company which is used from the audit simulator replicates a real life medium or large sized company. The company that is used in the Audit Simulator is a company called "Laptopworld.nl". The company sells laptops using an online platform (website) to individual customers and to businesses. All kinds of transactions such as cost of sales, sales, invoices, orders, bank transactions, client information etc. are included within. The audit simulator is accredited by the "Nederlandse Beroepsorganisatie Accountants", meaning it is used to train financial auditors. The simulation provides a unique opportunity to also include a ground-truth. Meaning it can be checked what the output of the algorithms is. So the output of the unsupervised algorithms can be verified (see also Section 4.3). It is possible to manipulate the data in such a way that more frauds and errors occur within the data. This makes additional experimentation possible.

Laptopworld's strategy is based upon buying large bulk purchases of laptops which gives large discounts. This gives the opportunity to sell laptops for a low price. The company has a CEO which is also 100% shareholder in the company. The company has a vision for the coming years that growth (in revenues) is highly important. The sales totally used for the "larger dataset" contains about 380.000 records and 300 million euro's in revenue and the "smaller dataset" have 40.000 records and 40 million euro's in revenue.

4.2 Feature Construction

The audit simulator provides multiple options to extract structured datasets from the simulator. The digital simulator has a large variety of different aspects such as salary of employees, cost of sales, other costs and other special transactions. In the context of this paper it is decided to focus on the sales of the company. The two reasons for this choice are: (1) sales is one of the richest contexts to have fraud risks from a financial auditing point of view, (2) sales have larger volumes which provide the opportunity to experiment with the selected algorithms. The amount of records can be changed manually. Two different sizes of datasets are used within the context of this paper, see section 4.3.

In order to obtain relevant features to be used in algorithms a short description is given of the contents of the digital simulator for the sales section. The sales process of the digital simulator has 15 different datasheets (such as: a price sheet with 8 features which holds information about the products an prices given, or an invoice datasheet with 19 features about the invoices sent to customers). The 15 datasheets contain in total 181 unique features. Not all features are directly relevant for detecting frauds or errors and need to be processed to be effectively used for the chosen algorithms in section 3.4.

The author of this paper is an experienced auditor. In order to obtain relevant features, creating combinations between the existing features or removing irrelevant ones, expert knowledge is used. The first two steps as discussed in section 2.2 are performed to construct features which are indicative of frauds or errors. This is a relatively challenging task to perform. It is to find a cross link between auditing methodology and data science. Note that the author of this paper was unaware of the ground truth of the true frauds and errors, but only at end of this research this was made available. This is mainly for the exploration purposes of this paper, in order to discover what process one has to go through if a financial auditor does not have the ground truth and tries to go through the process described in section 2.2.

Due to the confidentiality of the data not every variable is explained. To provide one example: a risk of fraud or error is the usage of discounts in sales. Discounts have internal guidance when they are given to clients. In many cases there is a fraud aspect in giving discounts which were not allowed or providing discounts which are not reported in the bookkeeping as such. The different categories of types of risk of frauds and errors are explained and they are listed below:

1. Completeness risks (amount of products). Not all the sales are recorded in the bookkeeping of the sales. A relevant feature which is created for this type of risk is that the amount of product that is ordered and invoiced are also shipped to the customer (meaning that all the products are accounted for in the sales). If some products are not invoiced but they are shipped this might cause a breach from a financial auditing point of view.
2. Correctness risks (on products). Incorrect products are recorded in the sales. Incorrect pricing is an example. A concrete feature which is used is that within the simulator there is a central price sheet which records the prices for every type of products sold. A breach from a financial auditing point of view might be that an invoice in sales holds the incorrect price which does not reconcile with the price sheet.
3. Tax risks. Risks related to the taxes of selling the products. There are different tax rates, sales taxes and the corporate taxes. The correct rates should be calculated and recorded. Some products have lower taxes than others. From an accounting point of view this might cause a risk.
4. Cutoff risks. The reporting of aspects related to sales are not reported in the correct period. The period which is under audit is the year 2018. Multiple features are created to check if transactions are recorded fairly and correctly in the year 2018.
5. Management Override risks. Risks related to the authority of high management or employees with large authorization rights.

A combination is found between the 180 features which are useful from a financial auditing point of view. The features are a combination of binary, categorical (more than two categories) and continuous variables. In total 20 features are used in the algorithms. From these 20 algorithms, 10 are real values, 3 are categorical (with more than 2 categories) and 7 are binary variables (which are features created to reconcile features as such. Few examples of the features are dates, prices, amount of products sold and the specific features to detect the risks mentioned earlier.

4.3 Datasets

All the algorithms are run on two 'larger' datasets which contain multiple sales transactions. The amount of records are 377,311 in the larger dataset. The number of errors and frauds in dataset 1 is small (called large dataset minor), where in dataset 2 (called large dataset major) this amount of frauds and errors is increased by threefold. Both datasets are still considered extremely imbalanced data [8]. However these types of distributions are quite common within a financial auditing context and describe a relative common (estimated) distribution for the current company audited (in the sales process).
In addition, the algorithms are also run on a smaller dataset. The reason is to explore the strength of the algorithms selected on smaller datasets. For instance neural network algorithms are known to use a large amount of data to work properly. Since companies with a revenue of 40 million are more common it is adds to the external validity of this paper to find the application to other datasize settings as well. All the datasets contain two distinct frauds and two distinct errors in each dataset (which are all divided amongst the total frauds/error records described below).

The used datasets are described as:

1. Small data set: 40,829 records of which 261 are fraud/error records (0,639%)
2. Big data set minor frauds and errors: 377,311 records of which 467 are fraud/error records (0,124%)
3. Big data set with more frauds: 377,311 records of which 1,662 are fraud/error records (0,440%)

The amount of fraudulous and error records shows that the datasets are highly unbalanced.

4.4 Optimization of Parameters Algorithms

4.4.1 Train, Test and Validation Methods

In order to obtain the reconstruction error the entire dataset is used for the determination of the reconstruction error. The task performed is a unsupervised learning task and the training and test data are not separated [9]. In addition no validation methods (such as Cross-Validation) are applied, due to the unsupervised learning task.

4.4.2 Threshold Reconstruction Error: Post-Processing

One of the more important parameters to validate is the threshold for the reconstruction error (for the Autoencoders). The Autoencoders use the reconstruction error to optimize for the replication strength of the algorithm. The higher the reconstruction error for *each* transaction the higher the probability this particular transaction is an anomaly. A threshold needs to be selected to determine the boundary for anomaly. Each transaction higher or lower than this boundary might be considered an anomaly. The main problem is there is an extreme imbalance in the anomalies and non-anomalies in the dataset. Determining the threshold is relatively challenging by a standard metric. No research has provided a robust heuristic to the knowledge of the author of this paper. Selecting the correct threshold is an important post-processing step to determine which transactions are considered anomalies.

A method to be used might be for instance using a standard percentile from the anomaly scores from the algorithms. A concrete example is to calculate the mean from the anomaly scores from an algorithm and consider everything above the three standard deviation from the mean as anomalous. This method is however often quite inaccurate and quite general.

The approach selected is to plot the reconstruction error for each transaction and see visually what an effective threshold might be. This method is not algorithmic, but provides enough basis for determining in a visual approach what the anomaly threshold should be. See figure 7 for an example on the subset of the data in determining a correct threshold. The red line shows what threshold is chosen for this specific dataset. This is done by using the plot, which is ordered by time stamps of the transactions in the dataset. Note the threshold is determined without a legenda fraud vs non fraud, meaning that all the data points are blue. This method gives a decent amount enough to determine what amount of anomalies to select (and the distinctness of the anomalies). This method has its limitation due to the subjective nature of selecting the threshold.

4.5 Distributions Fraud vs. non Fraud

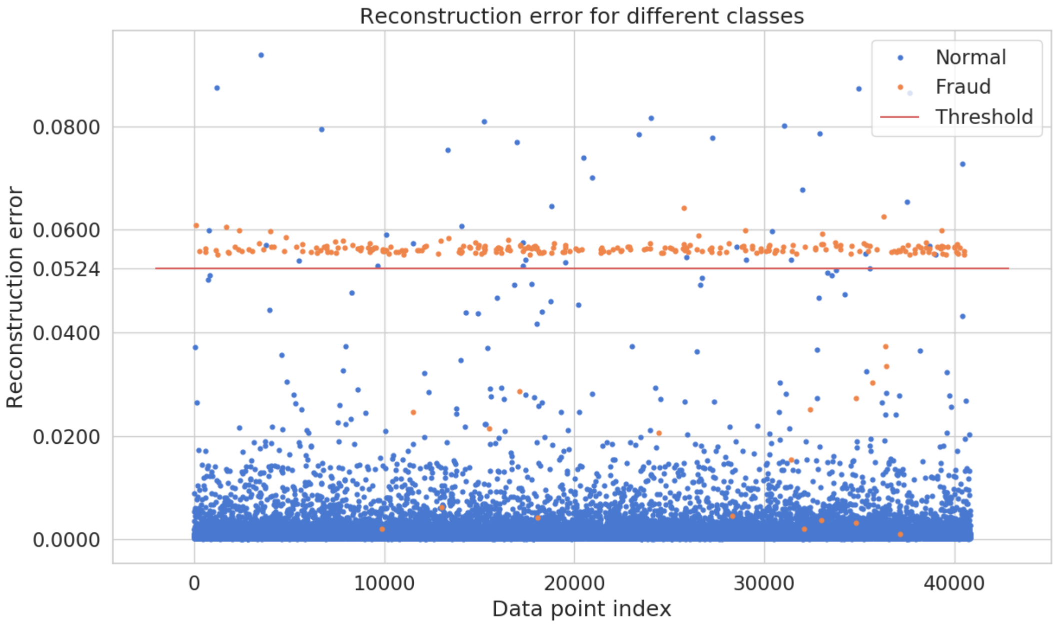
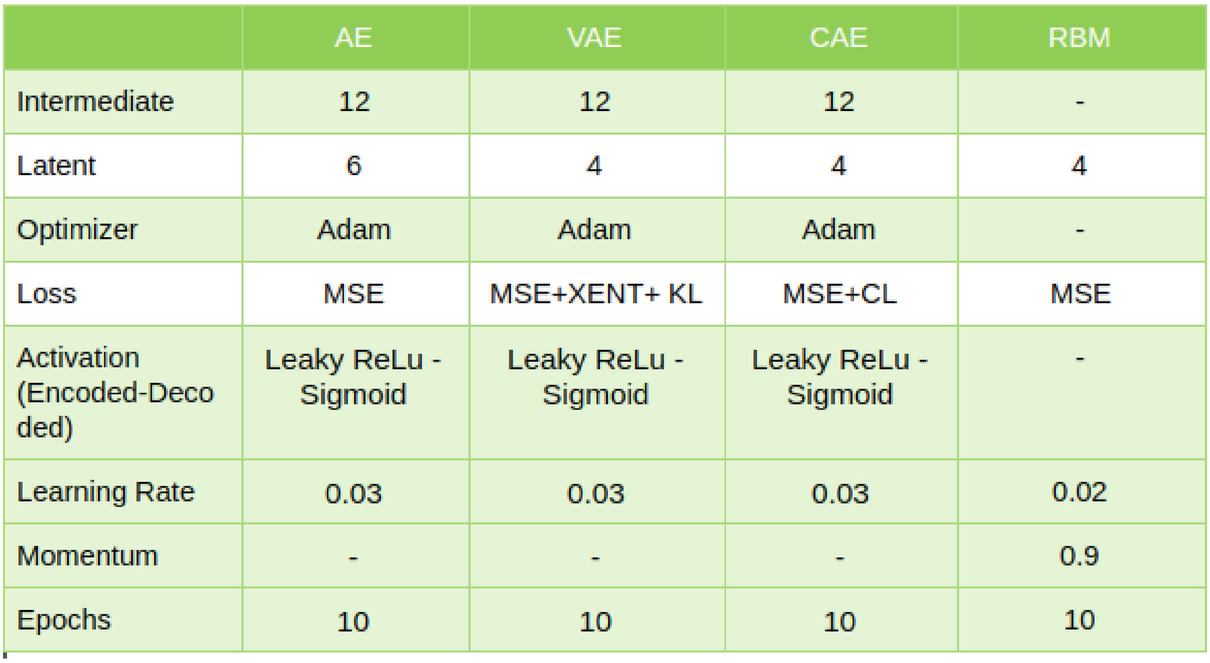


Figure 7: Visual Method of Selecting Anomalies

4.5.1 Parameters Unsupervised Algorithms

The parameters used within the unsupervised algorithms are not optimized in a strong fashion. The way the models are selected, at least the autoencoders, using a method of trial and error. It is demonstrated that using 1 hidden (intermediate) layer between the input layer (with 20 input nodes) and a latent layer (with 6 nodes) has the most robust and strong performance across all algorithms. Note that in a real life case the financial auditor would not be able to do this similar type of optimization described here since the real ground truth for frauds and errors is not present. Therefore relatively standard parameters and settings are used and are showed in table 2. The loss function, the activation function, the learning rates and epochs are all kept standard.



	AE	VAE	CAE	RBM
Intermediate	12	12	12	-
Latent	6	4	4	4
Optimizer	Adam	Adam	Adam	-
Loss	MSE	MSE+XENT+ KL	MSE+CL	MSE
Activation (Encoded-Decoded)	Leaky ReLu - Sigmoid	Leaky ReLu - Sigmoid	Leaky ReLu - Sigmoid	-
Learning Rate	0.03	0.03	0.03	0.02
Momentum	-	-	-	0.9
Epochs	10	10	10	10

Table 2: Parameter settings for algorithms

For the Autoencoders the Intermediate Layer is the layer between the Input layer and Code layer.

Isolation Forest: { 'max_samples' : (size of set)/10, 'contamination' : 0.01, 'n_estimators': 9, 'warm_start': True, 'bootstrap': True} The scikit-learn implementation for Isolation Forest has the default settings.

4.6 Programming Library

The main programming libraries to implement the algorithms are listed below. The implementation is done in Python 3.7. A short description is given for each library and the function is serves for the implementation.

4.6.1 Scikit-Learn

Scikit-Learn is a versatile library which includes multiple algorithms and error measurement capabilities. This library is constructed on library's as Numpy and can be used for regular data science learning datasets. Scikit-Learn is used for the implementation of Isolation Forest and multiple evaluation measures.

4.6.2 Pandas

Pandas is a library for data manipulation and analysis. The Pandas library provides fast and efficient ways to manage and explore data. It's main data objects are Series and DataFrames. Pandas is used mainly to manipulate the data sheets from the digital simulation environment into a final DataFrame with all (constructed) features for the algorithms.

4.6.3 TensorFlow

TensorFlow is a library which mainly supports the usage of computationally heavy algorithms. Tensorflow fundamental building block are the data-flow graphs to perform the computations of more complex neural network algorithms. The computations are performed in a combination of edges and nodes which the graph consist out of. Tensorflow is (partly) used to implement all the neural network algorithms as described in section 3.4.

4.6.4 Keras

From the Keras website: Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Also aspects of Keras are used for implementing the algorithms in section 3.4.

4.7 Evaluation Metrics

In order to evaluate the performance of the different algorithms multiple evaluation measures are used. A diverse number of evaluation measures are selected in order to capture multiple dimensions of performance measures. Here below the different evaluation measures are mentioned, the corresponding formula's (using the True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN)) and a short description of the evaluation measure.These measures are also captured in confusion matrices.

1. Accuracy. Formula: $\frac{TN+TP}{TP+FP+FN+TN}$. Accuracy is described as the percentage of the instances which received a good classification.
2. Precision. Formula: $\frac{TP}{TP+FP}$. A figure that describes the instances which are predicted to be frauds or errors, and truly are frauds or errors.
3. Recall. Formula: $\frac{TP}{TP+FN}$. This measure describes how effective an algorithm is able to identify frauds or errors.
4. Specificity. Formula: $\frac{TN}{TN+FP}$. Measures the effectiveness that an algorithm is able to identify regular instances.
5. False Positive Rate. Formula: $\frac{FP}{FP+TN}$. The amount of misclassified instances in the data.
6. ROC. Graph where True Positive Rate is plotted against the False Positive Rate.
7. F1-Measure. Formula: $\frac{2*(Precision*Recall)}{Precision+Recall}$. Is the weighted average of the precision and the recall.
8. Mathews Correlation Coefficient. Formula: $\frac{TP*TN-FP*FN}{[(TP+FP)*(FN+TN)*(FP+TN)*(TP+FN)]^(1/2)}$. Another balanced measure, such as the F1-Measure. Is noted to work well on strongly imbalanced datasets.

From a financial auditing point of view, recall is the most important measure. As described in section 2.1, the goal of the auditor is to make sure the final financial statements of the year are free of *material* errors or frauds. In addition precision is important from an efficiency perspective. The auditor cannot audit manually a disproportional amount of false positive instances extracted from the unsupervised algorithms. For the presentation of the results the simple Vanilla Autoencoder in section 3.4.2 is used as a baseline and the other algorithms are compared with it.

5 Results

This section concerns the reporting of the results. The reporting is split up in three subsections, corresponding to the different datasets in section 4.3. Section 5.1 reports on the results found with the smaller dataset (40.000 records). Section 5.2 reports on the larger dataset which contains less frauds (a larger imbalance) (377.000 records). Section 5.3 reports on a dataset with the same records as in section 5.2 (377.000 records), but where the imbalance of the classes is less prevalent.

5.1 Small Dataset

Relevant for the discussion and the presentation of the results is Table 3, the distribution plots of the error functions of each algorithm in the Appendix in section A.1 and the plots in the Appendix in section A.2 which show the chosen threshold for each of the algorithms as explained in section 4.4.2.

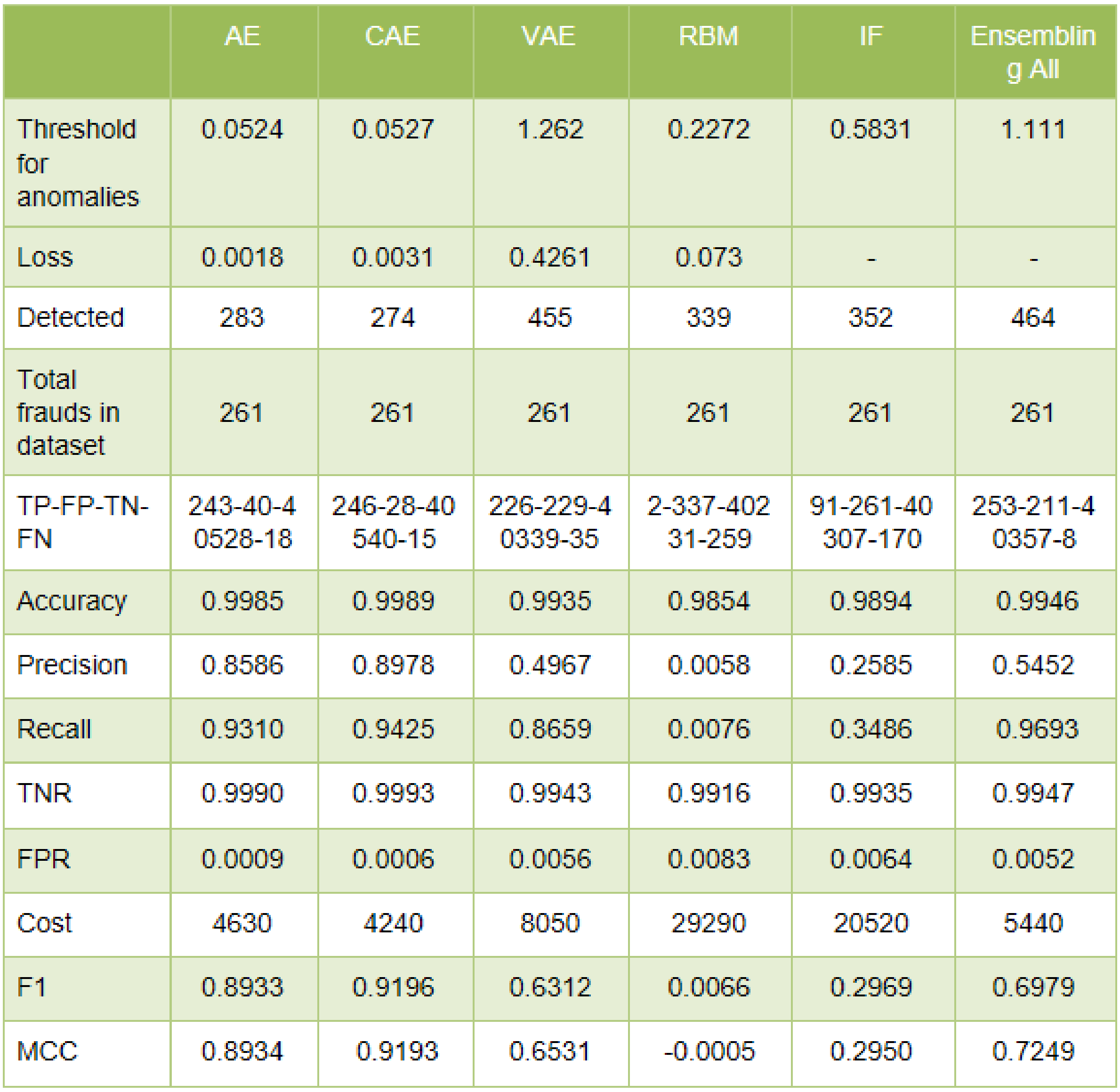
In general it can be stated that the Autoencoder algorithms outperform the RBM and the IF on all the important metrics as displayed in table 3. The F1 (0.00066 for RBM and 0.2969 for IF) and the MCC (-0.0005 for RBM and 0.2950 for IF) measures are of a weak performance for RBM and IF and can be considered not useful for financial auditing contexts. F1 and MCC are heavily influenced by a low precision and recall for RBM and IF algorithms. RBM has the worst performance. This is counter intuitive to the general performance of the model in previous papers (see 3.4.3). The hypothesis is that RBM's are not fully functional on using a diverse amount of data types. The features, as described in section 4.2, are mostly real valued features. These are the types of features which the RBM has more trouble working with. The absence of stronger discriminative power of the RBM algorithm is also made visible in Figure 19 of section A.2 where almost all true anomalies are assigned a lower error value compared to the other parts of the transactions.

For IF it might additionally be the case that the IF is able to not detect anomalies very strongly due to the fact there are four types of frauds/errors in the dataset. These are anomalies which have a recurring character throughout the time period of this dataset. If an anomaly can be discovered due to a "feature A", but it is a type of anomaly which occurs with some intervals, it means the anomaly can be seperated by another feature A and feature B. This means the probability these anomalies will have a deeper 'leaf' level in the IF algorithm is relatively high. Also the probability that normal transactions are split in a (set of) feature(s) in a certain way, making them anomalies by this algorithm is also higher from this point of view. This is visually supported by Figure 18 in section A.2, where fraudulous and normal transactions are at the top all scatter relatively evenly. It is visible the algorithm is able to assign a higher error to all the "orange" fraudulous dots (since none of these transactions are below the error rate of 0.450).

The Autoencoders perform relatively well. The VAE has a MCC of 0.65 and an F1 of 0.63. The results are of a decent value but lower than AE and CAE. Recall of the VAE is of a similar order of magnitude with the AE and CAE. The precision is of a lower quality. This might be due to the probabilistic nature of the algorithm and the problems that VAE might still have with using binary types of variables. The Gumbel-Softmax transformation [11] might assist in the conversion to a usable input for the VAE. But trying to obtain sampled Gaussian Distributions from binary features on which the Gumbel-Softmax [11] transformation has taken place might not fully solve all problems related to this.

The AE and the CAE both perform strongly on the dataset. AE has a F1 of 0.8933 and MCC of 0.8934 and CAE has a F1 of 0.9196 and MCC of 0.9193. Of the 261 frauds (or anomalies) which are in the dataset the algorithms had a True Positive count of 243 and 246 for AE and CAE respectively. This is a very strong performance for financial auditing contexts. The amount of work a financial auditor still needs to perform has reduced dramatically after using these algorithms. A financial auditor would in normal circumstances need to setup a large amount of manual procedures for detecting these types of anomalies. The algorithms give an almost full proof and more objective output on the entire dataset. Using a more qualitative judgement from the author, obtaining the same amount of assurance on the absence of anomalies or fraud would take four auditors and around ten manual procedures to obtain similar results. In auditing the types of anomalies are more closely.

Lastly the performance of the ensemble where every algorithm (AE, CAE, VAE, RBM, IF) is combined. It is clear the recall at 0.96 outperforms the recall of all other algorithms. The precision is of a worse quality at 0.54. Average Ranking Ensembling ensures a higher recall which can be useful within a financial auditing context, since the probability that Ensembling will have a more complete picture of what the anomalies truly are is useful within financial auditing contexts (since one wants to find all the possible anomalies affecting the financial statements. The precision however is not very strong, this is due to the fact that the precision of the VAE, RBM and IF are all lower than 0.5 this is a logical consequence.



	AE	CAE	VAE	RBM	IF	Ensembling All
Threshold for anomalies	0.0524	0.0527	1.262	0.2272	0.5831	1.111
Loss	0.0018	0.0031	0.4261	0.073	-	-
Detected	283	274	455	339	352	464
Total frauds in dataset	261	261	261	261	261	261
TP-FP-TN-FN	243-40-40528-18	246-28-40540-15	226-229-40339-35	2-337-40231-259	91-261-40307-170	253-211-40357-8
Accuracy	0.9985	0.9989	0.9935	0.9854	0.9894	0.9946
Precision	0.8586	0.8978	0.4967	0.0058	0.2585	0.5452
Recall	0.9310	0.9425	0.8659	0.0076	0.3486	0.9693
TNR	0.9990	0.9993	0.9943	0.9916	0.9935	0.9947
FPR	0.0009	0.0006	0.0056	0.0083	0.0064	0.0052
Cost	4630	4240	8050	29290	20520	5440
F1	0.8933	0.9196	0.6312	0.0066	0.2969	0.6979
MCC	0.8934	0.9193	0.6531	-0.0005	0.2950	0.7249

Legenda for Columns - AE: Vanilla Autoencoder, VAE: Variational Autoencoder, CAE: Contractive Autoencoder, RBM: Restricted Boltzmann Machine, IF: Isolation Forest. Legenda for rows - Threshold: The visual determination (see section 4.4.2 for anomalies, Loss: the loss obtained by the loss function of the algorithm (see section 4.5.1), Detected: The cases identified by the algorithm as anomalies, Total Frauds in Dataset: is the amount of real frauds/anomalies in the dataset. TP-FP-TN-FN: is the True Positive, False Positive, True Negative, False Negative count of the transactions. The performance measurements Accuracy, Precision, Recall, True Negative Rate, False Positive Rate, Cost, F1 and MCC are described in section 4.7 ”Evaluation Measures”. .

Table 3: Metrics results for small dataset

5.2 Big Dataset Less Frauds and Errors

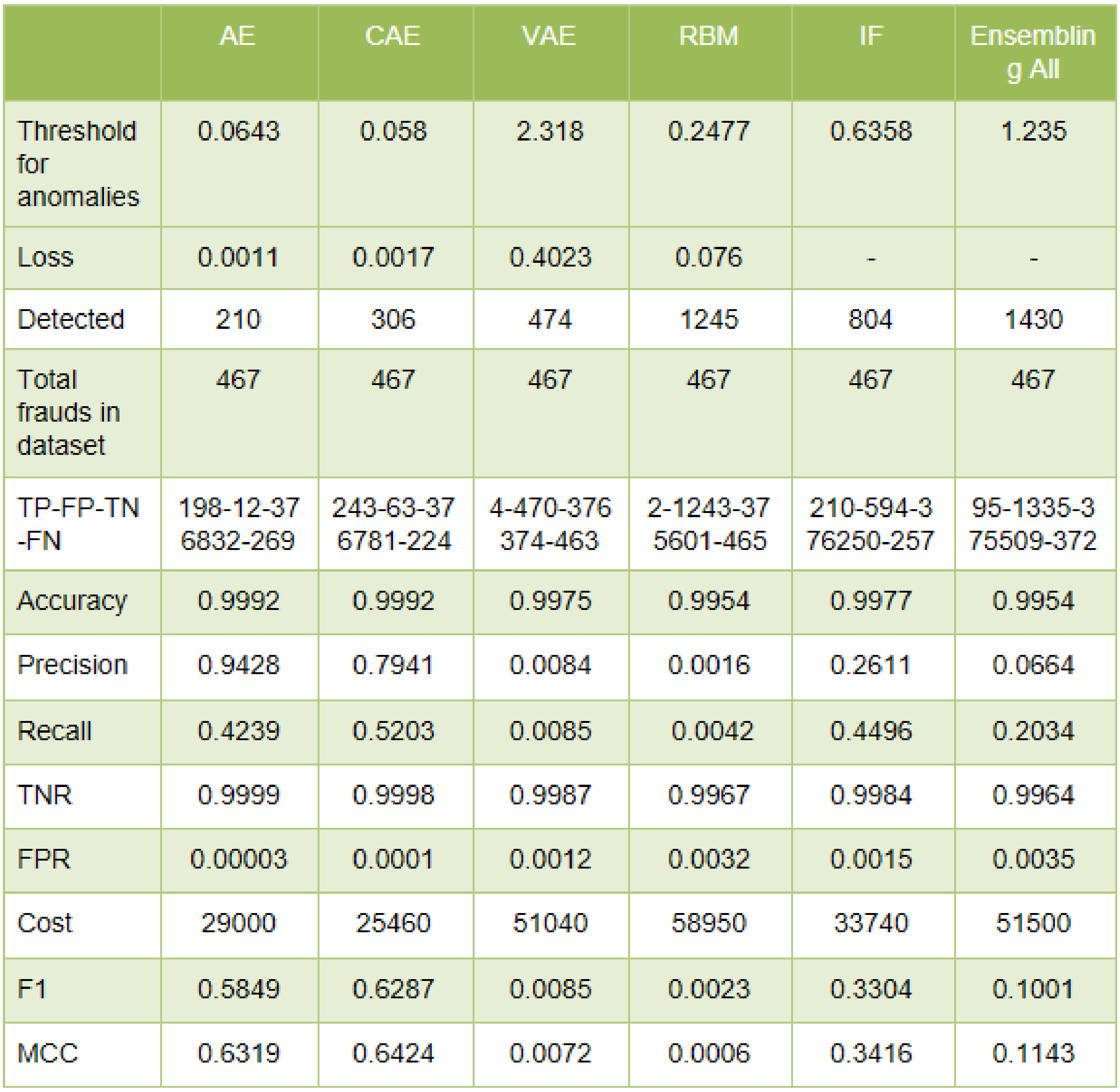
Relevant for the discussion and the presentation of the results is Table 4, the distribution plots of the error functions of each algorithm in the Appendix in section B.1 and the plots in the Appendix in section B.2 which show the chosen threshold for each of the algorithms as explained in section 4.4.2. Note that the same *type* of frauds and errors are present in this larger dataset compared to the smaller dataset in section 5.1, but the frequency of the anomalies has increased in the absolute sense.

When compared with the results for the smaller dataset in section 5.1 the overall performance of all algorithms have decreased. However the Precision metric has stayed relatively the same for the AE and the CAE. It can be noticed in Table 4 that RBM still has a low performance as already described with the smaller dataset in section 5.1 with a F1 of 0.0023 and MCC of 0.0006. The ensemble model is also heavily influenced by the low performance of RBM and VAE.

The VAE algorithm also has drastically reduced its performance compared to the smaller dataset, with a F1 measure of 0.0085 (0.6312 for the smaller dataset), MCC of 0.0072 (0.6531 for the smaller dataset). When looking at the reconstruction error plot in section B.2 Figure 28 there are multiple "layers" of errors present in the Varational Autoencoder. This might be due to the already discussed binary variables in the dataset, the VAE most likely has a harder time differentiating between transaction using binary variables. A larger number of the frauds and errors in this dataset can be detected using binary variables (in combination with real valued features as well).

Similar results are found in Table 4 compared to the smaller dataset is the IF algorithm which has a F1 measure of 0.3304 (for the smaller dataset 0.2969) and MCC of 0.3416 (for the smaller dataset 0.2950). The performance of the IF therefore has stayed relatively the same. It confers with the capacity for IF to be able to shift and adapt to new distributions [13].

The algorithms AE and CAE have a reduced recall, but have a good retained precision compared to the small dataset. It can be seen in the Appendix in section B.2 Figure 27 and Figure 29, the discrimination for respectively AE and CAE are quite strong and even have some uniformity (as in the smaller dataset) for the anomalies detected (with a few datapoints which have a larger error). The uniformity of the results may be described by the fact that many transactions might be considered an anomaly by the usage of binary features. In addition the dataset contains certain types of fraud, such as a "caroussel" fraud, which occurs monthly at a regular basis. This combined brings the conclusion that the results are plausible and also useful for financial auditing contexts.



	AE	CAE	VAE	RBM	IF	Ensembling All
Threshold for anomalies	0.0643	0.058	2.318	0.2477	0.6358	1.235
Loss	0.0011	0.0017	0.4023	0.076	-	-
Detected	210	306	474	1245	804	1430
Total frauds in dataset	467	467	467	467	467	467
TP-FP-TN-FN	198-12-376832-269	243-63-376781-224	4-470-376374-463	2-1243-375601-465	210-594-376250-257	95-1335-375509-372
Accuracy	0.9992	0.9992	0.9975	0.9954	0.9977	0.9954
Precision	0.9428	0.7941	0.0084	0.0016	0.2611	0.0664
Recall	0.4239	0.5203	0.0085	0.0042	0.4496	0.2034
TNR	0.9999	0.9998	0.9987	0.9967	0.9984	0.9964
FPR	0.00003	0.0001	0.0012	0.0032	0.0015	0.0035
Cost	29000	25460	51040	58950	33740	51500
F1	0.5849	0.6287	0.0085	0.0023	0.3304	0.1001
MCC	0.6319	0.6424	0.0072	0.0006	0.3416	0.1143

Legenda for Columns - AE: Vanilla Autoencoder, VAE: Variational Autoencoder, CAE: Contractive Autoencoder, RBM: Restricted Boltzmann Machine, IF: Isolation Forest. Legenda for rows - Threshold: The visual determination (see section 4.4.2 for anomalies, Loss: the loss obtained by the loss function of the algorithm (see section 4.5.1), Detected: The cases identified by the algorithm as anomalies, Total Frauds in Dataset: is the number of real frauds/anomalies in the dataset. TP-FP-TN-FN: is the True Positive, False Positive, True Negative, False Negative count of the transactions. The performance measurements Accuracy, Precision, Recall, True Negative Rate, False Positive Rate, Cost, F1 and MCC are described in section 4.7 ”Evaluation Measures”.

Table 4: Metrics results for large dataset relative smaller number of frauds and errors

5.3 Big Dataset More Frauds

Relevant for the discussion and the presentation of the results is table ,. Furthermore relevant information are the distribution plots of the error functions of each algorithm in the Appendix in section B.1 and the plots in the Appendix in section B.2 which show the chosen threshold for each of the algorithms as explained in section 4.4.2. Note that the same *type* of frauds and errors are present in this dataset compared to the larger dataset and relatively less anomalies in section 5.2, but the frequency of the anomalies has increased in an absolute sense (for each of the type of fraud). The only difference is the "Total Frauds in dataset" in Table 4 and Table 5, all other parameters and settings are similar.

The main differences found are that in table 5 compared to table 4 the Recall for AE and CAE has increased significantly with respectively 0.8104 (for the large dataset with relatively less anomalies 0.4239) 0.8104 (for the large dataset with relatively less anomalies 0.5203). The main cause is the lesser imbalance in the dataset and the possibility for these types of algorithms to consistently assign larger errors to the anomalies of this type.

An extremely weak performance is found for the Ensembling. The F1 Measure is 0.0311 and MCC 0.0299. It shows ensembling is not always a strong method to try to detect anomalies. It seems the individual models all have found different anomalies, which do not overlap in the rank averaging.

5.4 Overall Observations and Conclusions

This section contains more general observations on the results found in section 5.1, section 5.2 and section 5.3. This section is to provide additional insight in the results of the experiments.

5.4.1 Features

Important further experimentation shows that model selection is important as the different results demonstrate. But most important is the correct selection of features (see also step 2 in section 2.2. As mentioned 20 features are created or selected in order to cover a wide range of different frauds and errors. Four types of anomalies are contained in the dataset, three fraudulous transactions and one error type. By some trial and error six features are the most relevant to detect anomalies. If these features are not included all algorithms, including the Vanilla Autoencoder and the Contractive Autoencoder have a very low performance. So selecting and creating correct features is crucial to detecting anomalies within a financial auditing context.

The amount of features relevant for the sales transactions are in total 180 features (see also section 4.2). It requires a large amount of effort to obtain 20 relevant features using professional judgement to cover a wide range of frauds and errors. As mentioned in section 3.1, there is a lack of knowledge of data scientific methods in financial auditing and also thus the conversion of the qualitative risk judgements to a (set of) feature(s). The time the researcher of this paper invested into feature construction and creation was substantial and removed any speed advantages of using automation. This feature selection and creation however can be improved if more financial auditors obtain data science knowledge and best practices are created. Best practices in which combinations of features are useful to detect certain specific frauds and errors.

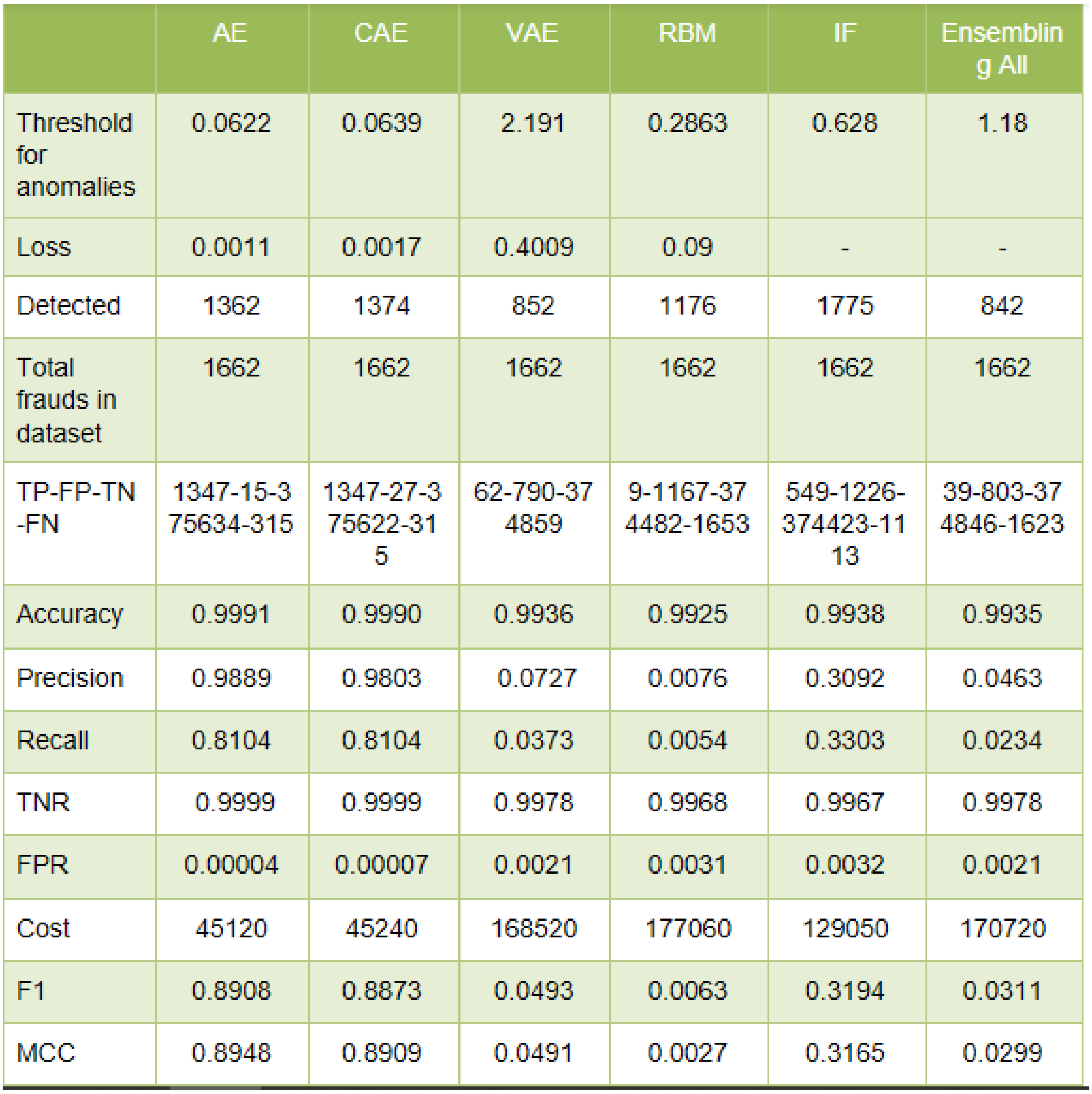
It can be observed that the Variational Autoencoder (VAE), the Restricted Boltzmann Machines (RBM) and the Isolation Forest (IF) algorithms are performing not as strong. Some discussion on the possible reasons for this are given in section 5.1. It indicates the Vanilla Autoencoder and the Contractive Autoencoder seem to have a more general purpose characteristic to observe anomalies than the other algorithms with respect to the different types of input.

5.4.2 Anomaly Detection

Another important aspect in financial auditing is the concept of *materiality*. This is governed in the regulations of the "International Standards Auditing" (ISA 320 specifically). This in an overgeneralized sense means the financial auditor needs to detect larger frauds and errors. In the results found for Vanilla Autoencoder (AE) and Contractive Autoencoder (CAE) the algorithms both are consistently able to obtain the larger frauds and errors. The other algorithms, for the larger dataset, are not necessarily able to obtain all larger frauds and errors.

Not all anomalies, frauds or errors, are detected for each algorithm. From a financial auditing perspective it is important that at least every *type* of anomaly in the dataset is detected. There are four anomaly types, three frauds and one error. When evaluating the types of anomalies found for the Vanilla Autoencoder (AE) and Contractive Autoencoder (CAE) all types of frauds and the error are all detected *at least once* in the anomalies. This is important for financial auditing purposes, since the auditor can evaluate the found anomalies by the algorithm and evaluate every found fraud and error and then perform a more extensive search for the type of anomaly found by the algorithm (if considered necessary).

Lastly there is a substantive qualitative advantage of using a data driven auditing approach. At least with the AE and CAE algorithms a higher completeness of fraud and error detection is evient. As mentioned in the paragraph above, every type of fraud and error is detected at least once (with the AE and CAE algorithm). One type of fraud: the so called "Carroussel Fraud" is more strongly present within this dataset (with the amount of transactions related to it and the size of each transaction). Within a normal audit this type of fraud would likely be detected. The error, which concerns multiple transactions are not booked in the correct year, would also be most likely detected by the auditor. The procedure for checking this error is quite simplistic and easily done within every single audit. However the two other frauds (so three frauds and one error is four types of anomalies) are most likely not detected (an employee fraud and uncommon transactions with foreign countries such as Saudi Aurabia). This is due to the fact that the type of transactions are very uncommon and not thought of by the qualitative risk judgement of the financial auditor, but do come up using the AE and CAE algorithms over different dataset contexts.



	AE	CAE	VAE	RBM	IF	Ensembling All
Threshold for anomalies	0.0622	0.0639	2.191	0.2863	0.628	1.18
Loss	0.0011	0.0017	0.4009	0.09	-	-
Detected	1362	1374	852	1176	1775	842
Total frauds in dataset	1662	1662	1662	1662	1662	1662
TP-FP-TN-FN	1347-15-375634-315	1347-27-375622-315	62-790-374859	9-1167-374482-1653	549-1226-374423-1113	39-803-374846-1623
Accuracy	0.9991	0.9990	0.9936	0.9925	0.9938	0.9935
Precision	0.9889	0.9803	0.0727	0.0076	0.3092	0.0463
Recall	0.8104	0.8104	0.0373	0.0054	0.3303	0.0234
TNR	0.9999	0.9999	0.9978	0.9968	0.9967	0.9978
FPR	0.00004	0.00007	0.0021	0.0031	0.0032	0.0021
Cost	45120	45240	168520	177060	129050	170720
F1	0.8908	0.8873	0.0493	0.0063	0.3194	0.0311
MCC	0.8948	0.8909	0.0491	0.0027	0.3165	0.0299

Legenda for Columns - AE: Vanilla Autoencoder, VAE: Variational Autoencoder, CAE: Contractive Autoencoder, RBM: Restricted Boltzmann Machine, IF: Isolation Forest. Legenda for rows - Threshold: The visual determination (see section 4.4.2 for anomalies, Loss: the loss obtained by the loss function of the algorithm (see section 4.5.1), Detected: The cases identified by the algorithm as anomalies, Total Frauds in Dataset: is the number of real frauds/anomalies in the dataset. TP-FP-TN-FN: is the True Positive, False Positive, True Negative, False Negative count of the transactions. The performance measurements Accuracy, Precision, Recall, True Negative Rate, False Positive Rate, Cost, F1 and MCC are described in section 4.7 ”Evaluation Measures”.

Table 5: Metrics results for large dataset with more frauds and errors

6 Conclusion & Discussion

6.1 Summary and Conclusions

A first exploration in combining data science and (financial) auditing is made within this paper. Different unsupervised algorithms are explored and applied within an audit simulation which provide real-life synthetic data which simulate a real company. This is highly unique data make it possible to research the possibility of using data scientific methods within an auditing setting. The absence of usable data are the main reason why almost no research exists in the financial auditing literature combined with data science [1]. The unsupervised algorithms explored are (multiple) Autoencoders, Restricted Boltzmann Machines and Isolation Forest. These algorithms are also combined within an ensemble by using an average ranking technique. The strongest results are found using a Vanilla Autoencoder and Contractive Autoencoder across different datasizes and anomaly/normal distributions. The main question posed in this paper is:

"To what extent can unsupervised algorithms detect (material) financial frauds or errors in a financial auditing context?"

The main question is answered by first answering the subquestions formulated in section 1.1.

6.1.1 Which algorithms can be taken into consideration to use, and why?

An important consideration in selecting algorithms is understanding the financial auditing environment. The financial auditor has to perform a task in detecting *material* frauds and errors in the financial statements of a company. The financial auditor has no existing labels providing a ground truth. In addition, the types of anomalies most common for the auditor are point anomalies (unique and extreme anomalies) and group anomalies (transactions which together are anomalous). Effective fitting types of algorithms are unsupervised neural network algorithms to perform within this context. The selected unsupervised neural network algorithms are the Vanilla Autoencoder (AE), the Variational Autoencoder (VAE), the Contractive Autoencoder (CAE) and the Restricted Boltzmann Machine (RBM). Besides these algorithms, Isolation Forest (IF) is explored. Using a different type of algorithm also provides additional insight to the performance of the algorithms.

6.1.2 How do the algorithms perform across multiple evaluation measures?

The best performing algorithms are the CAE and the AE. These algorithms have a relatively high F1 Measure and higher Matthews Correlation Coefficient compared to the other algorithms. The VAE has a mixed result pattern but in general is not a strong model and scores low on precision and recall. RBM consistently has a low performance for recall, precision. This is due to the nature of the fit of the real valued input features and the way the RBM functions. The IF algorithm provides a minimum performance but not strong enough to detect anomalies from all the different types of frauds and errors in the dataset. Mixed results (high and low performance) are found using average ranking Ensembling in which all algorithms are combined. In summary: the CAE and AE perform best on the important evaluation measures as Precision, Recall, Matthews Correlation Coefficient and the F1 Measure making these algorithms the most general purpose algorithms best fitted for the financial auditing context.

6.1.3 Can a robustness of results be found per algorithm using different fraud and error contexts?

The results for the dataset of 40,829 records with 0,639% of frauds and/or errors. The performance of the algorithms is overall stronger than the larger datasets. On evalation measures as Precision, Recall, F1 and Matthew Correlation Coefficient the algorithms perform better than the larger datasets. The strongest performing models on the smaller dataset are the AE (with a F1 Measure of 0.8933 and Matthew Correlation Coefficient of 0.8934), the CAE (a F1 Measure of 0.9196 and Matthew Correlation Coefficient of 0.9193) and the Ensembling of all algorithms together (an F1 Measure of 0.6979 and Matthew Correlation Coefficient of 0.7249).

The results on the larger datasets are of a lower performance than on the smaller dataset. There are two larger datasets, which are identical, but they have a different mix of anomalies, one larger dataset of 377,311 with a fraud/error percentage of 0,124%, and a larger dataset of 377,311 percentage of 0,440%. On the larger dataset the best performing algorithms are AE (F1 measure of 0.5849 and Matthew Correlation Coefficient of 0.6319) and CAE (F1 measure of 0.6287 and Matthew Correlation Coefficient of 0.6424). All other algorithms have a very low performance on this dataset. For the larger dataset with more anomalies the AE (a F1 measure of 0.8908 and Matthew Correlation Coefficient of 0.8948) and CAE (a F1 measure of 0.8873 and Matthew Correlation Coefficient of 0.8909).

It is noted that the performance of the AE and CAE decline with the larger dataset which has an anomaly percentage of 0,124%. For a financial auditing context, it is not specifically worrying since the total amount of fraud and error is already not that high. Whereas the extremity of the imbalance of frauds and errors becomes less then the algorithms of AE and CAE work well.

It is an interesting observation that the IF algorithm performs structurally consistent over all different dataset contexts (within all context provides a F1 measure and Matthew Correlation Coefficient of around 0,30). Ensembling performs decently well on the smaller dataset, but has the worst performance when used on the larger dataset. Demonstrating that using ensembling is not always a safe method to use to detect anomalies to combine the strength of different algorithms.

To answer the main question:

"To what extent can unsupervised algorithms detect (material) financial frauds or errors in a financial auditing context?"

The first indications in implementing known unsupervised algorithms demonstrate the ability to detect frauds or errors is strongly possible. Overall the Vanilla Simple Autoencoder is able to outperform all other algorithms on mostly all evaluation metrics. Unsupervised neural network algorithms in general seem to be able to take over subtasks of a financial auditor and perform strongly by integrally audit the entire (sales) dataset. If features are constructed in alignment with a decent qualitative risk analysis by a financial auditor, the chances are that the unsupervised neural network algorithms will be able to extract useful information for the auditor.

In addition, the results seem to indicate the lower the number of frauds/errors are the lower the precision and recall is. This is however not of a large consequence for the auditor. Since it is the case that the number of frauds and errors are not significant enough to be considered a problematic misstatement for the financial statements of the company.

The practical implications for the financial auditing are substantive. A first exploration for using unsupervised algorithms in auditing is made with satisfying results. This paper can be a first step in the direction of creating a new subfield within the financial auditing profession where individuals with financial auditing domain knowledge and understanding of data science can perform audits over much larger amounts of data. As a consequence it provides a more complete analysis of possible frauds and errors. Multiple practical aspects within related aspects of the profession, such as the standardization of the general ledger structure over multiple bookkeeping systems (twinfield, exact, oracle etc.), feature construction can also standardized. Creating features of interest for the audit can then be made much easier and be inputted faster in the unsupervised algorithm. This could all mean a new future where the called '4th industrial revolution' can also be feasibly and effectively start to happen within the financial auditing profession.

6.2 Limitations

While researching this subject some limitations to the conclusions must be made.

An audit simulator is used. Even though it is one of the more reliable methods to truly do research with, one has to take into consideration the audit simulator always has some limitations in comparison with the real world applications. In addition, some type of frauds (not implemented in the audit simulator) might still be harder to detect using the unsupervised algorithms proposed.

The audit case used is called laptop world and is a company which has the typology of 'trade'. From a financial auditing point of view, when classifying the company to it's type, it is the most simple form which a company can take, since products come in and products go out. It does not for instance provide services or produces goods. Different type of frauds and errors can occur which are different and sometimes harder to detect.

Not all class imbalance settings are tried due to time and budget constraints. An interesting case to try to see is how the unsupervised algorithms perform by using a larger percentage of frauds or errors. Even though it is quite rare to see many frauds, it is the type of context a financial auditor would need in order to be able to work the unsupervised algorithms.

6.3 Future Work and Suggestions

An aspect for further research might be solving the extreme imbalance class problem for unsupervised settings. For instance multiple oversampling and undersampling methods exist for supervised settings. No true heuristic or method is developed which can reliably counter the extreme imbalance problem. Further research might devise methods for solving that problem.

Another interesting research angle might be looking at methods of combining the algorithms which are explored in this paper. This might be through pre-training methods, or creating larger ensembles. The results show the way the algorithms learn are different from each other. The combined power might provide more strength to the unsupervised methods.

As mentioned in the limitations, this research focuses on a single sector called 'trading'. Also the sales process is mainly evaluated. Different types of frauds, errors, datastreams and company sectors can be explored in a similar fashion in order to see what the results might be for the financial auditing context.

It might be argued that some form of semi-supervised learning might be possible. By using the errors and frauds detected (by hand) in the previous years, it can be a way of obtaining stronger results for the current year under audit. Looking at the possibility of using semi-supervised learning withing the context of financial auditing can be beneficial.

The explainabilty of the (true) positive cases which the unsupervised algorithms predict. One aspect which can help drastically in the work of the financial auditor, using the output of the data, is some indication why the unsupervised algorithms exported the specified cases as a fraud or an error. Central to the work of a financial auditor is to be able to report on why mistakes are selected, and be judged in light of the prevailing regulations. It might be time consuming if the auditor needs to manually check every output of the unsupervised algorithm.

References

[1] T. J. O. T. S. A. Geppa, M. K. Linnenlueckeb. Big data techniques in auditing research and practice: Current trends and future opportunities. *Journal of Accounting Literature*, 40:102 – 115, 2018.

[2] H. Abdi. Factor rotations in factor analyses. encyclopedia for research methods for the social sciences. *Sage: Thousands Oaks*, pages 792–795, 2003.

[3] S. Agrawal. Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60(10):708–713, 2015.

[4] M. Albashrawi. Detecting financial fraud using data mining techniques: A decade review from 2004 to 2015. *Journal of Data Science*, 14:553 – 570, 2016.

[5] M. D. O. Brandas, C. Muntean. Intelligent decision support in auditing: Big data and machine learning approach. 2019.

[6] T. G. Dietterich. Ensemble methods in machine learning. in international workshop on multiple classifier systems. *In International workshop on multiple classifier systems*, pages 1–15, 2000.

[7] D. S. . R. B. Dutta, I. Detecting financial restatements using data mining techniques. *Expert Systems with Applications*, 90:374 – 393, 2017.

[8] N. T. G Menardi. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28(1):92, 2014.

[9] . U. S. Goldstein, M. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4), 2016.

[10] X. G. A. P. B. U. C. B. S. M. A. L. I. Higgins, L. Matthey. Early visual concept learning with unsupervised deep learning. *Procedia Computer Science*, 2016.

[11] S. G. Jang, E and B. Poole. Categorical reparameterization with gumbel-softmax. *preprint arXiv*, 2016.

[12] M. W. Kingma, D.P. Auto-encoding variational bayes. *arXiv preprint*, 2013.

[13] T. K. M. . Z. Z. H. Liu, F. T. Isolation-based anomaly detection. *Transactions on Knowledge Discovery from Data*, 6(1):1, 2012.

[14] S. R. Chalapathy. Deep learning for anomaly detection: A survey. 2019.

[15] X. M. X. G. Y. B. S. Rifai, P. Vincent. Contractive auto-encoders: Explicit invariance during feature extraction. *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 833–840, 2011.

[16] . V. M. A. Thiprungsri, S. Cluster analysis for anomaly detection in accounting data: An audit approach. *The International Journal of Digital Accounting Research*, 11:69 – 84, 2011.

[17] C. R. J. . S. J. Zimek, A. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *ACM Sigkdd Explorations Newsletter*, 15(1):11–22, 2014.

Appendices

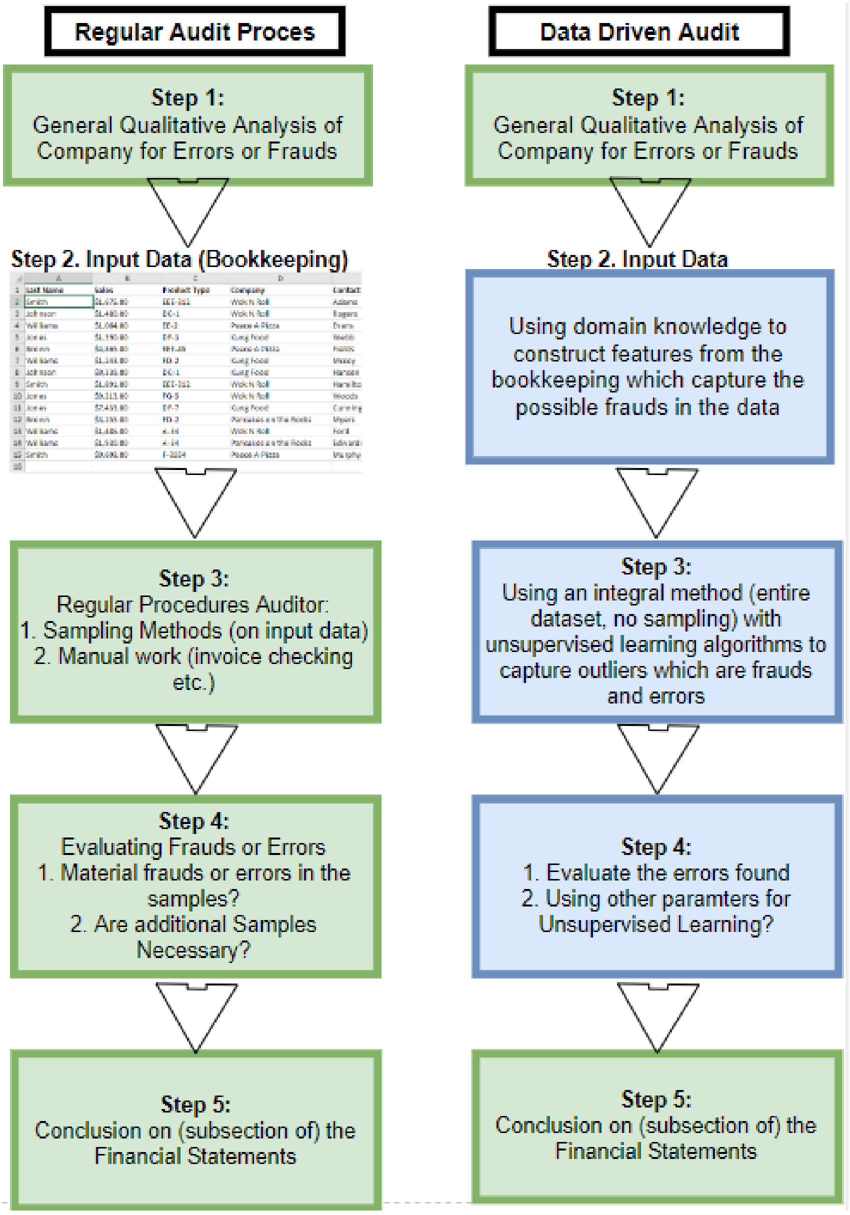

Regular Audit Proces
Data Driven Audit
Step 1:
General Qualitative Analysis of Company for Errors or Frauds
Step 1:
General Qualitative Analysis of Company for Errors or Frauds
Step 2. Input Data (Bookkeeping)
Step 2. Input Data
Using domain knowledge to construct features from the bookkeeping which capture the possible frauds in the data
Step 3:
Regular Procedures Auditor:
1. Sampling Methods (on input data)
2. Manual work (invoice checking etc.)
Step 3:
Using an integral method (entire dataset, no sampling) with unsupervised learning algorithms to capture outliers which are frauds and errors
Step 4:
Evaluating Frauds or Errors
1. Material frauds or errors in the samples?
2. Are additional Samples Necessary?
Step 4:
1. Evaluate the errors found
2. Using other paramters for Unsupervised Learning?
Step 5:
Conclusion on (subsection of) the Financial Statements
Step 5:
Conclusion on (subsection of) the Financial Statements

Figure 8: General Auditing Process

A Plots for Small Dataset

A.1 Distributions Fraud vs. non Fraud

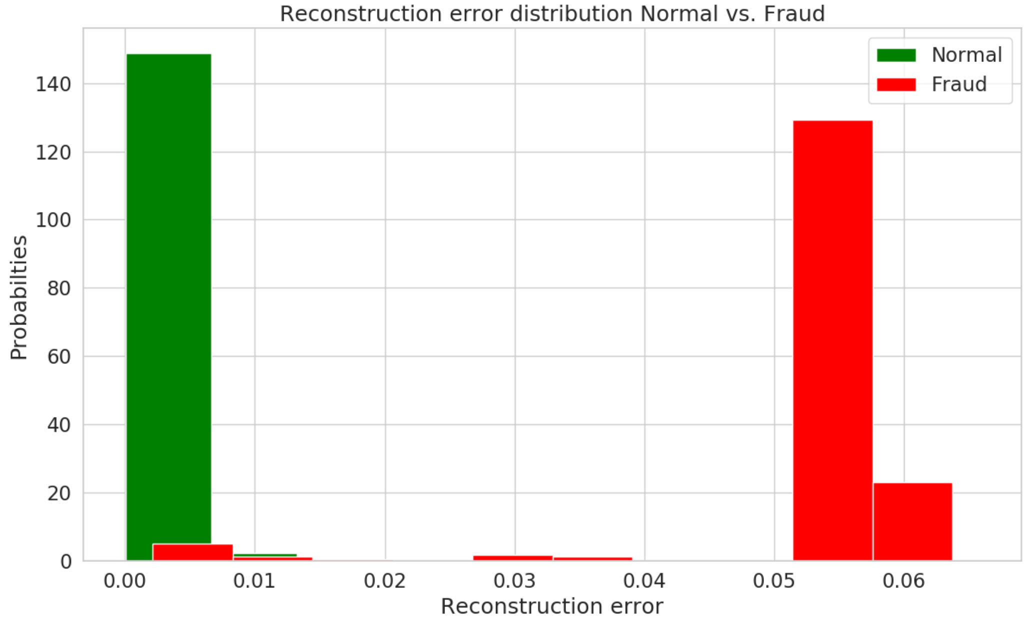


Figure 9: Distribution Small Dataset - Vanilla Autoencoder

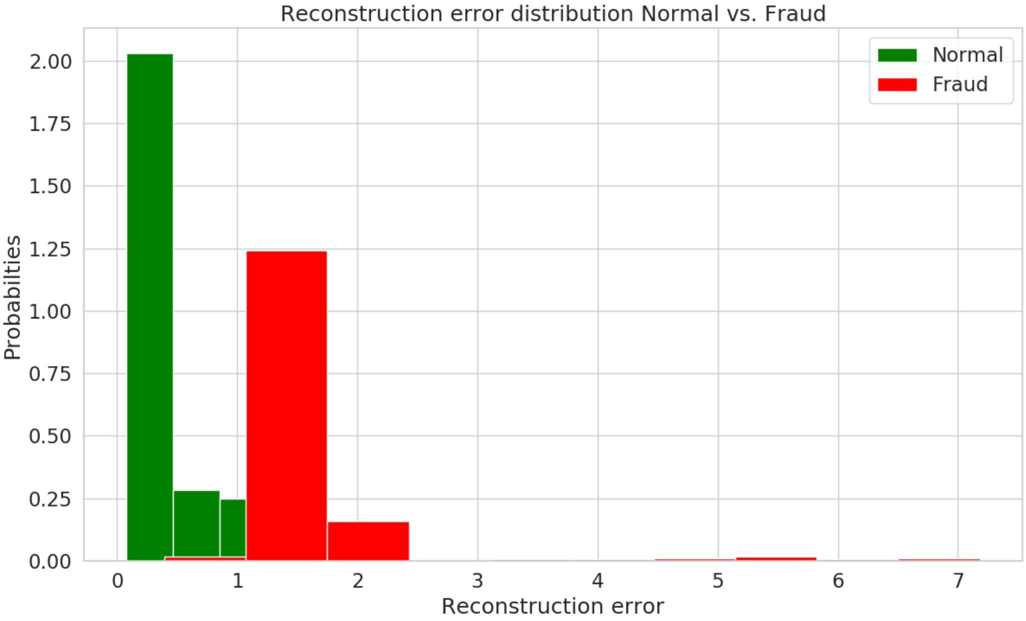


Figure 10: Distribution Small Dataset - Variational Autoencoder

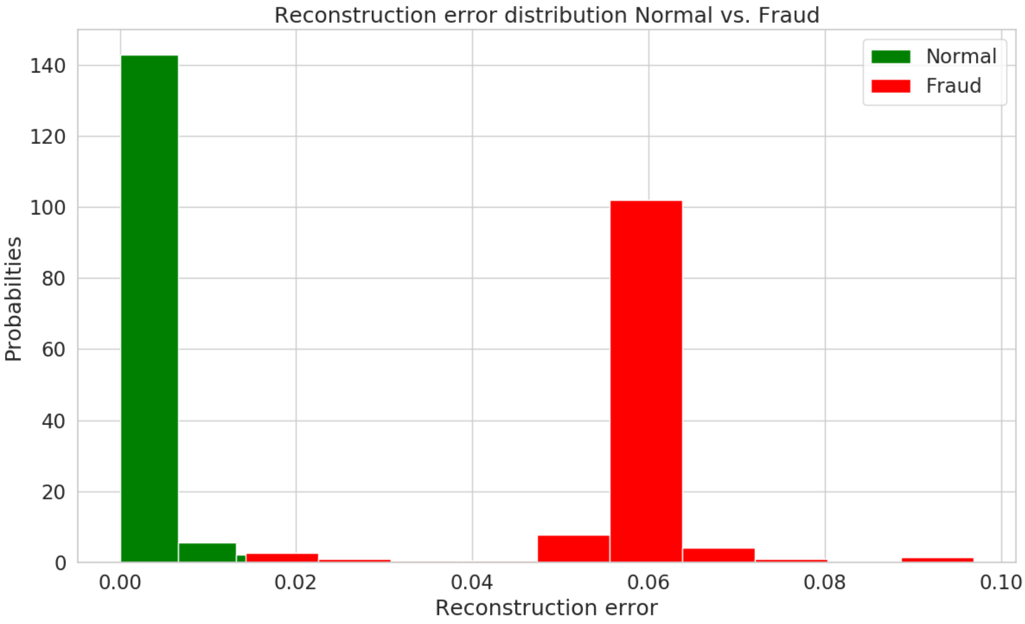


Figure 11: Distribution Small Dataset - Contractive Autoencoder

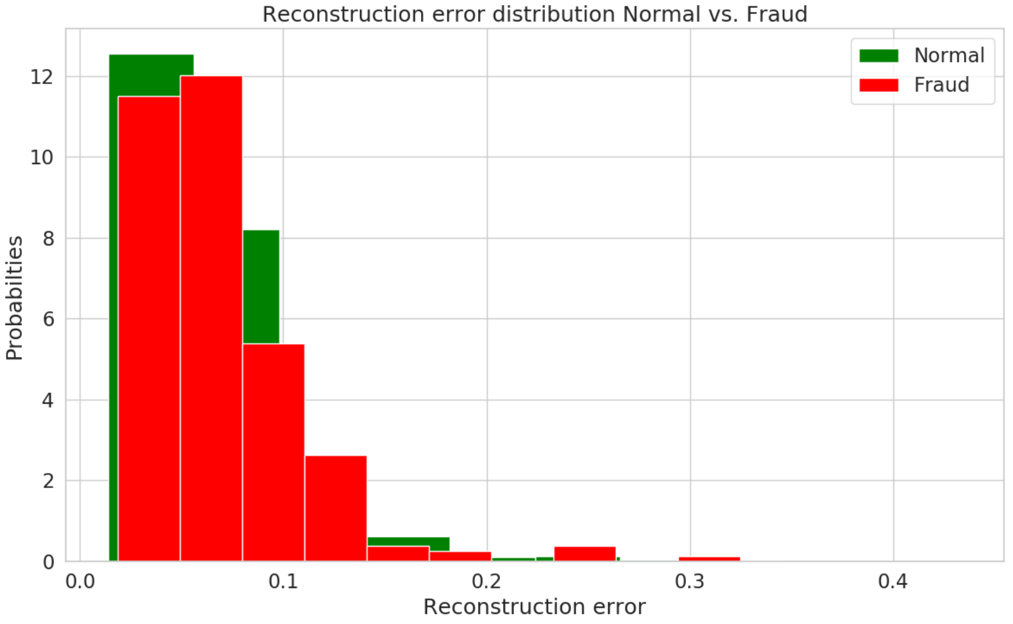


Figure 12: Distribution Small Dataset - Restricted Boltzmann Machines

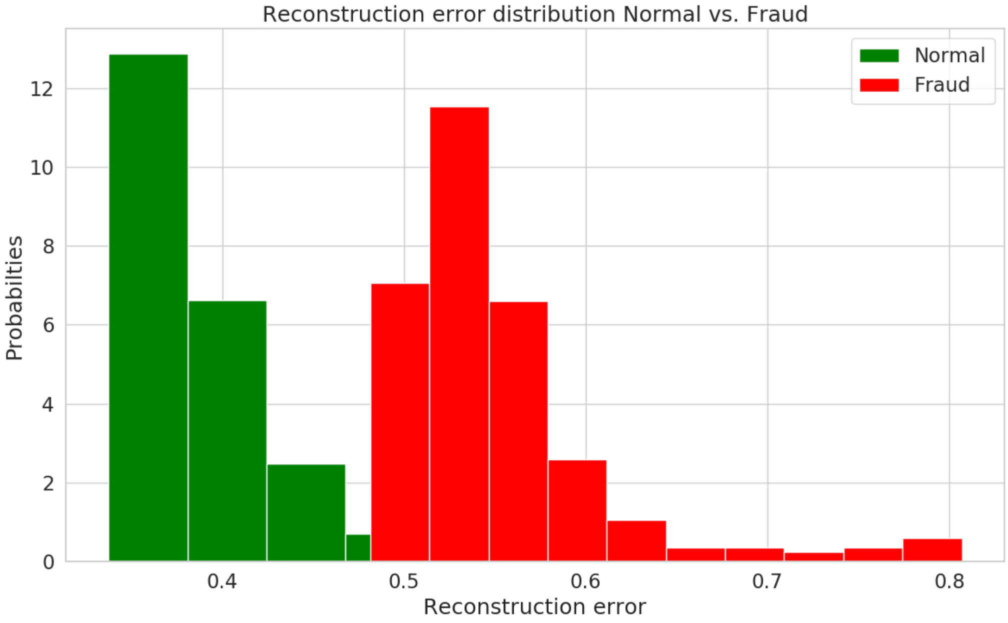


Figure 13: Distribution Small Dataset - Isolation Forest

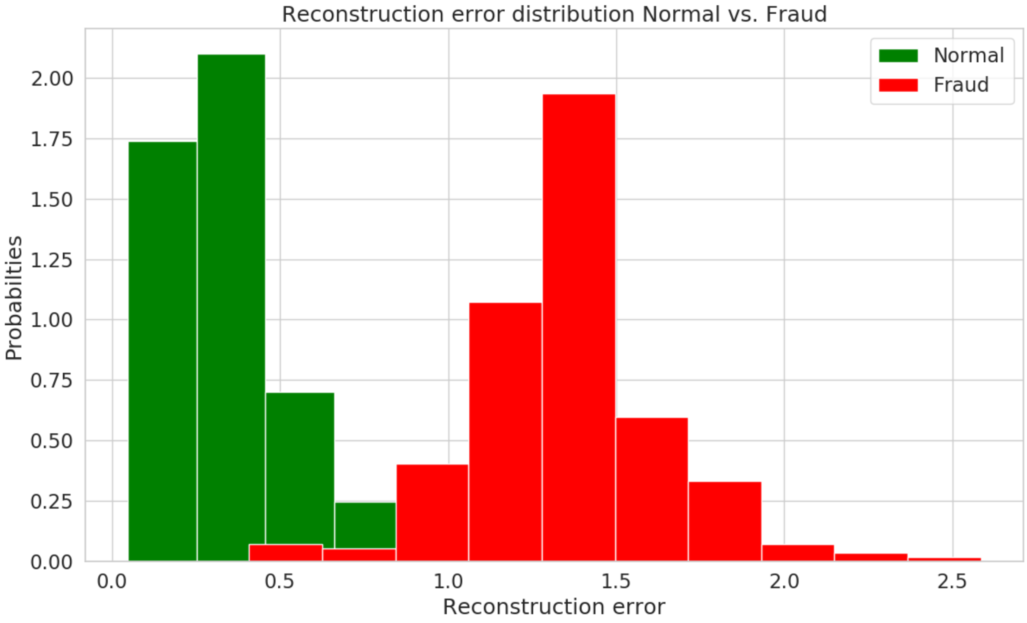


Figure 14: Distribution Small Dataset - Ensemble

A.2 Visual Threshold

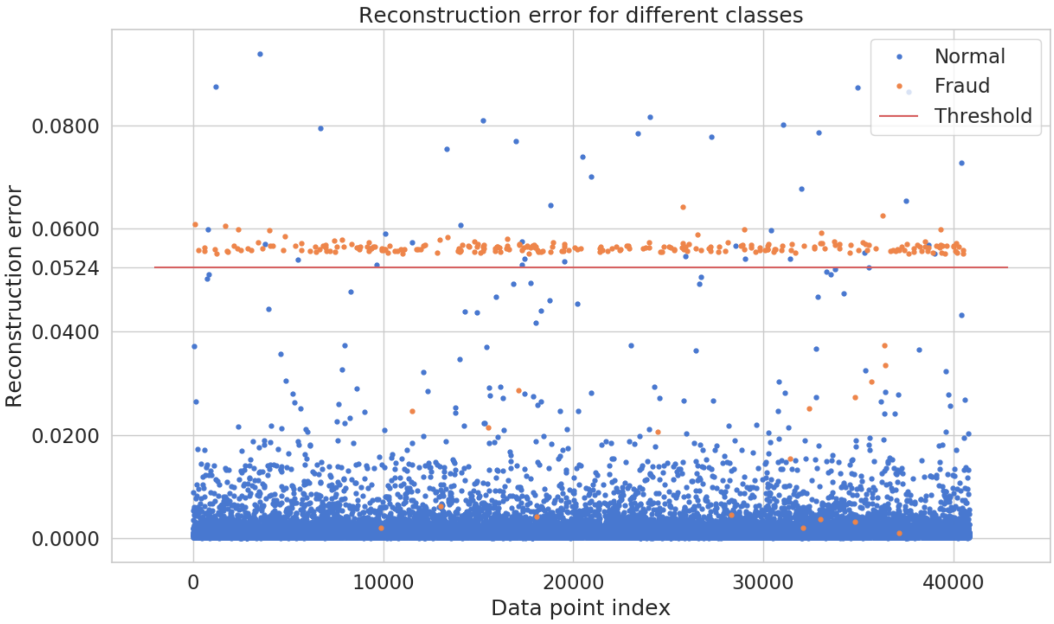


Figure 15: Threshold Determination Small Dataset - Vanilla Autoencoder

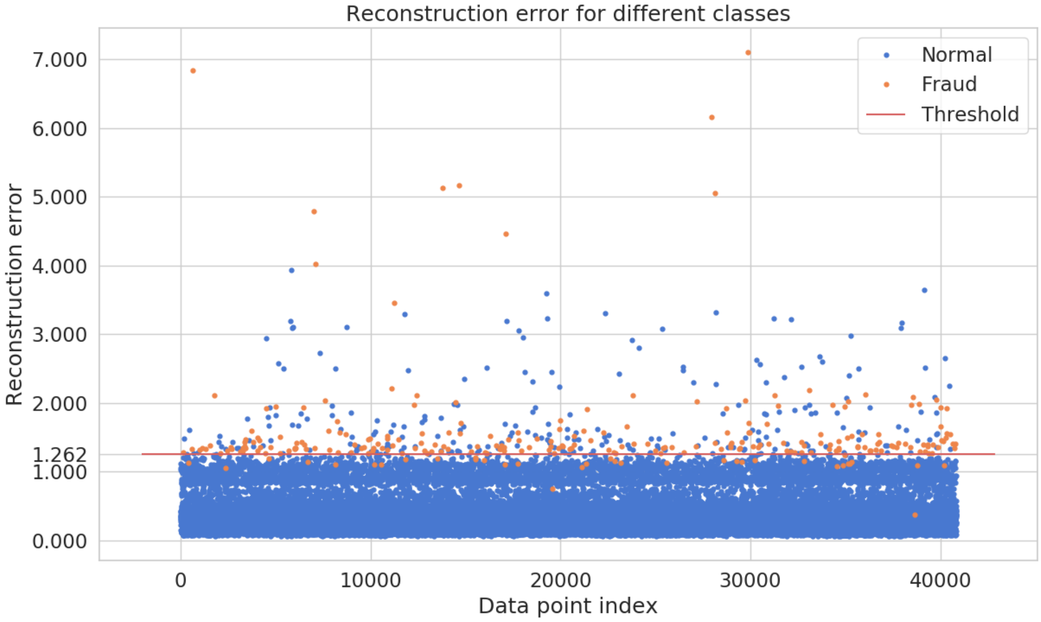


Figure 16: Threshold Determination Small Dataset - Variational Autoencoder

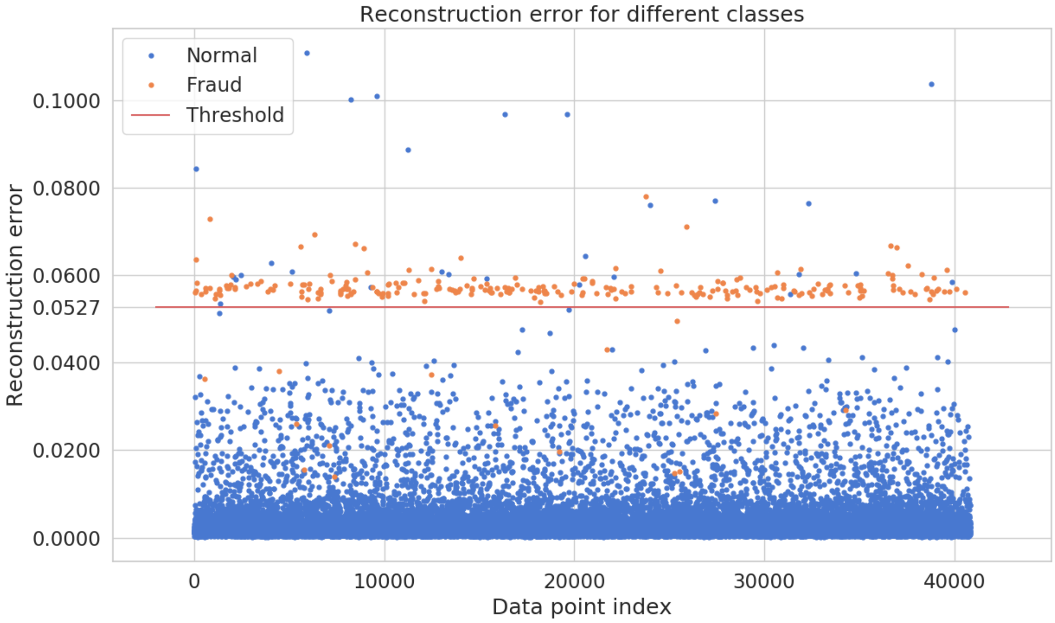


Figure 17: Threshold Determination Small Dataset - Contractive Autoencoder

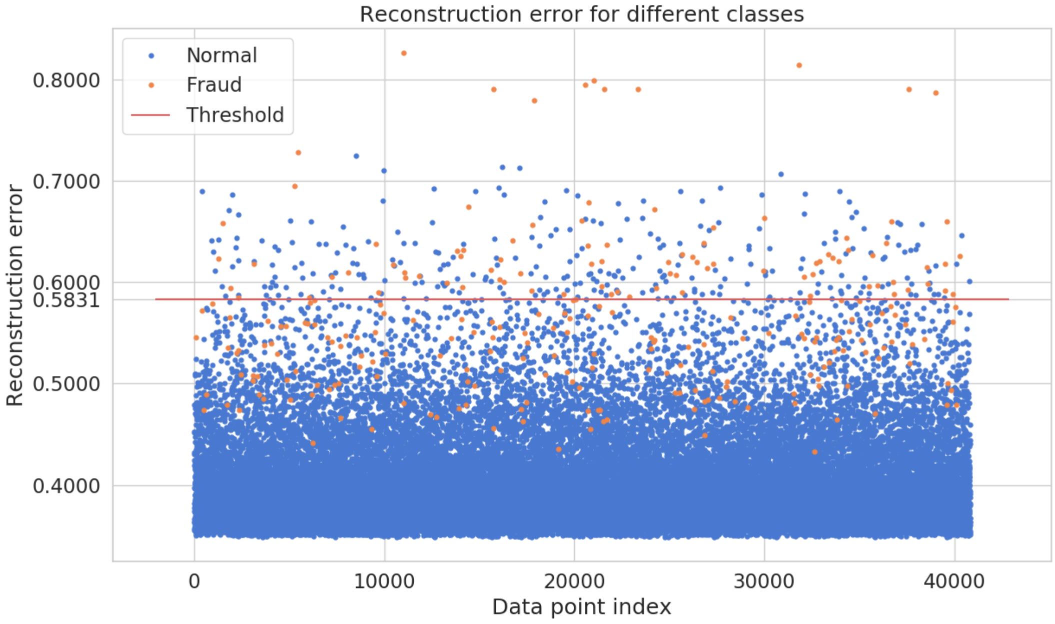


Figure 18: Threshold Determination Small Dataset - Isolation Forest

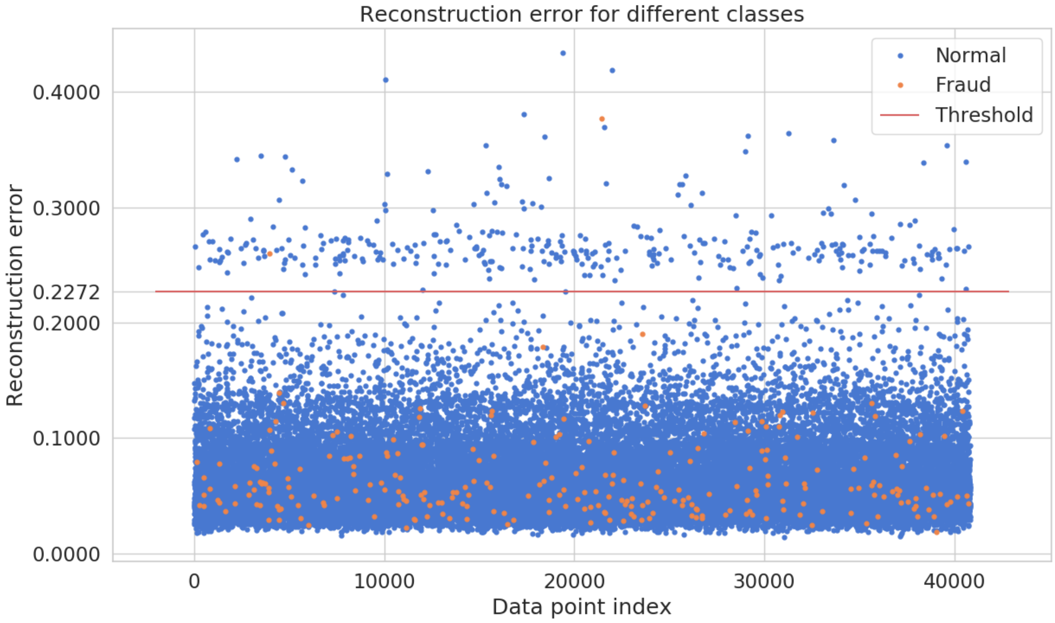


Figure 19: Threshold Determination Small Dataset - Restricted Boltzmann Machines

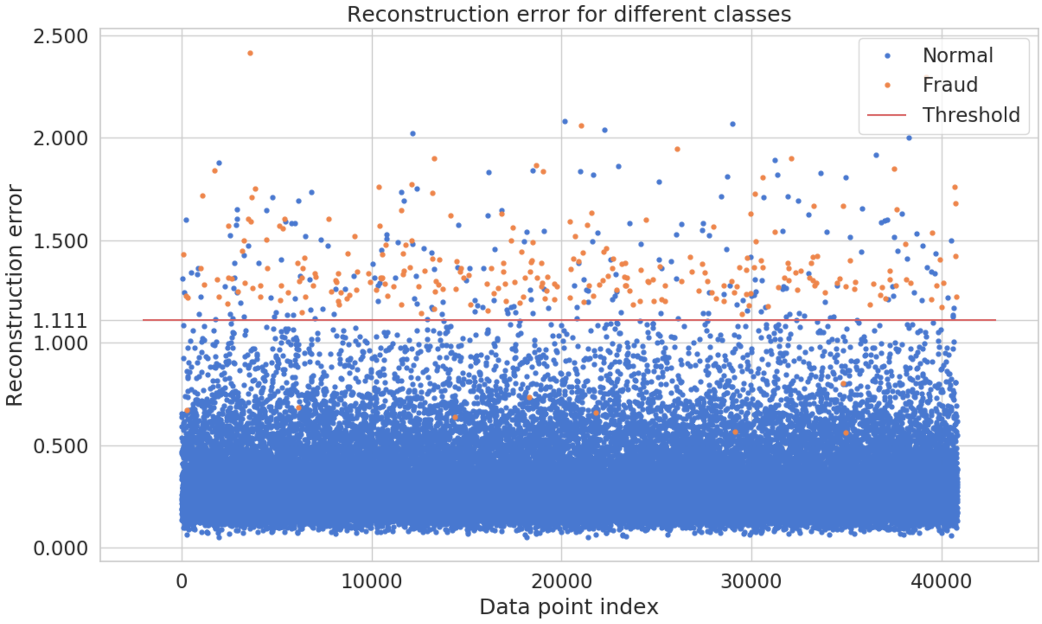


Figure 20: Threshold Determination Small Dataset - Ensemble Modelling

B Plots for Large Dataset and Relatively Small number of Frauds

B.1 Distributions Fraud vs. non Fraud

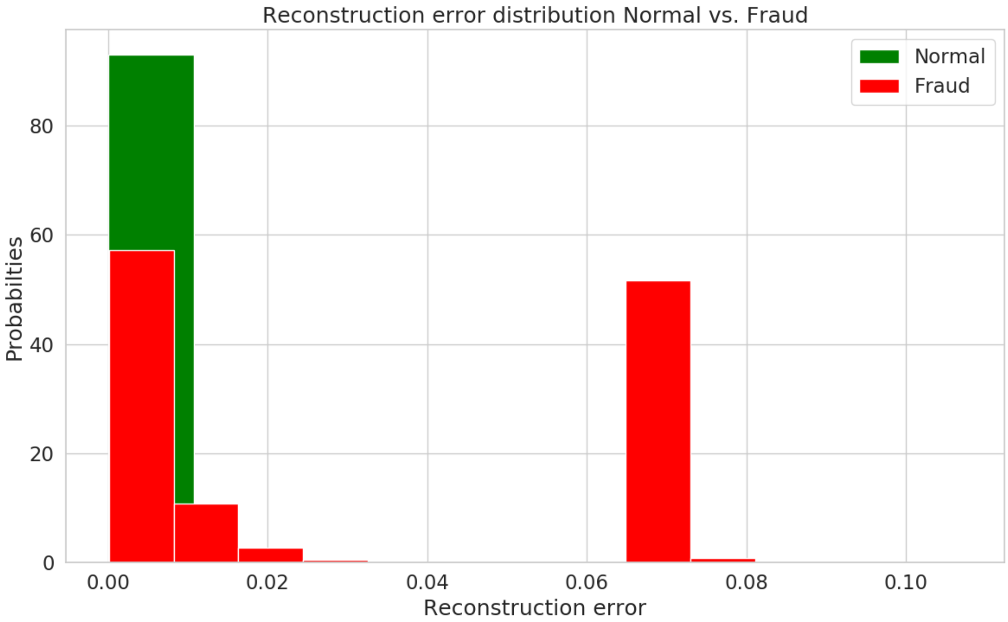


Figure 21: Distribution large Dataset - Vanilla Autoencoder

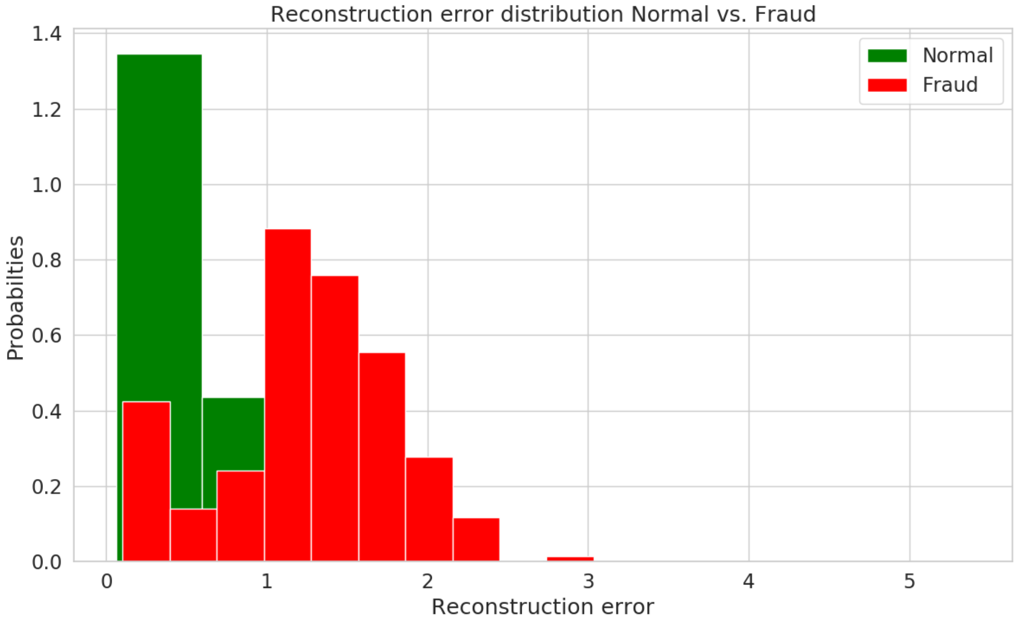


Figure 22: Distribution large Dataset - Variational Autoencoder

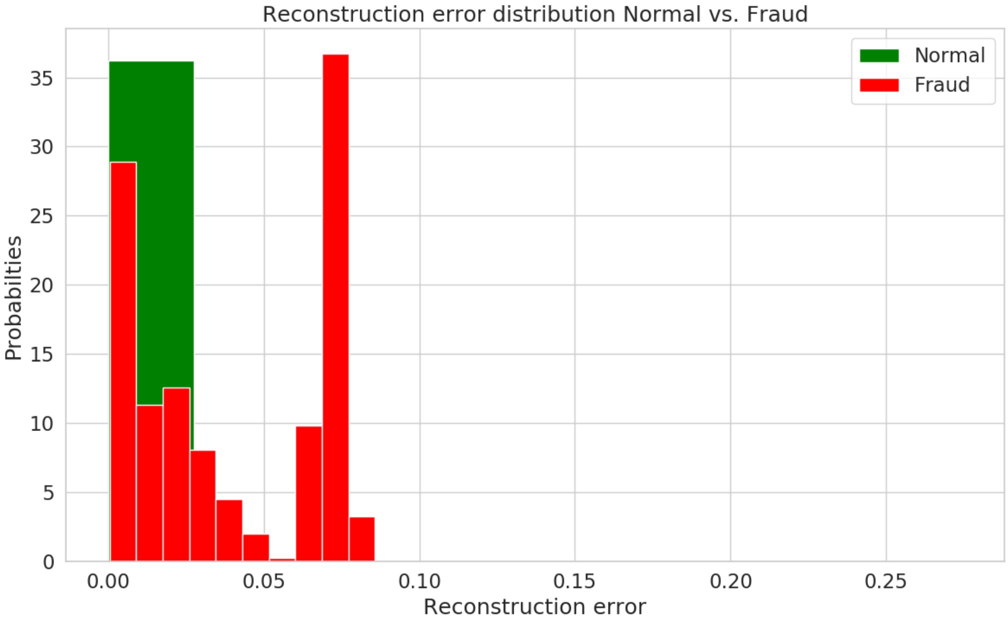


Figure 23: Distribution large Dataset - Contractive Autoencoder

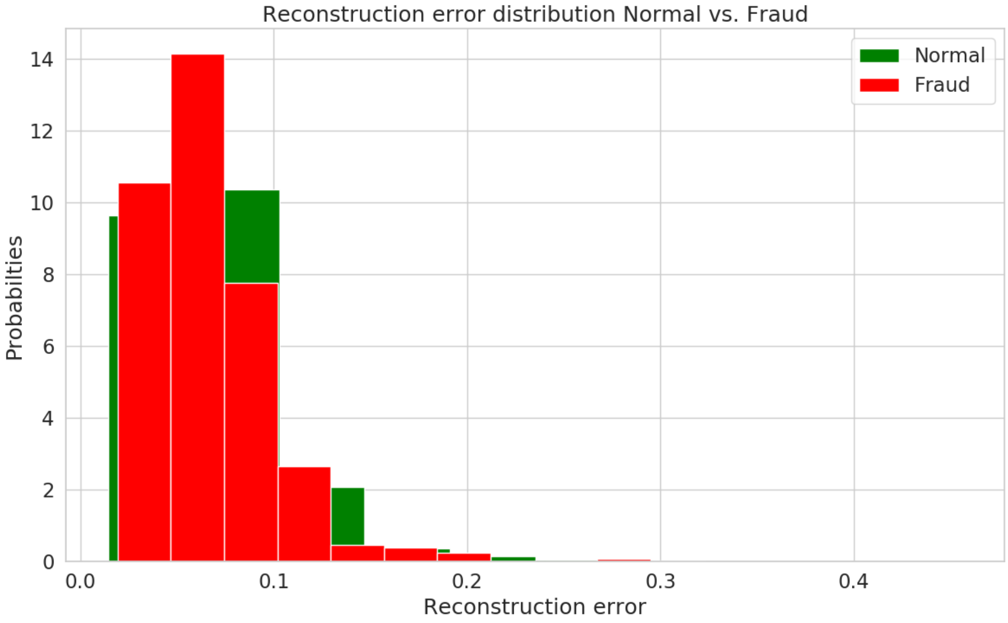


Figure 24: Distribution large Dataset - Restricted Boltzmann Machines

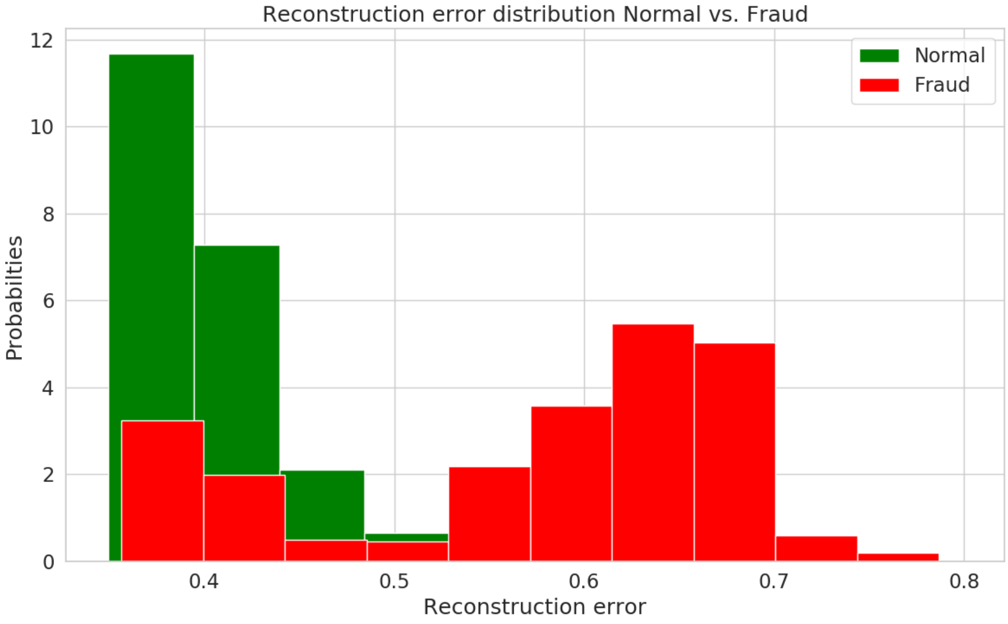


Figure 25: Distribution large Dataset - Isolation Forest

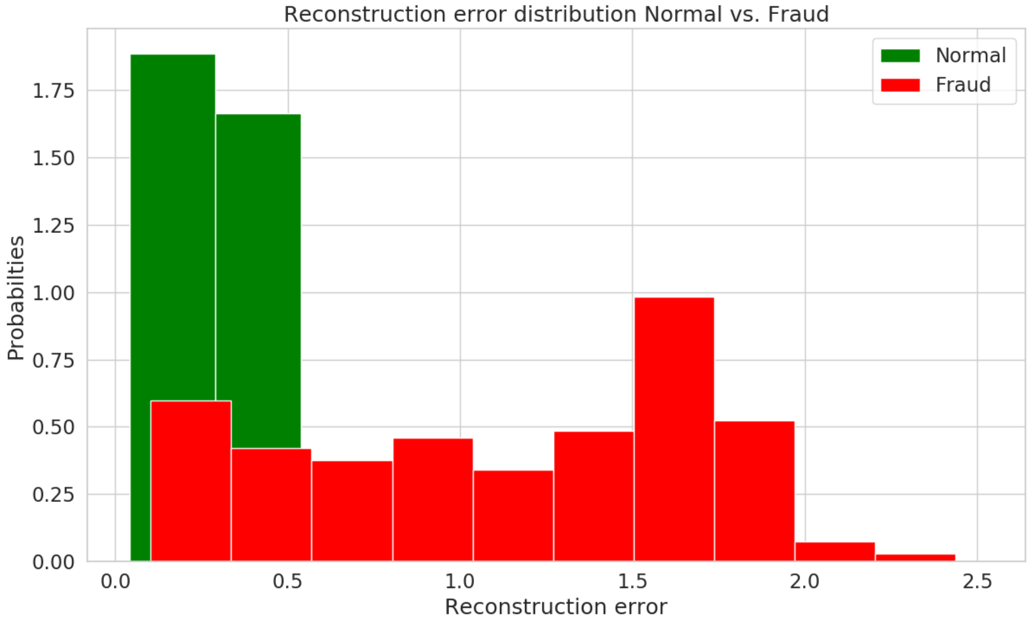


Figure 26: Distribution large Dataset - Ensemble

B.2 Visual Threshold

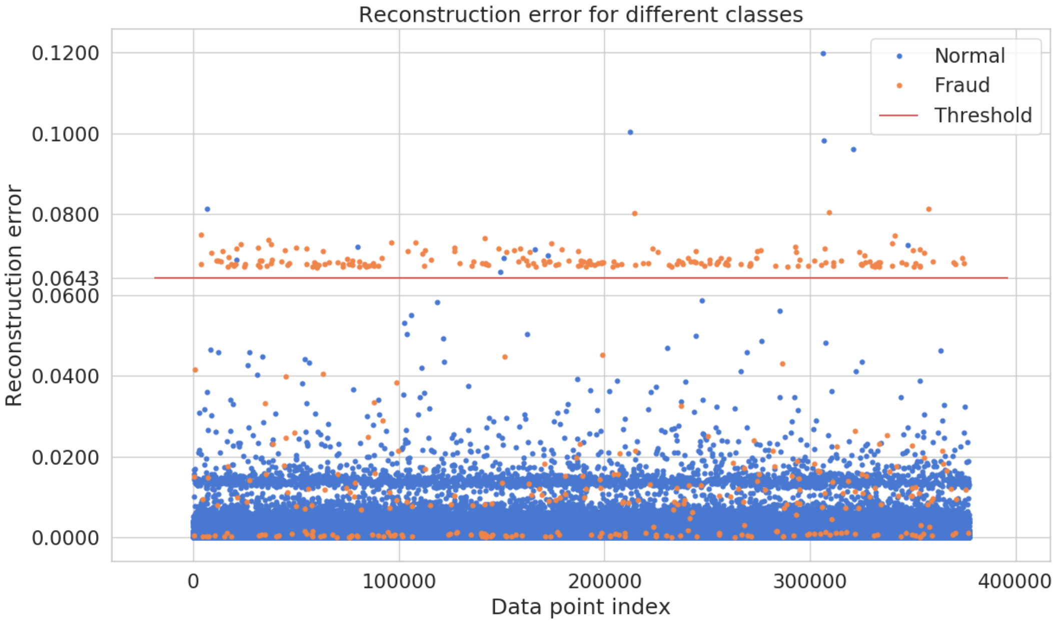


Figure 27: Threshold Determination large Dataset - Vanilla Autoencoder

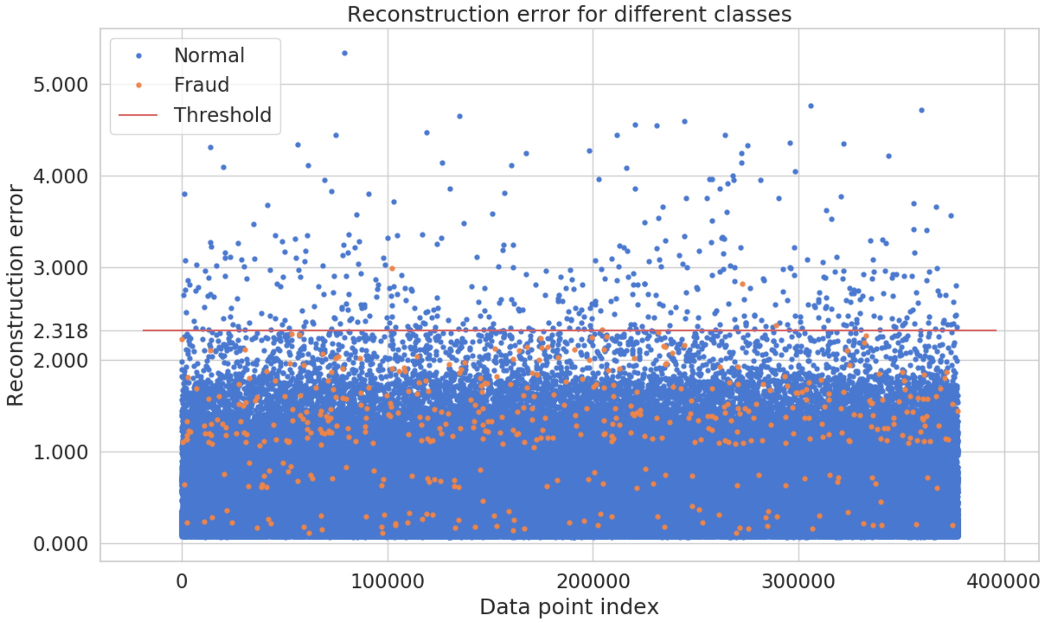


Figure 28: Threshold Determination large Dataset - Variational Autoencoder

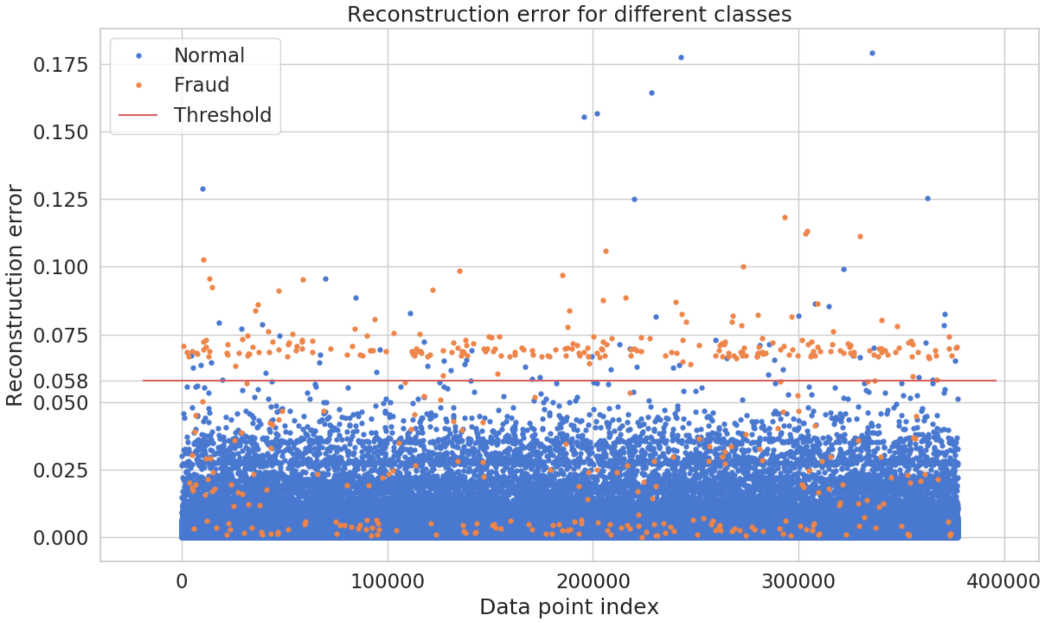


Figure 29: Threshold Determination large Dataset - Contractive Autoencoder

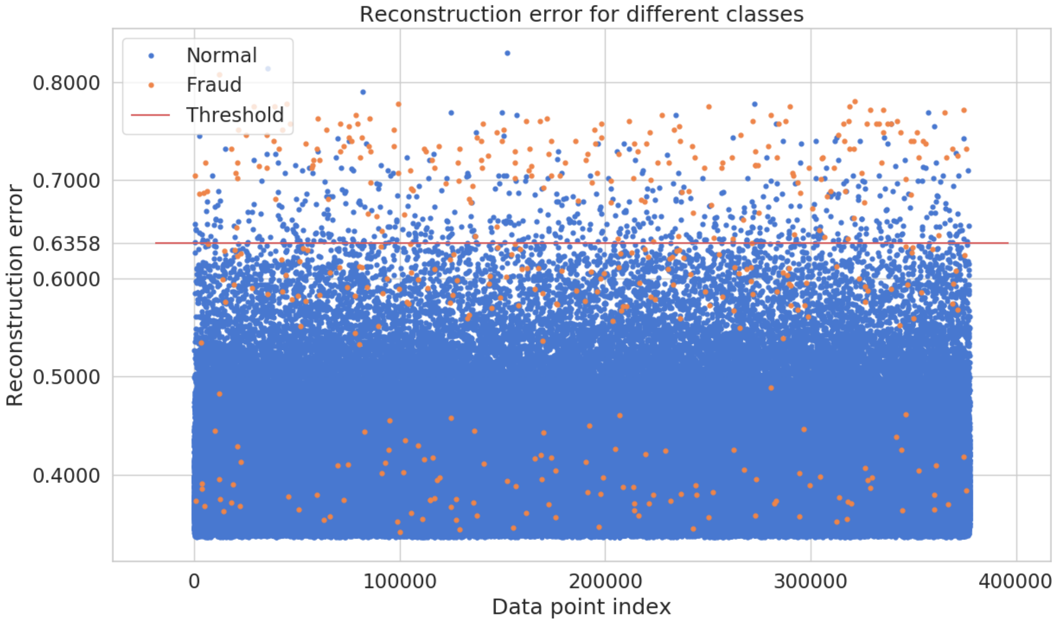


Figure 30: Threshold Determination large Dataset - Isolation Forest

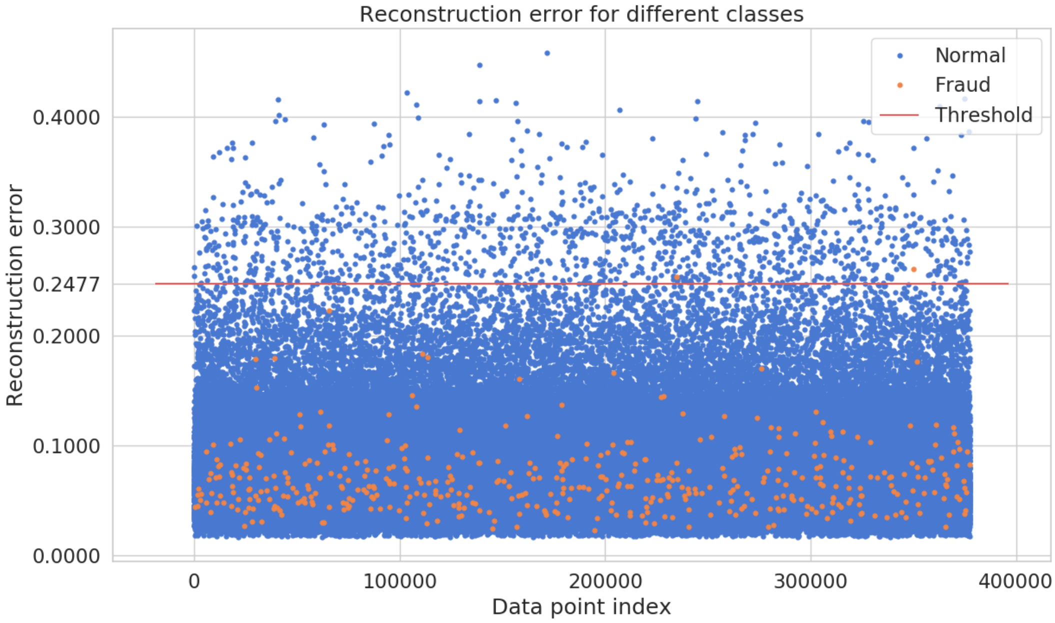


Figure 31: Threshold Determination large Dataset - Restricted Boltzmann Machines

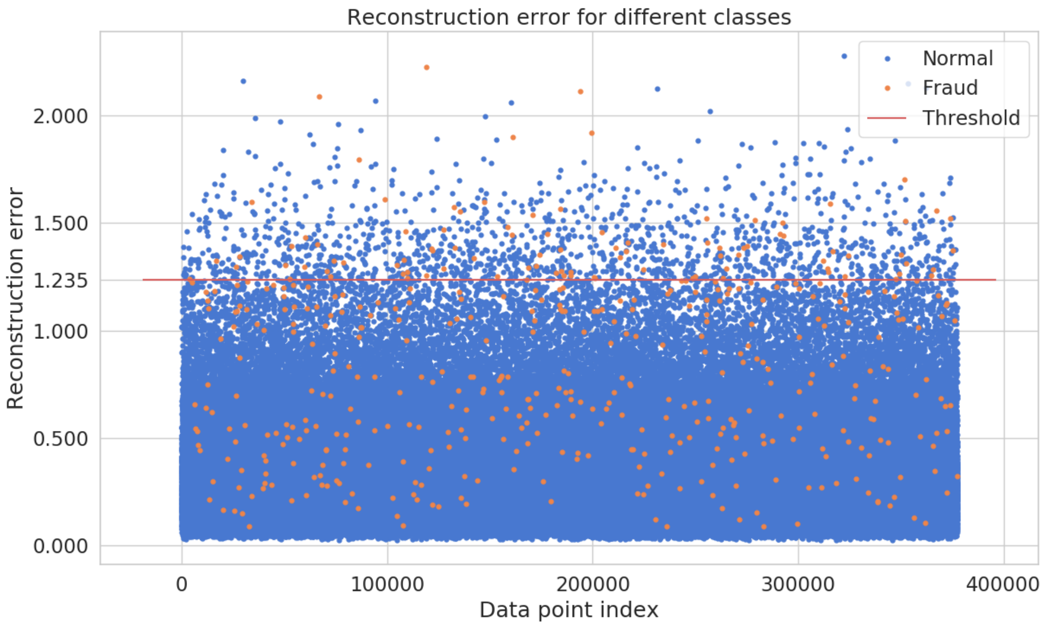


Figure 32: Threshold Determination large Dataset - Ensemble Modelling

C Plots for Large Dataset and Relatively More Frauds and Errors

C.1 Distributions Fraud vs. non Fraud

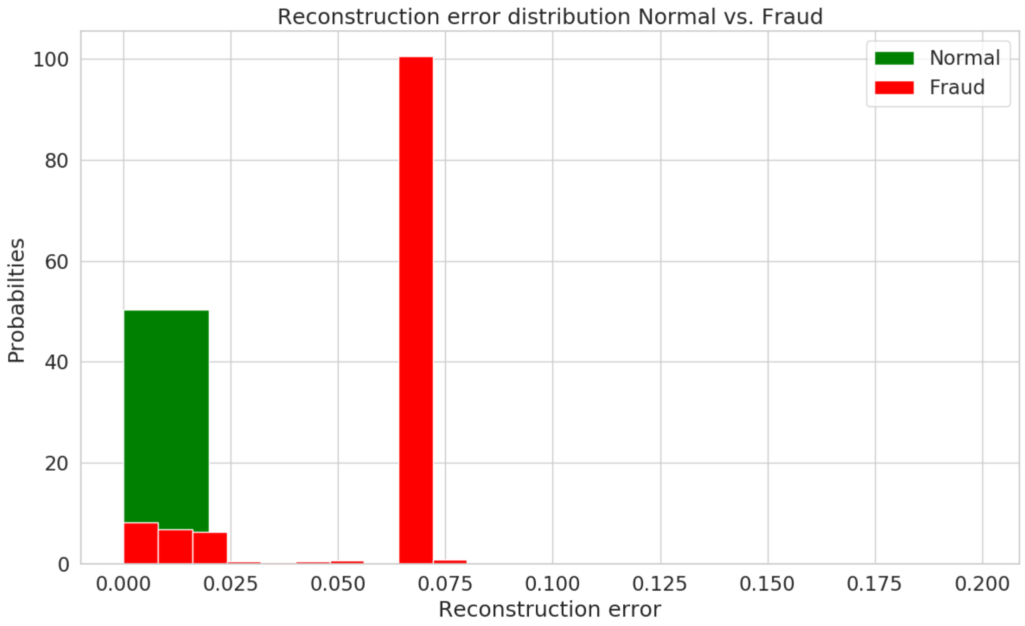


Figure 33: Distribution large Dataset - Vanilla Autoencoder

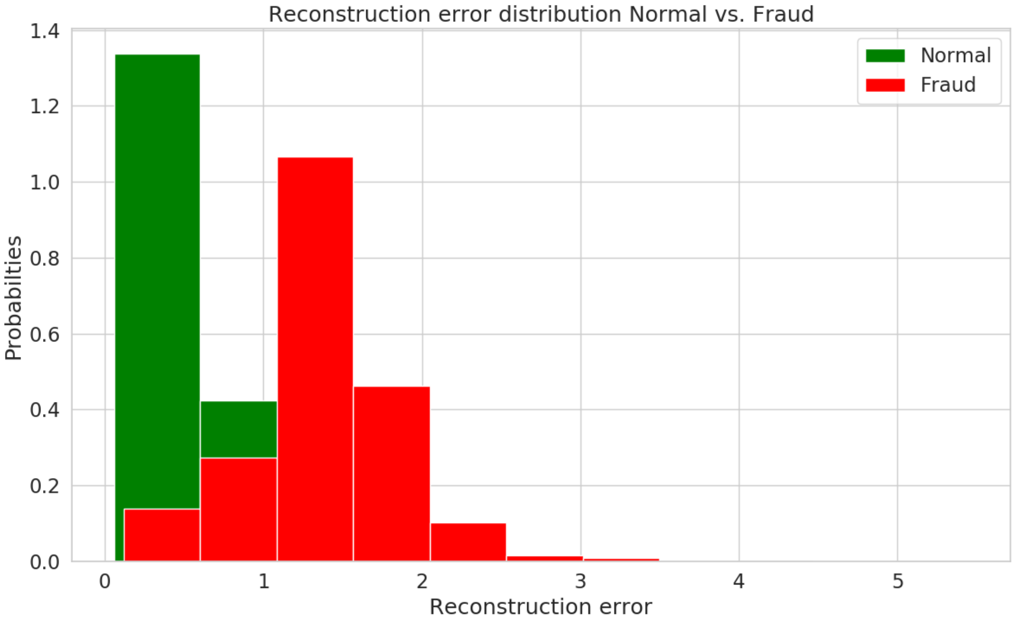


Figure 34: Distribution large Dataset - Variational Autoencoder

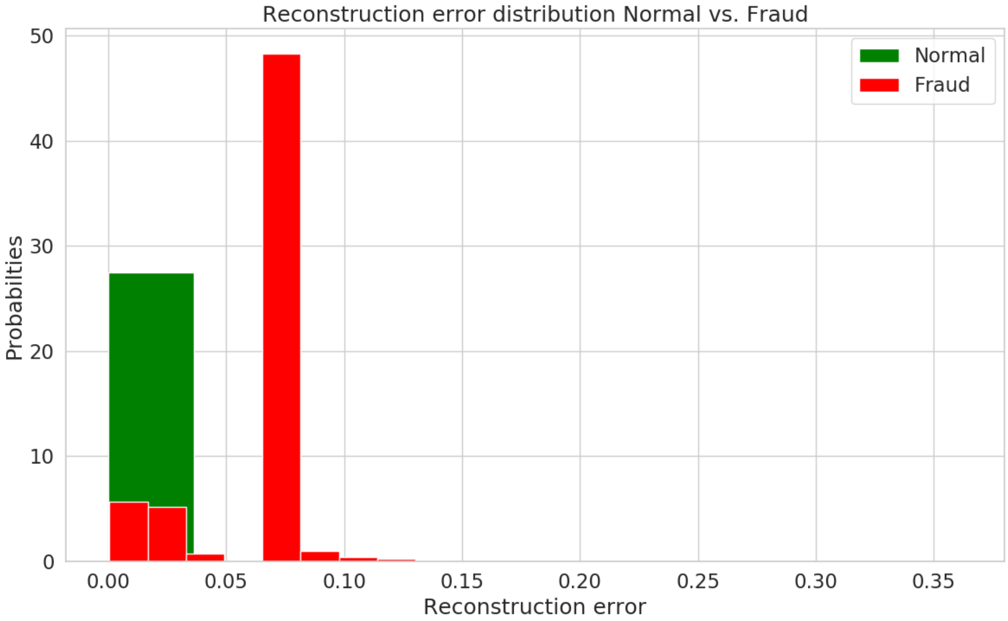


Figure 35: Distribution large Dataset - Contractive Autoencoder

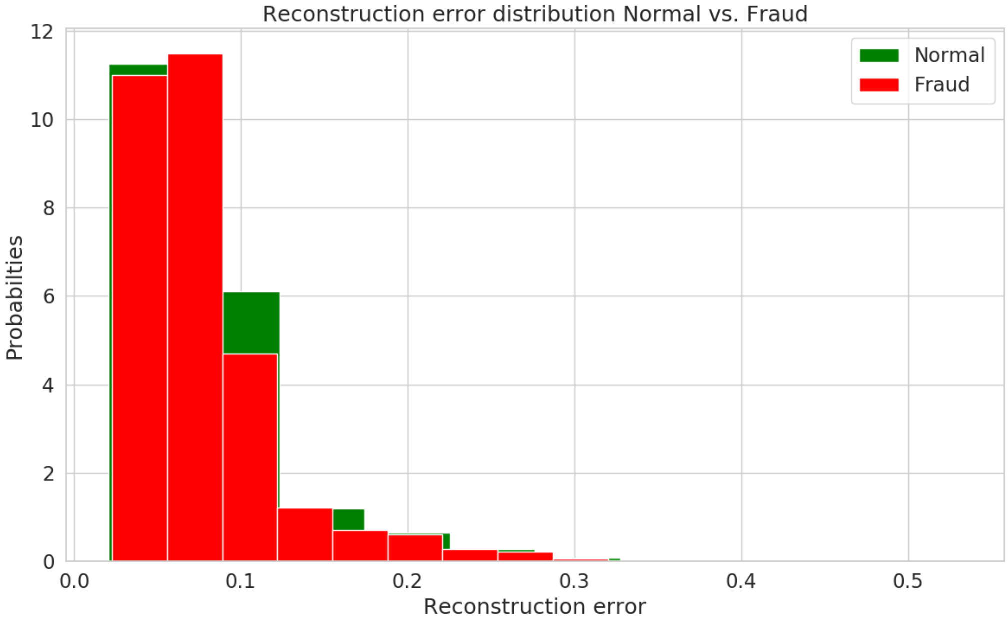


Figure 36: Distribution large Dataset - Restricted Boltzmann Machines

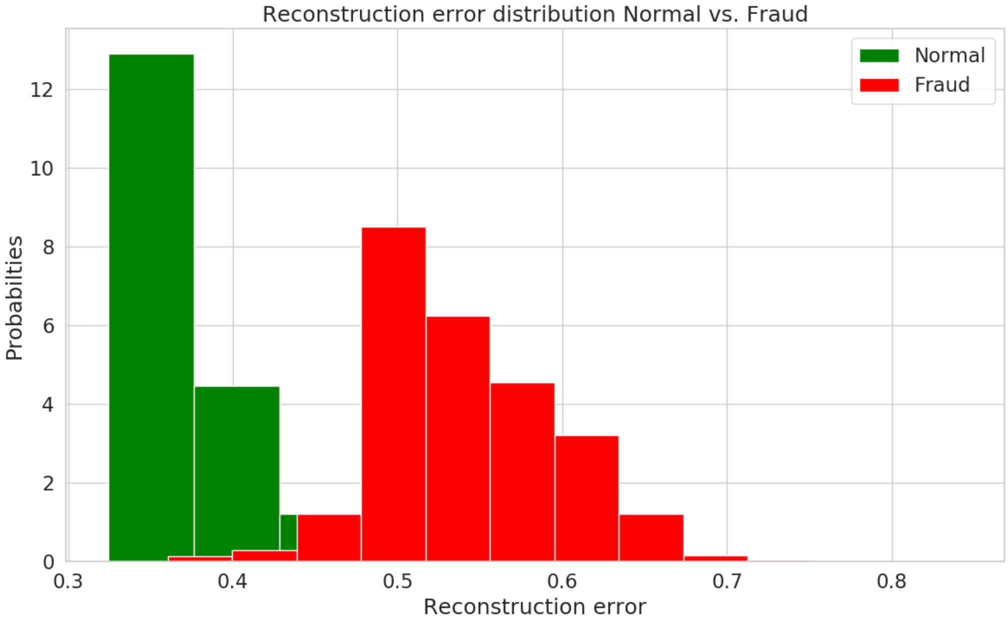


Figure 37: Distribution large Dataset - Isolation Forest

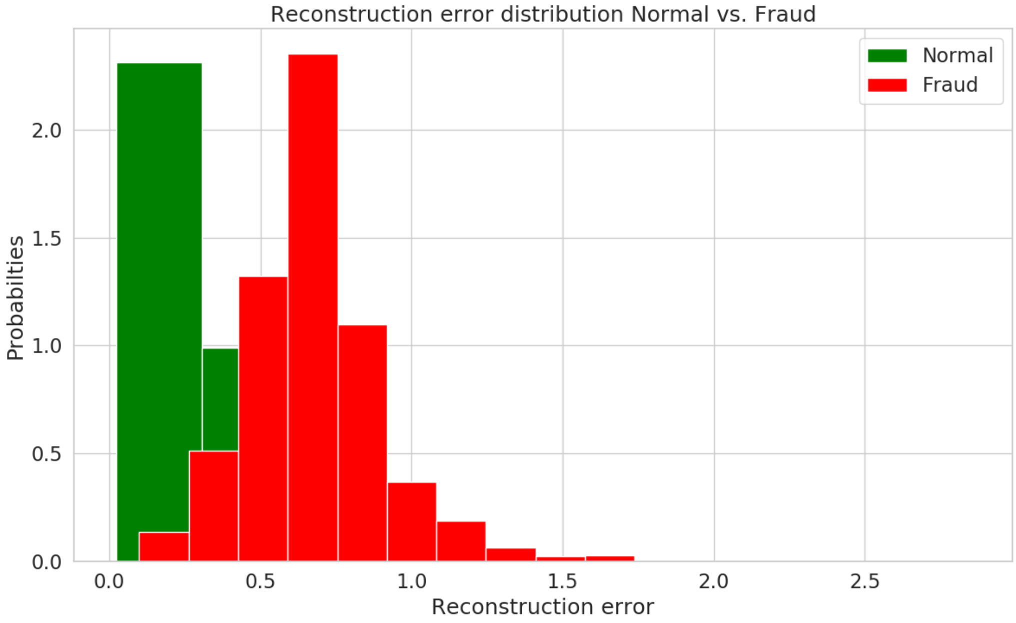


Figure 38: Distribution large Dataset - Ensemble

C.2 Visual Threshold

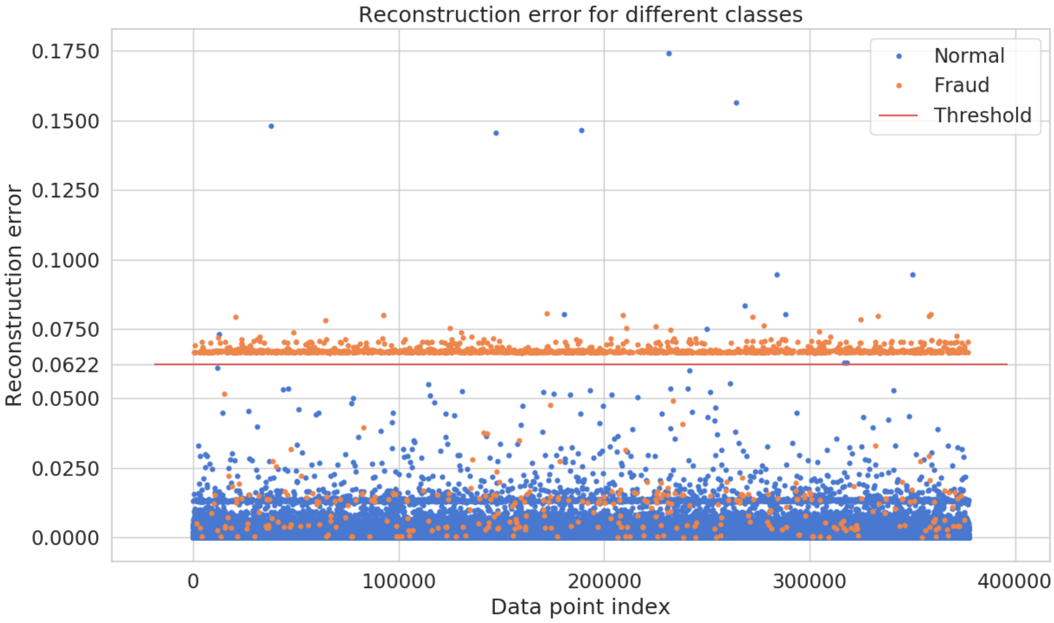


Figure 39: Threshold Determination large Dataset - Vanilla Autoencoder

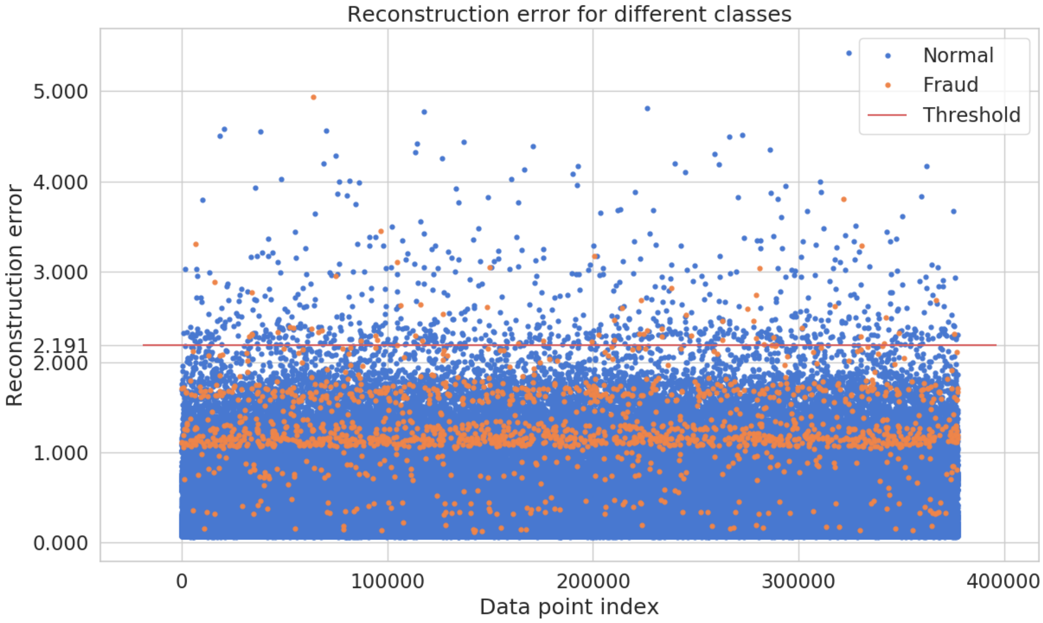


Figure 40: Threshold Determination large Dataset - Variational Autoencoder

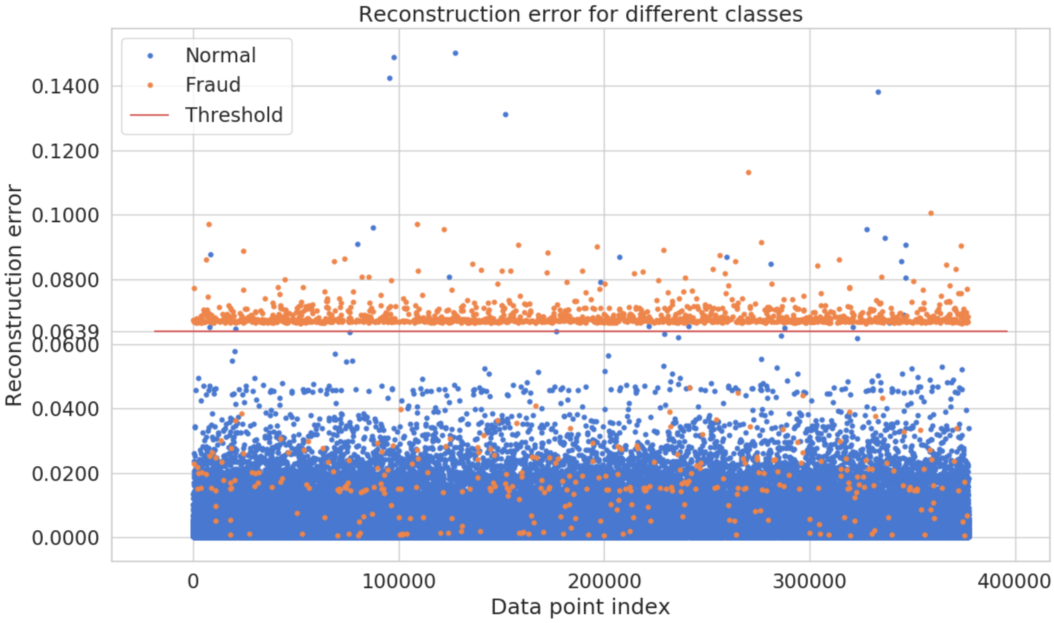


Figure 41: Threshold Determination large Dataset - Contractive Autoencoder

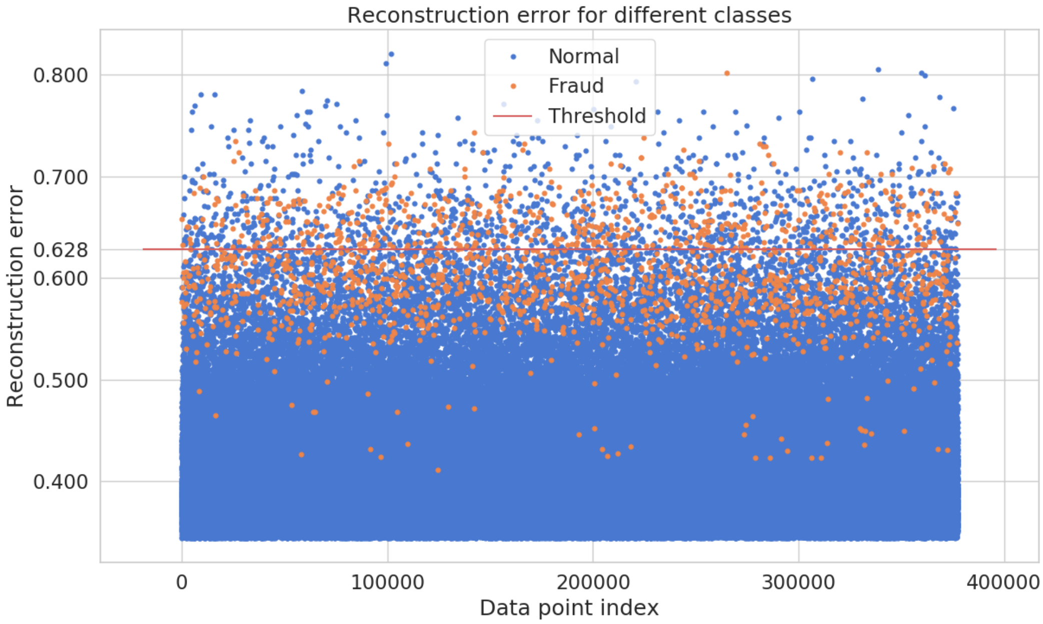


Figure 42: Threshold Determination large Dataset - Isolation Forest

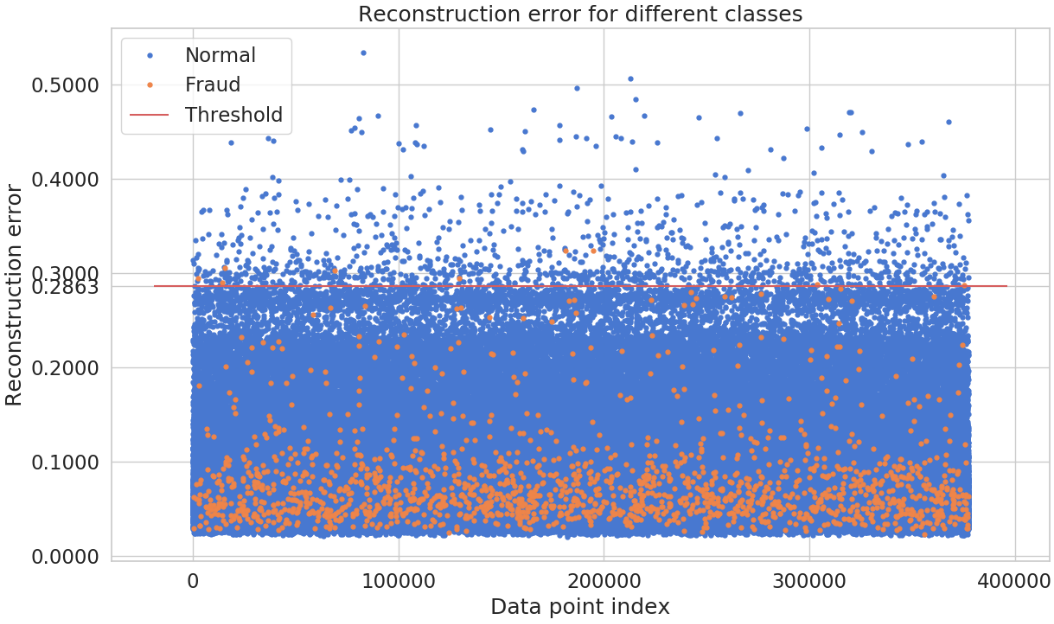


Figure 43: Threshold Determination large Dataset - Restricted Boltzmann Machines

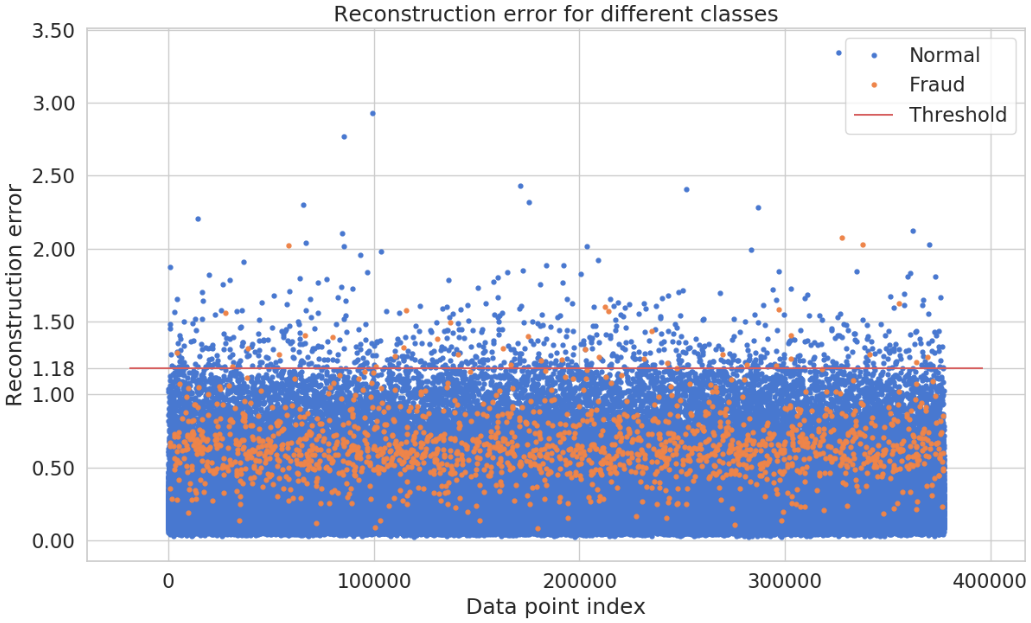


Figure 44: Threshold Determination large Dataset - Ensemble Modelling