Cyprus University of Technology

Faculty of Engineering and Technology

**Doctoral Dissertation**

# Robust Financial Crime Detection in Big Data via Uncertainty-Aware Deep Learning Techniques

**Christos Kleanthous**

**Limassol, February 2021**

CYPRUS UNIVERSITY OF TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING, COMPUTER ENGINEERING AND INFORMATICS

Doctoral Dissertation

# Robust Financial Crime Detection in Big Data via Uncertainty-Aware Deep Learning Techniques

Christos Kleanthous

Limassol, February 2021

**Approval Form**

Doctoral Dissertation

**Robust Financial Crime Detection in Big Data via Uncertainty-Aware Deep Learning Techniques**

Presented by

Christos Kleanthous

Supervisor: Dr. Sotirios Chatzis, Assistant Professor, Cyprus University of Technology

Signature _____

Member of the committee: Dr. Dimitrios Kosmopoulos, Associate Professor, University of Patras

Signature _____

Member of the committee: Dr. Stelios Z. Xanthopoulos, Associate Professor, University of the Aegean

Signature _____

Cyprus University of Technology

Limassol, February 2021

**Copyrights**



The approval of the dissertation by the Department of Electrical Engineering, Computer Engineering and Informatics does not imply necessarily the approval by the Department of the views of the writer.

I would like to thank ....

My supervisor Dr. Sotirios Chatzis for his invaluable support, the Commissioner of the Cyprus Tax Department, Mr. Yiannis Tsangaris, for his investment in the pursued state-of-the-art technology, my wife and children who have been a pillar of strength.

**ABSTRACT**

Taxation is one of the most important sources of revenue for the European Union and Value Added Tax (VAT) accounts [1] to EUR 1,2T and as such it is prevalent target for tax evasion. The European commission has estimated the difference between the estimated and collected VAT (VAT GAP) to be EUR 147B or 12.3% of the VAT revenue [2].

It is unfortunate that many EU Tax departments rely on outdated technology like rules-based systems to target high-yield taxpayers for audit in their effort to decrease the VAT GAP. In addition, the absence of research in state of the art technology by the Tax Departments is surprising, meaning that they have not benefited from advancements in intelligent systems.

This thesis draws inspiration from the most recent machine learning advances in areas like visual recognition and speech perception. We seek to introduce cutting edge technology in the tax departments arsenal against tax evasion. Specifically, we target the selection of high-yield taxpayers for audit. In our work, we rely on intelligently processed raw data obtained from available tax returns. The high-dimensional nature of the available data calls for the development of machine learning techniques that can learn to extract meaningful lower-dimensional representations to drive the predictive inference process. We address these needs in a comprehensive manner, yielding a novel a novel set of supervised and semi-supervised techniques. In all cases, we take special care mitigating the epistemic

uncertainty our problem is fraught with, as a result of the limited number of audited (labelled) data.

The success of this thesis would not have been possible without the wholeheartedly assistance of the Cyprus Tax Department and the inspired mentoring of the Taxation Commissioner Mr Yiannis Tsangaris. Specifically, with their approval, we were given anonymized access to over a million submitted VAT returns and the tax audit results, pertaining to the period 2013-2019. This availability of a large corpus of real-world data was a crucial factor that allowed for us to successfully pursue our research goals.

**Keywords:** Value Added Tax, audit selection, representation learning, epistemic uncertainty.

**TABLE OF CONTENTS**

**LIST OF FIGURES**

**LIST OF ABBREVIATIONS**

| | |
|---|---|
| AI: | Artificial Intelligence |
| AVI: | Amortized Variational Inference |
| BBVI: | Black Box Variational Inference |
| BP: | BackPropagation |
| cdf: | cumulative distribution function |
| CNN: | Convolutional Neural Networks |
| DGMs: | Deep Generative Models |
| DP: | Dirichlet process |
| DPM: | Dirichlet process mixture |
| DT: | Decision tree |
| DNNs: | Deep Neural Networks |
| EBF: | Empirical Bayesian Forests |
| ELBO: | Evidence Lower Bound |
| EM: | Expectation Maximization |
| GD: | Gradient Descent |
| i.i.d.: | independent and identically distributed |
| KL: | Kullback Leibler (divergence) |
| MC: | Monte Carlo |
| ML: | Maximum Likelihood |
| pdf: | probability density function |

PY: Pitman-Yor process

RBF: Radial Basis Function

ReLU: Rectified Linear Unit

RF: Random Forests

SGD: Stochastic Gradient Descent

SVM: Support Vector Machines

VAEs: Variational AutoEncoders

VAT: Value Added Tax

**LIST OF PUBLICATIONS**

Christos Kleanthous and Sotirios P. Chatzis. ”Gated Mixture Variational Autoencoders for Value Added Tax Audit Case Selection”. In Knowledge-Based Systems, (2019), KNOSYS 105048.

Christos Kleanthous, Theodoros Christophides and Sotirios P. Chatzis. ”Power-law Mixtures of Bayesian Forests for Value Added Tax Audit Case Selection”. In the 2020 ACM International Conference on AI in Finance.

# Chapter 1

# Introduction

Tax departments have the responsibility of increasing the tax compliance and maximizing the tax revenue. Since experienced auditors are a scarce resource, tax audits must be carefully selected so as to maximize the return from each audit. Therefore, one of the most difficult tasks tax departments face is the identification of the taxpayers with the highest tax yield for audit with high accuracy and minimal resources. In the past, tax departments have used many different labor intensive methods to this end, like one-by-one review of the submitted tax returns and rule based systems. In both cases, a taxpayer is selected for audit based on human expert knowledge.

Tax departments have not benefited from recent advancements in artificial intelligence because of the absence of publicly available tax data for research. Only recently few large tax departments of the European Union have launched an in house AI application development for audit case selection. Smaller tax departments with very limited resources cannot perform research, and an alliance with a university researchers is the only option to obtain such technology.

Whether a taxpayer will actually yield high tax after a tax audit depends on a plethora of factors that characterize the taxpayer behavior. Each characteristic has different weights and complexity, making it almost impossible for tax experts to find the rules and axioms that analytically predict the taxpayer behavior. To analyze the submitted VAT return information efficiently and build models fit for purpose, we pursue strong innovation in machine learning techniques [3, 4, 5].

## 1.1 Machine learning

Machine learning techniques use probabilities for a) modelling the relation between the submitted VAT return information (input) and the taxpayers with high tax yield after audit (output), in a discriminative model or b) find which probability distribution can best generate the taxpayer behavior, generative model. The correlation of generative and discriminative machine learning models is analyzed in [6, 7, 8, 9, 10] papers.

Machine learning pursues the probability distribution which represents most accurately the taxpayer information in generative and discriminative models. An efficient machine learning model trained to detect taxpayers with high tax yield must rely on the available tax data. To this end the machine learning model must exploit data by being flexible with adaptive parameters. This flexibility is the representation of a group of probability distributions.

A parametric machine learning model has a predefined number of adaptive pa-

rameters which are set heuristically. On the other hand non parametric models parameters depend on the input data and thus have significantly higher computational needs. Hereafter we will discuss the use of parametric models, and they will be referred simply as 'models'.

Maximum Likelihood parameter estimation is one of the simplest ways to train a machine learning model [11] but it has two major drawbacks. The first is the failure of the model to predict accurately on previously unseen data (overfitting) [12, 13]. The use of regularizers can minimize overfitting by optimizing the size of parameters. Secondly when maximum likelihood is used to train Deep Neural Networks (DNNs) some parameters are under-trained [14] and therefore the model capability to identify latent regularities of the data is undermined.

DNNs have the advantage of integrating many neural networks with non linear dependence, and have been successful in many areas of artificial Intelligence (AI). The learning capacity of DNNs made possible visual object recognition [15, 16, 17, 18, 19], speech perception [20, 21, 22, 23], language comprehension [24, 25] and other [26, 27, 28]. These achievements demonstrate the ability of the DNNs to exploit multi dimensional data and use these representations to make accurate predictions on previously unseen data.

The shallow models of neural network were popular before deep neural networks with a single latent layer (perceptrons) [29], kernel regression [30], support vector machines [31, 32, 33] for example and were popular in the machine learning community because the of the easy training. They were commonly

used with convex loss functions, reducing the training task to a convex optimization one. The learning capacity of these systems was limited to simple structures since these are not capable of extracting complex structures from rich input [34]. Their need for large amount of labeled data during training, limits their application since in the real world labeled data is scarce. The findings on how the biological visual cortex is performing the object recognition [35, 36, 37], and the limitations of single layer techniques influenced the building of deep architectures that consist of several hidden layers of nonlinear processing. These deep models contain many layers of hidden variables and a plethora of parameters which must be learned efficiently. Visual object recognition, information retrieval, classification, regression are possible because of deep architectures and GPU processing power [38, 39, 40, 41].

## 1.2 Contribution

In this thesis, we present two innovative approaches to address the considered problem. In both cases, our goal is to derive accurate predictions and inferences about the taxpayers using actual submitted VAT returns. The tax information of each taxpayer is mapped in a cause and effect fashion, in an attempt to make accurate predictions on previously unseen VAT return. This can be seen as a model where the tax return information (input) is represented in a low level multi dimensional matrix, and taxpayers with high tax yield after audit (output) is a higher abstract representation.

First, we build upon the large volume of existing works that utilize Random

Forests (RFs). RFs constitute a standard method in the Value Added Tax (VAT) audit case selection literature. Despite, though, their success, their predictive performance is still below the expectations of tax authorities, that need to timely detect cases of significant audit yield potential. This lackluster performance is mainly attributed to the fact that RFs cannot deal with data that entail non-stationary nature, multiple modalities, or discontinuities. These are common characteristics of real-world datasets; thus, the incapacity to properly address them is a major suspect for undermining their performance. We address these issues by considering a generative model with power-law behavior, capable of generating multiple distinct RFs over the observations space of the modeled data. This way, our approach enables capturing an indefinite number of distinct classification patterns, while being able to effectively handle outliers. The latter advantage is of paramount importance for the effectiveness of the modeling procedure in cases where few large parts of the observations space can be modeled by few RF classifiers, yet there is a large number of small parts of the observations space that require distinct RFs to be properly modeled (power-law nature).

Second, we take a bolder step in order to address the shortcomings stemming from the limited availability of labeled data. This is a challenging problem that has remained rather elusive for EU-based Tax Departments, due to the inadequate quantity of tax audits that can be used for conventional supervised model training. To this end, we resort to a semi-supervised learning approach. Specifically, we devise a novel Gated Mixture Variational Autoencoder deep network,

that can be effectively trained with data from a limited number of audited taxpayers, combined with a large corpus of filed VAT returns.

We developed and evaluated the out-of-sample accuracy of our methods in collaboration with the Cyprus Tax Department, and experimentally deployed it to facilitate its audit selection process. To this end, we used actual VAT data from Cyprus-based taxpayers. Empirical system performance assessment was performed in an out-of-sample fashion, in the context of potential yield-based taxpayer selection. This way, we obtained strong empirical evidence that our approach can greatly facilitate the VAT audit case selection process.

The remainder of the thesis is organized as follows: In Chapter 2, we provide more information on the business elements aspects, namely VAT, tax compliance, rules-based systems, data mining systems and our collaboration with the Cyprus Tax Department.

In Chapter 3, we provide a brief introduction to modern Bayesian inference, especially in the context of deep architectures.

In Chapter 4, we present the first contribution of this thesis, which builds upon the currently popular paradigm of Bayesian Forests, while addressing several of their pressing shortcomings.

In Chapter 5, we elaborate on the second contribution of this work, which constitutes a semi-supervised Deep Generative Model capable of making inference by using a limited number of audited taxpayers and a large corpus of filed VAT returns.

Finally, Chapter 6 of this thesis provides the conclusions of our work, elaborates the potential of our outcomes, and discusses directives for future research endeavors on related topics in the taxation area which remain open.

# Chapter 2

# Business elements

## 2.1 Value Added Tax

VAT is a consumption tax charged on the value of almost all the goods and services sold or consumed within the European Union (EU). It constitutes an indirect tax collected by enterprises on behalf of the state, and is ultimately paid by the final consumer. As such, it represents an important source of revenue for all EU Member States; according to the European Commission Taxation Trends Report, 2018 edition [1], indirect taxes comprise more than 30% of the total tax revenue in the EU.

## 2.2 Tax Compliance

To achieve maximum taxpayer compliance and thus maximize tax revenue with the limited resources available the tax departments need to be allocated carefully in order to achieve the highest possible taxpayer compliance with the tax laws and the best tax collections. One critical aspect of this issue is the priori-

tisation of the compliance action to be taken against different taxpayers. The taxpayer behaviour is very difficult to analyze due to diversity of compliance behaviour, absence of taxpayer motives for tax legislation compliance and the tax legislation complexity.

In general taxpayer compliance figure 2.1 can be classified in four categories: i) Decided not to comply ii) Do not want to comply iii) Try to comply and iv) willing to comply. Therefore a systematic approach is usually adopted to identify the major risks in respect not only to the number of taxpayers not being tax compliant but also to the amount of tax and how it will be addressed.

The Legislation provides the tax departments with a plethora of different measures that can be employed to minimize non compliance from the taxpayers with different consumption of resources. Easy to navigate web pages, taxpayer education, publications, written communication require negligible resources from the tax authorities and can address thousands of taxpayers at the same time. The result is maximum voluntary compliance from taxpayers who are willing to comply or try to comply with the tax legislation with minimum effort. For example taxpayers who have not filed the latest tax return one week before the deadline can be send a reminder letter.

On the other hand in the case of taxpayers who decided not to comply or do not want to comply the opposite is true. Control audit visits are labor intensive and require limited and expensive resources. For an audit of a single taxpayer with a non compliant past a team of expert tax auditors is required.

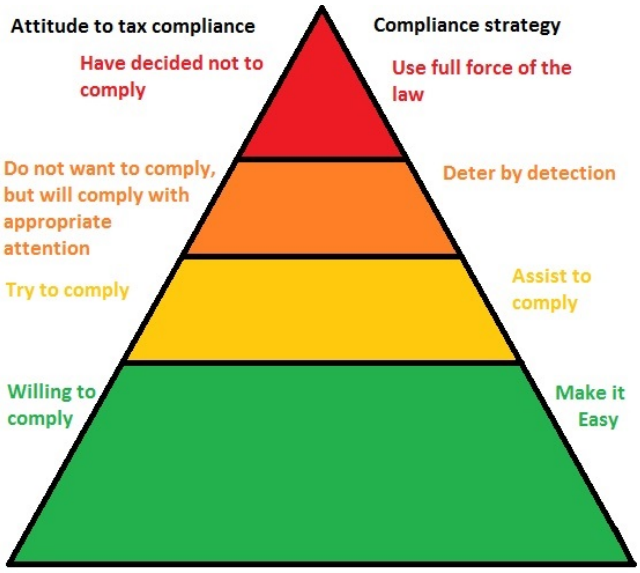


Figure 2.1: Attitude to tax compliance and compliance action

Therefore, it is key that we come up with effective audit case selection processes, with the aim of selecting the audit cases that are expected to yield significant tax. This procedure has to be performed by leveraging some sort of automation. To this end, a variety of automation methods are currently employed. Traditionally, expert auditors resort to rules-based risk modelling systems for selecting audit cases [42]. However, this method of risk modelling is highly biased as it reflects the experts' understanding of taxpayer behavior, which may be partial and incomplete. In addition, as it is needed to write hundreds of rules, this is an expensive and time consuming procedure. Finally, it suffers from the limited capability of human experts to express their acquired experience in the form of rules. Besides, under this paradigm, selection is performed on the basis of the number of satisfied rules exceeding some heuristically-set benchmark. Apparently, this threshold value has to change each time we need to accommodate more or updated rules; this renders system maintenance almost prohibitive.

Due to these disadvantages, many EU tax authorities are currently considering advanced data analytics and machine learning as a promising alternative towards automated audit case selection [43]. Indeed, tax authorities collect and process millions of tax returns that contain a plethora of diverse information. Thus, it is reasonable to assume that tax return corpora naturally lend themselves to the formation of appropriate training datasets for successful machine learning-based audit case selection systems. A prevalent characteristic of the existing developments in the field is the lack of tailor-made machine learning models that can make the most out of the vastly available tax return data. For instance, Random Forests (RFs) constitute one of the most commonly used machine learning approaches in the context of tax audit case selection [44]. The main reasons behind their prominence can be traced to: (i) their capacity to effectively deal with high-dimensional data; (ii) their computational efficiency, both when it comes to training and when it comes to prediction generation; (iii) their notable robustness to outliers and non-linear features in the training data; (iv) their capability to effectively learn from unbalanced classification data, which are typical in real-world datasets stemming from tax audits.

## 2.3 Rules-based and data mining systems

The software SAS Enterprise Miner[1] has been used in the past to select audit cases for VAT purposes. Before use, an extensive feature engineering is a must to cater for missing data and categorize data in different groups.

[1] https://www.sas.com/en_us/software/enterprise-miner.html

The Irish Revenue Office has addressed the issue of selecting high tax yield non compliant taxpayers with the employment of data mining [45] using the SAS software (SAS Enterprise Miner and SAS Enterprise Guide). The experience gained from the banks and insurance companies is utilized against the non compliant taxpayers. According to the IDC [46] SAS and IBM have a 43% (revenue volume) combined market share of tools for predictive analytics.

The task of selecting few taxpayers with the highest expected audit yield accurately is almost impossible to be performed manually. The allocation of the experienced tax auditors needs high degree of accuracy to be translated in to maximum revenue as the number of audits that can be performed is limited. This cannot be achieved manually consistently.

Since the process of audit case selection cannot be facilitated manually the automation process of identifying high risk taxpayers and audit case selection was spearheaded by heuristic rules set by expert auditors [42]. A decision to audit a taxpayer depends if the number of rules that apply exceed a threshold, for example if fifty rules ”fire” an audit is performed. This process is highly subjective since is based on the opinion of the experts and requires time and effort from a team of experts who create hundreds of rules for different areas like number of late returns, sales fluctuations, inconsistencies in the amounts declared etc.

After relying to the rules-based systems for many years the tax departments moved away from these complicated, subjective rules and thresholds and considered more reliable automated alternatives like advanced analytics and machine

learning models [43]. Tax departments collect vast amounts of data for each and every taxpayer like filed returns which are not utilized. If exploited successfully this data can open the road for an automated process for selecting accurately taxpayers with high audit yields.

# Chapter 3

# Bayesian Inference

The popularity of Bayesian Inference and its extensions increased recently in machine learning against the frequentist approach. The benefits of treating the unknown set of parameters as random variable instead of fixed values are the reason. Fixed values are associated with frequentist inference which is sourced from the classic view of probabilities, the frequency of the events.

The random variables refer to the uncertainty quantification. We associate randomness, by introducing a distribution that encapsulates the uncertainty about their true values before receiving information (prior distribution). After the information is received by the model the values are updated to a more relative belief (posterior distribution). Parameter learning can be seen as an inverse procedure where we attempt to derive the parameters from data. The Bayesian inference is also an inversion procedure formulated in a probabilistic context.

Below is the mathematical formulation of Bayes theorem in a statistical inference context. Given a set of observations y, which are controlled by the unknown set of parameters $\theta$ we can write:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{3.1}$$

Where $p(y|\theta)$ is the likelihood function of the environment. For the inversion procedure all we need is to guess the prior distribution $p(\theta)$.

Most commonly the prior distribution is selected based on our intuition about the probability of the underlying regularities with a known family of distributions. After a reasonable choice of the prior is made, the advantages from the Bayesian approach arise and will be further discussed in the next sections.

**Model Selection**

The model selection task under the Bayes view uses probabilities to portray the uncertainty of the optimal model. The task is transformed into posterior distribution calculation over a set of models given some a priori knowledge and some information. The a priori knowledge is encapsulated in the form of a prior distribution for each model $p(\mu_i)$. Applying Bayes rule and the considering the new information as y, the posterior of the model is given by:

$$p(m_i|y) = \frac{p(y|m_i)p(m_i)}{p(y)} \tag{3.2}$$

$$p(y) = \sum p(y|m_i)p(m_i) \tag{3.3}$$

The first term in the numerator in the equation 3.2 is the evidence or marginal likelihood of a specific model and is significant quantity in Bayesian inference.

It can be analyzed over the model's parameters as:

$$p(y|m_i) = \int p(y|\theta, m_i)p(\theta|m_i)d\theta \tag{3.4}$$

Where $p(\theta|m_i)$ a prior distribution of the model's parameters. Furthermore, we can compute the posterior distribution of the parameters for each model structure.

$$p(\theta|y, m) = \frac{p(y|\theta, m)p(\theta|m)}{p(y|m)} \tag{3.5}$$

A simple way to calculate the integral is to estimate the value with Maximum Likelihood or Maximum a Posteriori methods.Maximum likelihood aims to maximize the first term of the right side of the integral 3.4.

$$\theta_{ML} = arg_{\theta} max[p(y|\theta, m)] \tag{3.6}$$

Maximum a Posteriori aims to maximize both the terms of the right hand side of the integral 3.4.

$$\theta_{MAP} = arg_{\theta} max[p(y|\theta, m)p(\theta|m)] \tag{3.7}$$

The above models are per datapoint solutions attempting to solve the equation with fixed parameters. A more universal solution is to solve the evidence equation by marginalizing out the unknown parameters $\theta$. As this is not always possible we use deterministic approximation techniques such as variational models

that try to optimize a tractable lower or upper bound that is close enough to the true expectation of the random variable. The bound has analytical solutions which can be performed in a limited time and the problem now changes to bound optimization; making the bound as tight as possible.

### Evidence Function

The model selection task tries to find the probability of the model as a way to measure the significance of the model.

$$p(m_i|y) = \frac{p(y|m_i)p(m_i)}{p(y)} \tag{3.8}$$

Where $p(m_i)$ is the prior probability of model $\mu_i$ which provides a quantification of our uncertainty as compared to all alternative models.

It can be seen like our intuition on how possible a model is as compared to the alternatives, before obtaining information.

Finding the best model can be simplified considering the following: • The denominator p(y) marginalizes out the models dependence

$$p(y) = \sum [p(y|m_i)p(m_i)] \tag{3.9}$$

Assigning equal probabilities to all possible models. The first observation simplifies the task of finding the most probable model by maximizing only the numerator and the second maximizing only the evidence/likelihood function $p(y|\mu_i)$.

This is the reason why this specific probability density function is known as the evidence function for the model. Although in practice we are content with using the most probable model, we should not choose among models. In a more Bayesian orthodox way inference is performed by summing over all models.

The use of specific model derives from the practical observation that the evidence function of practical problems is more relevant to a specific model, simplifying the task at hand.

### Parameter Fitting

Under the parameter selection context the Bayesian approach transforms to:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{3.10}$$

$$p(y) = \int p(y|\theta)p(\theta)d\theta \tag{3.11}$$

The $p(\theta|y)$ is the posterior distribution we intend to maximize, $p(y|\theta)$ the likelihood function of the parameters, $p(\theta)$ our prior intuition about the parameters and $p(y)$ the marginal likelihood (normalizing constant).

The unknown parameters $\theta$ are treated as fixed but unknown (deterministic variables) and therefore the prior distribution and the normalizing constant can be omitted. The purpose is to maximize the likelihood function by selecting the best parameters (Maximum Likelihood).

$$\theta_{ML} = arg_{\theta} max[p(y|\theta)] \tag{3.12}$$

When the unknown parameters $\theta$ are treated as random variables and setting the normalizing constant independent of the parameters we obtain the Maximum Posteriori. Where considering the best parameters becomes the task of maximizing the evidence function augmented with the prior distribution.

$$\theta_{MAP} = arg_{\theta} max[p(y|\theta)p(\theta)] \tag{3.13}$$

The Bayesian approach makes no simplifying assumptions about the marginal and needs to be able to calculate the integral. Recall that in order to make a rich model of the environment we need to use families of distributions, these families are controlled by a set of parameters. So the unobserved random variables $\theta$ can be changed to latent variables z governed by deterministic parameters $\theta$.

$$p(z|y) = \frac{p(y|z;\theta_2)p(z;\theta_1)}{p(y)} \tag{3.14}$$

Since the latent variables are unobserved, finding the optimal parameters can be performed using an expectation maximization algorithm which iterates between the optimal parameters and optimal posterior distribution. In the next three subsections we will get a closer view on the three methods, Maximum Likelihood, Maximum a Posteriori and Expectation Maximization.

**Maximum Likelihood**

The maximum likelihood (ML) method is used to estimate the parameters of the input data. The ML method chooses a set of values for which maximizes the likelihood function $p(y|\theta)$ using the fixed input data and a flexible statistical model.

The unknown parameter is handled as a deterministic variable, parameterizing the probability density function describing the output vector of the information. The value of the parameters is determined only by the information sourced from the experiments sequence.

The ML model is a fixed problem data modeler because there is no need to generalize, every input has predefined output like a memory storage. Since the model is not capable of finding the underlying regularities which exist beneath the data, the ML models are overfitting the data and fail to generalize on previously unseen data.

**Maximum a Posteriori**

A method related to the ML but with more advanced features for estimating the parameters of the input data is the MAximum Posteriori (MAP). A prior distribution is added over the parameters as a boosted optimization. The fixed input data are used in a flexible statistical model which selects the set of the values which maximize the likelihood function $p(y|\theta)$ boosted with a prior distribution $p(\theta)$

The unknown parameter is considered to be a random vector in MAP, which is tuning the probability density function describing the output vector if the information.

The parameters value are now controlled by the obtained information and the choice of prior. One way of finding the parameters is by assuming conjugate formulation of both prior and conditional probability density functions e.g. Gaussian, which leads to the posterior to be of the same distribution and exploit their predefined maximum.

$$\theta_{MAP} = \mathrm{E}[\theta|\mathrm{y}] \tag{3.15}$$

The MAP estimate can be considered as a regularized version of the ML with the use of augmented extra parameters. While these mitigate the problem of overfitting highlighted above require to be optimized by a different dataset, thus increasing the size of the input data required to run the model.

#### Expectation Maximization

An iterative approach towards the ML or MAP estimates of parameters in statistical models is the expectation-maximization (EM) algorithm, the latent variables dictate the model.

Given a set of observations $x$, a set of unobserved latent variables $z$ and their joint distribution parameterized in terms of a vector of unknown parameters $\theta$, we can calculate the complete data log-likelihood $p(x,z;\theta)$ by an iterative

method. The EM iteration method alternates between an expectation (E) step and a maximization (M) step. It begins by initializing randomly the parameters $\theta$, then calculates the expectation of the complete data log-likelihood given fixed $\theta$ and then calculates the new $\theta$ by maximizing the expectation of the previous step. This is done till a criterion is achieved such as time limit, iteration limit or the convergence of $\theta$.

We refer to the set $x, z$ as the complete data set and to the set of observations $x$ as the incomplete one. The EM success lies in the assumption that although the latent variables are unobserved the posterior distribution $p(z|x;\theta)$ is fully specified, given the values $\theta$ and $x$. In the case where the assumption does not hold we have to resort to variants of the EM, which will attempt to approximate it.

In case the complete log-likelihood $p(x,z;\theta)$ was made available, ML would solve the problem.

The EM algorithm can be outlined as: 1. Randomly initialize $\theta 0$ 2. Expectation E-step: At the $l+1$, compute posterior distribution $p(z|x;\theta l)$ and then the expectation of the complete data $log$ likelihood $p(x,z;\theta l)$. 3. Maximization M-step: Determine $\theta l+1$ so that maximizes the expectation. 4. Check for convergence according to a criterion. If it is not satisfied go to 2.

The use of EM algorithm presupposes that working with the joint probability density function $p(x,z;\theta)$ is computationally tractable. This is, for example, the case when working within the exponential family probability density functions,

where the E-step may require only the computation of a few statistics of latent variables.

The algorithm's convergence is slower than the quadratic convergence of Newtontype searching techniques, although near an optimal point a speed up may be possible. However, the convergence of the algorithm is smooth and its complexity more attractive to Newton-type schemes, with no matrix inversions involved.

The EM algorithm can be expanded to obtain the MAP estimation. To this end, the M-step is changed to maximize the expectation of the complete data log likelihood $p(x,z;\theta l)$ plus log $p(\theta)$ where $p(\theta)$ is the prior probability density function associated with. One disadvantage of using the EM algorithm is that it is sensitive to the initialization of $\theta 0$. In practice, we run the algorithm from different initial points and keep the best the results. Initialization techniques have been developed to alleviate this issue.

**Lower Bound Maximization**

In this section we will derive to the EM algorithm from another perspective. Let us consider the functional

$$F(q,\theta) = \int q(z) \log \frac{p(x,z;\theta)}{q(z)} dz \tag{3.16}$$

$q(z)$ is any probability density function defined over the latent variables. The functional

depends on $\theta$ and on $q(\Delta)$. Because $p(x;\theta)$ does not depend on $q(z)$ we

have:

$$F(q,\theta) = \int q(z) \log \frac{p(z|x;\theta)}{q(z)} dz + \log p(x;\theta) \tag{3.17}$$

The first term on the right-hand side is the negative of the Kullback-Leibler divergence between $q(z)$ and $p(z|x;\theta)$, which we will denote as $KL(q \parallel p)$. Thus finally we get:

$$\log p(x;\theta) = F(q,\theta) + KL(q \parallel p) \tag{3.18}$$

Because KL divergence is a non-negative quantity we have:

$$\log p(x;\theta) \geqslant F(q,\theta) \tag{3.19}$$

$F(q,\theta)$ is consider a lower bound of the log-likelihood, which is equal if and only if $q(z) = p(z|x;\theta)$. Under the previous formulation we can maximize the log-likelihood by trying to maximize its lower bound. Keep in mind that the functional depends on two terms q(·) and $\theta$ and they can be optimized with an iterative procedure under the following steps.

1. Randomly initialize $\theta$0 2. Holding $\theta$ fixed, optimize with respect to $q(\Delta)$. 3. Holding $q(\Delta)$ fixed, optimize with respect to $\theta$. 4. Check for convergence according to a criterion. If it is not satisfied go to 2.

Step 2 is achieved if we set $q(z)$ equal to the posterior $p(z|x;\theta)$. Comparing it to the EM algorithm its clear that the log-likelihood $p(x;\theta)$ is guaranteed not to

decrease after each iteration.

Methods that are based on optimizing a lower or upper bound instead of the original cost are also known as minorize-maximization or majorize-minimization methods.

## 3.1 Approximate Bayesian Inference

To achieve an efficient training of the models utilizing the unobserved latent random variables the evaluation of the posterior distribution $p(z|x)$ is required and the evaluation of expectations w.r.t the distribution. The posterior distribution must also be complex enough to be able to capture the true distribution.

The unobserved hidden variables can be discrete, continuous or a hybrid of the two. To train the model we must calculate the integral/sum marginalizing out the unobserved variable. In the case of continuous variables, the required integrations may not have closed-form analytical solutions. For discrete variables, the marginalizations involve summing over all possible configurations of the hidden variables, and although this is always possible in principle, the exponential number of latent states prohibit the exact calculation. These are considered intractabilities and are very common and appear even in cases of moderately complicated models.

To have efficient inference we need to resort to approximation schemes. Stochastic approximation techniques such as Markov chain Monte Carlo have helped the widespread use of Bayesian methods across many domains. Stochastic tech-

niques have the property that can generate exact results if given infinite computational time and resource. Their approximation arises from the use of finite amount of time. However even with this practical limitation sampling methods are still computationally demanding, limiting their use to small scale problems. Main reasons that prohibit numerical integration is the dimensionality of the space and the complexity of the integrand. Furthermore, it is difficult to know whether a sampling methods generates independent samples from the intractable distribution.

A viable option is deterministic approximation techniques.

### Variational Bayes

One of the most popular deterministic approximate inference techniques is Variational Bayes. Variational Bayes is the core of variational inference. The idea is to approximate the intractable posterior with a simpler family of distributions that are analytically tractable. And then seek the distribution that minimizes a similarity measure. This consists in constructing a lower (or upper) bound and maximize it instead of the intractable log marginal likelihood. This simplifies the posterior estimation task to an optimization problem. Variational methods are consider approximations by constricting the range of functions over which the optimization is performed. The aim is to transform the intractable issues to analytically tractable ones by obtaining a variational approximation of the intractable model posterior. This can be achieved by exploiting known expectations of distributions. One way to achieve this is with a conjugate formulation of

the distributions. This ensures the tractability of the analytical expressions but severely limits the expressiveness of the model. Before discussing extensions of Variational Bayes lets observe it under a mathematical point of view. Consider the marginal likelihood $p(X|\theta)$ where $X = [x1, x2, .., xn]$ are the individual datapoints and $\theta$ the model's parameters (model's identity). Given independent and identically distributed $(i.i.d.)$ datapoints it can be rewritten as:

$$p(X|\theta) = p(x_1, x_2, .., x_n|\theta) = \prod p(x_i|\theta) \tag{3.20}$$

For simplicity and without loss of generality we use the natural logarithm. The natural logarithm is a monotonic transformation i.e the estimated parameters derived by maximizing the likelihood are identical for both formulations. It's also concave function.

$$\log p(X|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta) \tag{3.21}$$

Introducing latent variables into the mix, maximizing the marginal likelihood of an individual datapoint can be expanded as:

$$p(x|\theta) = \int p(x|z, \theta) p(z|\theta) dz \tag{3.22}$$

or

$$p(x|\theta) = \sum_{j=1}^{M} p(x|z_j, \theta) p(z_j|\theta) \tag{3.23}$$

depending on the identity of the variable (discrete and/or continuous). For simplicity from here on we will use the integral. Keep in mind that the summation and integral at this point are interchangeable.

Under a Bayesian view, $x \sim p(x|z,\theta)$is a random variable described by a probability distribution parameterized by $z$ and $\theta$, $z \sim p(z|\theta)$ an unobserved random variable described by a probability distribution parameterized by $\theta$.

To train this model we must calculate the integral marginalizing out the unobserved variable z but this under the above formulation is intractable. To use Variational Bayes we introduce a new probability distribution $q(z|\varphi)$ which we will try to approximate to the true sought posterior $p(z|x,\theta)$.

$$\log p(x|\theta) = \log \int \frac{q(z|\varphi)}{q(z|\varphi)} p(x,z|\theta) \tag{3.24}$$

With the use of Jensen's inequality, specifically $logE[x] \geqslant E[\log x]$ we have:

$$\log p(x|\theta) \geqslant \int q(z|\varphi) \log \frac{p(x,z|\theta)}{q(z|\varphi)} dz = \mathscr{L}(q) \tag{3.25}$$

where L(q) the lower bound that we seek to maximize under the Variational Bayes setup. This bound is often referred to as the free energy of the model.

Then the marginal likelihood can be rewritten as:

$$\log p(x|\theta) = \mathscr{L}(q) + D_{KL}(q(z|\varphi) \parallel p(z|x,\theta)) \tag{3.26}$$

where $D_{KL}$the Kullback-Leibler divergence between the two distributions. The

divergence is always positive and equals to zero when the two distributions match. This leads to the conclusion that minimizing the divergence maximizes the lower bound. The problem is now transformed into finding a tighter bound to the real distribution that is scalable to larger applications.

Following is the proof that equation 3.26 holds.

$$log p(x|\theta) = \mathscr{L}(q) + D_{KL}(q(z|\varphi) \parallel p(z|x,\varphi))$$

$$= \int q(z|\varphi) \log \frac{p(x,z|\theta)}{q(z|\varphi)} dz + \int q(z|\varphi) \log \frac{q(z|\varphi)}{p(z|x,\theta)} dz$$

$$= \int q(z|\varphi) \log \frac{p(z|x,\theta)p(x|\theta)}{q(z|\varphi)} dz + \int q(z|\varphi) \log \frac{q(z|\varphi)}{p(z|x,\theta)} dz$$

When $q(z|\varphi) = p(z|x,\theta)$ we have

$$\log p(x|\theta) = \int q(z|\varphi) \log p(x|\theta) dz + \int q(z|\varphi) \log 1 dz$$

$$= \log p(x|\theta) \int q(z|\varphi) dz = \log p(x|\theta)$$

Before we continue to more modern techniques of approximate Bayesian inference we will discuss Mean-Field Variational Inference.

### Mean-Field Variational Inference

In the previous section we discussed Variational Bayes, however for simplicity we omitted the dimensionality of the latent variables. In this section we will discuss mean-field Variational Inference which makes the assumption of independent latent variables.

A rich predictive model to be expressive enough needs to include more than one latent variable. So the previous formulation for the parametric form of the approximate distribution is better written as $Z = [z_1, z_2, .., z_n]$. This imposes a difficulty on the training due to the dependence of the variables.

This can be simplified with the mean-field assumption where the latent variables are independent.

$$q(Z|\varphi) = \prod_{j=1}^{J} q(z_j|\varphi_j) \tag{3.27}$$

However, introducing independence between the latent variables leads to a family of approximate distributions that are less expressive and will not include the sought posterior. Mean-Field Variational Bayes and the simple Variational Bayes limit their use to mainly conjugate models.Non-conjugate models can be trained with ad hoc models of variational inference algorithms such as approximations [47, 48], alternative bounds [49, 50, 51] and numerical quadrature [52]. This includes models like Bayesian logistic regression [49], Bayesian generalized linear models, discrete choice models [47], Bayesian item response models[53] and non-conjugate topic models [50]. Also Wang and Blei [54]

developed two extensions to Mean-Filed Variational Inference that can be applied to wider range of non-conjugate models, Laplace Variational Inference and Delta method variational inference. The first uses Laplace approximations and the latter Taylor expansions. Machine Learning scientists work in order to derive inference procedures where the distributions can be easily changed without model specific analytics. In the upcoming sections we will discuss techniques that extend the Variational Bayes algorithm to train even more expressive models.

### Normalizing Flows

The core problem of variational inference is the appropriate choice of approximate posterior distribution. Simple structured approximations have a significant impact of the inference power of the model. Normalizing flows [55] aims to construct a flexible, arbitrary complex and scalable distributions without loosing efficiency.

Normalizing flows introduce approximations constructed through a normalizing flow. A simple distribution is transformed to a complex one with the assistance of invertible transformations. These transformations provide a tighter variational lower bound with linear time complexity. The first transformation is called the flow $q_0(z_0)$ and then next transformations are the normalizing flow $q_k(z_k)$ . These invertible flows can be visualized as a sequence of expansions or contraction of the initial probability. With the help of the law of unconscious statistician (LOTUS) we can compute the expectations w.r.t the transformed

probability $q_K$ without explicitly knowing $q_K$.

However, inverse transformations for training require the computation of the Jacobian. The formulation of the model requires only one Jacobian for the sequence of transformations at each hidden layer but we still have $O(LD^3)$ computational complexity, where D the dimension of the hidden layers and L is the number of hidden layers used. Also their gradient involves additional operations with $O(LD^3)$ complexity that even may be unstable. A way to limit the computational complexity is the use of linear time transformations that still derive a complex enough model [55]. Another option is the use of Volume Preserving flows that have Jacobian determinant equal to one [56].

### Inference Networks

A common practice for Variational Inference is the use of an inference network a.k.a recognition model. The inference network represents the approximate posterior distribution $q(z|\varphi)$ under an inverse map from observations x to latent variables z. Formulating the approximate posterior distribution as $q(z|x,\varphi)$ achieves global variational parameters alleviating the need to compute them per datapoint. Thus, the cost of inference is amortized by generalizing between the posterior estimates for all latent variables through the parameters of the inference network, under a simple feed-forward computation scheme with complexity $O(N)$. One of the simplest recognition model we can use is by assuming the sought posterior to be of Gaussian form with diagonal covariance and postulate the approximate posterior as :

$$q(z|x,\varphi) = N(z|\mu_\varphi(x), \sigma^2_\varphi(\mathrm{x})) \tag{3.28}$$

### Amortized Variational Inference

Combining the above techniques we get closer to a complete framework that is able to train rich probabilistic models with variational inference. A framework that will allow us to make changes on the distributions assumptions at ease without the need for add hoc statistics. An example of the progress to that direction is Amortized Variational Inference (AVI)[55] which combines inference networks and stochastic back-propagation. However, AVI is limited to continuous latent variables due to high variance on the discrete case.

# Chapter 4

# Power-law Mixtures of Bayesian Forests for Value Added Tax Audit Case Selection

## 4.1 Bayesian Forests

RF models constitute one of the most popular methods for both regression and classification. Their functionality revolves around the concept of decision trees (DTs) [57]. As DTs are formulated by means of a random partition procedure, they constitute weak learners the performance of which may become underwhelming when dealing with difficult classification or regression tasks. To compensate for this weakness, RFs resort to the ensemble learning rationale: They fit multiple DTs on the same dataset, each performing different hierarchical random splits, $\theta$, of the input space. Then, prediction is performed on the basis of an appropriate voting mechanism.

Recently, [58] introduced an alternative view towards RFs: Their empirical Bayesian forest (EBF) algorithm replaces the Poisson distribution, from which the tree parameters $\theta$ are drawn, with an Exponential (or Dirichlet, when nor-

malized) posterior, $\mathscr{T}(\theta)$. In this context, a suggested sample (random partition), $\theta$, is retained if it facilitates the minimization of the Gini impurity index on the training dataset. This inferential treatment has been shown to induce a reliable performance gain across diverse application areas.

## 4.2 Bayesian Nonparametrics

Nonparametric Bayesian modeling techniques, especially Dirichlet process mixture (DPM) models, have become very popular for performing nonparametric density estimation [59, 60, 61]. Briefly, a realization of a DPM can be seen as an infinite mixture of distributions with given parametric shape (e.g., Gaussian). This theory is based on the observation that an infinite number of component distributions in an ordinary finite mixture model tends on the limit to a Dirichlet process (DP) prior [60, 62]. Eventually, as a part of the model fitting procedure, the nonparametric Bayesian inference scheme induced by a DPM model yields a posterior distribution on the proper number of model component densities (inferred clusters) [63], rather than selecting a fixed number of mixture components. Hence, the obtained nonparametric Bayesian formulation eliminates the need of doing inference (or making arbitrary choices) on the number of mixture components (clusters) necessary to represent the modeled data.

## 4.3 The Pitman-Yor (PY) process

DP models were first introduced by Ferguson [64]. A DP is characterized by a base distribution $G_0$ and a positive scalar $\alpha$, usually referred to as the innova-

tion parameter, and is denoted as $\mathrm{DP}(\alpha, G_0)$. Essentially, a DP is a distribution placed over a distribution. Let us suppose we randomly draw a sample distribution $G$ from a DP, and, subsequently, we independently draw $M$ random variables $\{\Theta_m^*\}_{m=1}^M$ from $G$:

$$G|\alpha, G_0 \sim \mathrm{DP}(\alpha, G_0) \tag{4.1}$$

$$\Theta_m^*|G \sim G, \quad m = 1, \ldots M \tag{4.2}$$

Integrating out $G$, the joint distribution of the variables $\{\Theta_m^*\}_{m=1}^M$ can be shown to exhibit a clustering effect. Specifically, given the first $M-1$ samples of $G$, $\{\Theta_m^*\}_{m=1}^{M-1}$, it can be shown that a new sample $\Theta_M^*$ is either (a) drawn from the base distribution $G_0$ with probability $\frac{\alpha}{\alpha+M-1}$, or (b) is selected from the existing draws, according to a multinomial allocation, with probabilities proportional to the number of the previous draws with the same allocation [65]. Let $\{\Theta_c\}_{c=1}^C$ be the set of distinct values taken by the variables $\{\Theta_m^*\}_{m=1}^{M-1}$.

The PY process functions similar to the DP. Let us suppose we randomly draw a sample distribution $G$ from a PY process, and, subsequently, we independently draw $M$ random variables $\{\Theta_m^*\}_{m=1}^M$ from $G$:

$$G|\delta, \alpha, G_0 \sim \mathrm{PY}(\delta, \alpha, G_0) \tag{4.3}$$

with

$$p(\Theta_M^*|\{\Theta_m^*\}_{m=1}^{M-1}, \delta, \alpha, G_0) = \frac{\alpha + \delta C}{\alpha + M - 1} G_0 + \sum_{c=1}^{C} \frac{\nu_c^{M-1} - \delta}{\alpha + M - 1} \delta_{\Theta_c} \tag{4.4}$$

where $\nu_c^{M-1}$ is the number of values in $\{\Theta_m^*\}_{m=1}^{M-1}$ that equal to $\Theta_c$, $\delta \in [0, 1)$

is the discount parameter, $\alpha > -\delta$ is its innovation parameter, and $G_0$ the base distribution.

This way, the PY process gives rise to a rich-gets-richer clustering property, i.e., the more samples have been assigned to a draw from $G_0$, the more likely subsequent samples will be assigned to the same draw. Further, the more we draw from $G_0$, the more likely a new sample will again be assigned to a new draw from $G_0$. These two effects together produce a *power-law distribution* where many unique $\Theta_m^*$ values are observed, most of them rarely [66], thus allowing for better modeling observations with heavy-tailed distributions. In particular, for $\delta > 0$, the number of unique values scales as $\mathcal{O}(\alpha M^\delta)$, where $M$ is the total number of draws. Note also that, for $\delta = 0$, the PY process reduces to the DP.

A characterization of the (unconditional) distribution of the random variable $G$ drawn from a PY process, $\mathrm{PY}(\delta, \alpha, G_0)$, is provided by the stick-breaking construction of Sethuraman [67]. Consider two infinite collections of independent random variables $v = (v_c)_{c=1}^\infty$, $\{\Theta_c\}_{c=1}^\infty$, where the $v_c$ are drawn from a Beta distribution, and the $\Theta_c$ are independently drawn from the base distribution $G_0$. The stick-breaking representation of $G$ is then given by [68]

$$G = \sum_{c=1}^{\infty} \varpi_c(v)\delta_{\Theta_c} \tag{4.5}$$

where

$$p(v_c) = \mathrm{Beta}(1-\delta, \alpha + \delta c) \tag{4.6}$$

$$\varpi_c(v) = v_c \prod_{j=1}^{c-1}(1-v_j) \quad \in [0,1] \tag{4.7}$$

and

$$\sum_{c=1}^{\infty} \varpi_c(v) = 1 \tag{4.8}$$

The stick-breaking representation of the PY process makes clear that the random variable $G$ is discrete. It shows explicitly that the support of $G$ consists of a countably infinite sum of atoms located at $\Theta_c$, drawn independently from $G_0$. Indeed, under the stick-breaking representation of the PY process, the atoms $\Theta_c$, drawn independently from the base distribution $G_0$, can be seen as the parameters of the component distributions of a mixture model comprising an unbounded number of component densities, with mixing proportions $\varpi_c(v)$.

## 4.4 Proposed Approach

Let us consider a dataset $\mathscr{D} = \{x_n, y_n\}_{n=1}^{N}$, where $y_n \in \{0,1\}$ is the result of the $n$th audit, with the value of 1 corresponding to an audit yield deemed satisfactory by the tax authority (i.e., exceeding some threshold). The specific selection of this threshold value used in the development of our model will be discussed in the experimental section. In our work, the observed data points $x_n$ are obtained from the VAT records of the taxpayers selected for audit. These are 47-dimensional vectors that comprise the following attributes: (i) economic activity type, classified according to the Eurostat NACE classification(ii) district codes; (iii) type of person (physical, legal); (iv) declared amounts, including VAT due/local sales, VAT due/EU purchases, VAT refundable (purchases), VAT payable, net value of sales, net value of purchases, value of zero-rated sales, value of purchases from EU (goods and services), and value of sales to EU

(goods and services).

On this basis, the audit case selection task can be framed as a binary classification task. Since RFs currently constitute the most popular machine learning approach used for audit case selection, we initiate the formulation of our model considering that the random decision variables $y_n$ can be expressed via a function $f_\theta(x_n)$, where the latent variables $\theta$ are drawn from an EBF, $\mathscr{T}(\theta)$. Further, we postulate that the classification mechanism encoded into the distribution of the random variables $y_n$ cannot be uniquely described by a single latent function $f_\theta(x_n)$, but $f_\theta(x_n)$ is only an instance of the (possibly infinite) set of possible latent functions $f_{\theta_c}(x_n)$, $c = 1, \ldots, \infty$, parameterized from different EBFs, $\theta_c \sim \mathscr{T}_c(\theta)$. Then, to determine the association between observations, $x_n$, and latent functions, $f_{\theta(\cdot)}$, we impose a PY process prior over this set of functions. The power-law nature of the PY process prior distribution allows for effectively handling cases of heavy-tailed observable data, which are prevalent in VAT audit case selection processes, as discussed previously.

Let us introduce the set of variables $\{z_{nc}\}_{n,c=1}^{N,\infty}$, with $z_{nc} = 1$ if the function modeling the correlation pattern between the observation $x_n$ and the corresponding classification decision $y_n$ is captured by the $c$th inferred (component) EBF, otherwise $z_{nc} = 0$. Based on this assumption, and the descriptions of the EBF model as well as the PY process prior, the prior configuration of the proposed PYP-EBF model is defined as follows:

$$p\big(y_n|x_n, z_{nc} = 1\big) = \text{Bernoulli}(y_n|f_{\theta_c}(x_n)) \tag{4.9}$$

$$p(z_{nc} = 1|v) = \varpi_c(v) \tag{4.10}$$

$$\varpi_c(v) = v_c \prod_{j=1}^{c-1}(1 - v_j) \in [0, 1] \tag{4.11}$$

with

$$\sum_{c=1}^{\infty} \varpi_c(v) = 1 \tag{4.12}$$

$$p(v_c) = \text{Beta}(1 - \delta, \alpha + \delta c) \tag{4.13}$$

and the $c$th inferred EBF is:

$$p(\theta_c) = \mathscr{T}_c(\theta) \tag{4.14}$$

while the functional form of the decision probability $f_{\theta_c}(x_n)$ is the mean of the probabilities pertaining to the $y = 1$ class over the trees consisting the $c$th EBF (as encoded into the inferred vector $\theta_c$).

## 4.5 Inference algorithm

Inference for nonparametric models can be conducted under a Bayesian setting, typically by means of variational Bayes (e.g., [69]), or Monte Carlo techniques (e.g., [70]). Here, we prefer a variational Bayesian approach, due to its considerably better scalability in terms of computational costs. Our variational Bayesian inference algorithm for the PYP-EBF model comprises derivation of a family of variational posterior distributions $q(.)$ which approximate the true posterior distribution over the infinite sets $Z$, $v = (v_c)_{c=1}^{\infty}$ and $\{\theta_c\}_{c=1}^{\infty}$, and the innovation parameter $\alpha$. Apparently, Bayesian inference is not tractable under this setting, since we are dealing with an infinite number of parameters.

For this reason, we employ a common strategy in the literature of Bayesian nonparametrics, formulated on the basis of a truncated stick-breaking representation of the PY process [69]. That is, we fix a value $C$ and we let the variational posterior over the $v_i$ have the property $q(v_C = 1) = 1$. In other words, we set $\varpi_c(v)$ equal to zero for $c > C$. Note that, under this setting, the treated PYP-EBF model involves a full PY process prior; truncation is not imposed on the model itself, but only on the variational distribution to allow for tractable inference procedure. Hence, the truncation level $C$ is a variational parameter which can be freely set, and not part of the prior model specification.

Let $W \triangleq \{v, \alpha, Z, \{\theta_c\}_{c=1}^{C}\}$ be the set of all the parameters of the PYP-EBF model the (posterior) distributions of which we need to train w.r.t. the available dataset $\mathscr{D}$. Variational Bayesian inference introduces an arbitrary distribution $q(W)$ to approximate the actual posterior $p(W|X,Y)$ which is computationally intractable [71]. Under this assumption, the log marginal likelihood (log evidence), $\log p(X,Y)$ becomes [72]

$$\log p(X,Y) = \mathscr{L}(q) + \mathrm{KL}(q||p) \tag{4.15}$$

where

$$\mathscr{L}(q) = \int \mathrm{d}W q(W) \log \frac{p(X,Y,W)}{q(W)} \tag{4.16}$$

and $\mathrm{KL}(q||p)$ stands for the Kullback-Leibler (KL) divergence between the (approximate) variational posterior, $q(W)$, and the actual posterior, $p(W|X,Y)$. Since KL divergence is nonnegative, $\mathscr{L}(q)$ forms a strict lower bound of the log evidence, and would become exact if $q(W) = p(W|X,Y)$. Hence, by maximiz-

ing this lower bound $\mathscr{L}(q)$ (evidence lower bound, ELBO) so that it becomes as tight as possible, not only do we minimize the KL-divergence between the true and the variational posterior, but we also implicitly integrate out the unknowns $W$. For simplicity, we consider that the posterior $q(W)$ factorizes over each one of the parameters, similar to the imposed prior (mean-field assumption [73]). By construction, this iterative, consecutive updating of the variational posterior distribution is guaranteed to monotonically and maximally increase the ELBO $\mathscr{L}(q)$ [74].

Let us denote as $\langle . \rangle$ the posterior expectation of a quantity. Based on the previous discussion, ELBO maximization yields

$$q(v_c) = \text{Beta}(v_c|\beta_{c,1}, \beta_{c,2}) \tag{4.17}$$

where

$$\beta_{c,1} = 1 - \delta + \sum_{n=1}^{N} q(z_{nc} = 1) \tag{4.18}$$

$$\beta_{c,2} = \alpha + c\delta + \sum_{c'=c+1}^{C} \sum_{n=1}^{N} q(z_{nc'} = 1) \tag{4.19}$$

Similarly, regarding the posteriors over the latent variables $Z$ that assign each data point to the inferred EBFs, we have

$$q(z_{nc} = 1) \propto \exp\left(\langle \log \varpi_c(v) \rangle\right) f_{\theta_c}(x_n) \tag{4.20}$$

where

$$\langle \log \varpi_c(v) \rangle = \sum_{c'=1}^{c-1} \langle \log(1 - v_{c'}) \rangle + \langle \log v_c \rangle \tag{4.21}$$

with

$$\langle \log v_c \rangle = \psi(\beta_{c,1}) - \psi(\beta_{c,1} + \beta_{c,2}) \tag{4.22}$$

$$\langle \log(1 - v_c) \rangle = \psi(\beta_{c,2}) - \psi(\beta_{c,1} + \beta_{c,2}) \tag{4.23}$$

On the other hand, the sampled EBFs, $\theta_c$ are inferred by resorting to the standard CART algorithm, as employed in the case of a single trained EBF [58], but presented with a subset of the available training dataset, $\mathscr{D}$. This subset is obtained by sampling from the posterior $q(z_n)$ of each training example, and collecting the set of the data points, $x_n$, with $z_{nc} = 1$.

The estimates of the posteriors prescribed above are updated consecutively and in an iterative fashion until convergence of the model ELBO. That is, on each training algorithm iteration, we update the expressions of the variational posteriors, resample assignments from the posteriors $q(z_{nc} = 1)$, and rerun the CART algorithm to obtain samples $\theta_c$ from their corresponding posteriors. This concludes the derivation of the inference algorithm of our PYP-EBF model.

## 4.6 Prediction Generation

After training the proposed PYP-EBF model on a dataset pertaining to tax audits and their outcomes, $\mathscr{D} = \{x_n, y_n\}_{n=1}^{N}$, we end up with a set of inferred EBF's with trees encoded into the vectors $\{\theta_c\}_{c=1}^{C}$. These can be used to generate predictions for unseen data $x_*$. In the context of our addressed problem of tax audit case selection, this corresponds to deciding whether a tax payer with VAT records summarized into the vector $x_*$ may result in an audit yield exceeding the

set threshold.

To this end, we employ a maximum a posteriori (MAP) rationale. Specifically, we first compute the posterior probability of the test data point , $x_*$, being assigned to the trained component EBF's. On this basis, we determine the *winner component* that maximizes $q(z_{*c})$. Then, we perform the classification task using the output probabilities of the winner EBF. We emphasize that this is in contrast to the more typically used mean-field approach, under which one would compute the average of the probabilities obtained from the inferred EBF's, weighted by the corresponding posteriors $q(z_{*c})$. However, we adopt this MAP approach as we have found it to perform consistently better.

## 4.7 Experimental Evaluation

### 4.7.1 Dataset Collection

To evaluate our approach, we have managed to get access to an extensive real-world dataset of a European tax authority. Specifically, we use a dataset comprising over 10,000 VAT returns audited in the last six years. The generated label information is set to 1 if the tax audit generated a yield which exceeded a set threshold. Following the instructions of the collaborating tax authority, we consider four *alternative thresholds*: (i) The yield value that the tax authority currently considers barely worth the required resources for audit *(Base scenario)*; (ii) this amount increased by 16%; (iii) the base amount increased by

32% and (iv) increased by 48%.[1]

### 4.7.2 Experimental Setup

The proposed approach was implemented in Python, using the scikit-learn library [75]. To allow for some comparative results, apart from our method we also evaluate the following competitors: (i) a baseline RF model [44]; (ii) the EBF algorithm that our novel PYP-EBF approach is inspired from [58]; and (iii) a popular kernel-based approach that relies on similarity criteria selected in an ad-hoc manner, namely the label propagation (LP) algorithm [76]. All the evaluated RF-type algorithms, i.e. PYP-EBF, EBF, and RF, comprised 500 samples (trees). The truncation threshold, $C$, of our approach is set to $C = 10$. This selection is reasonable, since we do not expect more than 10 distinct binary classification patterns in the limited available dataset. In all cases, the criterion used for retaining a proposed split in a sampled tree is the Gini impurity index, as suggested in [58]. The LP algorithm is evaluated using the RBF kernel, which is the default selection in the scikit-learn library. All the developed models are run on a Desktop PC, and do *not* require any specialized hardware, e.g. graphical processing units (GPUs).

### 4.7.3 Results

Our quantitative evaluation is performed on an out-of-sample basis, that is on test data different from the training set. To this end, we perform 4-fold stratified

[1] Note that, since actual VAT returns and VAT audit results are used, we are restricted from disclosure of the actual threshold values, as they constitute privileged information.

cross-validation. In figures 4.1,4.2,4.3,4.4, we concisely illustrate the obtained performance of the evaluated algorithms. Specifically, we summarize the misclassification error, precision score, recall score, and F1 metrics obtained over the conducted four folds of cross-validation in the form of box-plots. We provide this illustration across all the four considered experimental scenarios (alternative tax yield thresholds). As we observe, our approach yields a significant performance improvement over the competition, which is consistent across all the employed evaluation metrics. Even more importantly, the obtained performance appears to be robust to an increase in the adopted audit threshold value. These outcomes provide overwhelming empirical evidence that our method offers a significantly more reliable outcome than the state-of-the-art in the field, thus better addressing the need of tax authorities to maximize the returns from the audits they can perform with their limited available resources.

Further, in figure 4.5 we demonstrate the computational times required for model training and testing, both in the case of our approach and the considered competitors. As we observe, the training time of PYP-EBF is increased over the alternatives, but only moderately so. This was expected, since our proposed approach entails fitting more parameters; this normally induces some computational overhead. On the other hand, the time required for generating predictions on our test set (which comprised almost 7,500 cases) exhibits only a barely notable increase over the competition. This finding vouches for the viability of our solution, which allows for a significant improvement in the quality of the audit case selection process, without compromising computational tractability. This

is important for tax authorities, which need rapid development and response times, and cannot easily invest in high-performance computing facilities.

### 4.7.4 An Insight on the Power-Law Behavior

Further, we needed to examine how many mixture components remain effective after model training, and whether the fitted PYP-EBF model does actually yield a heavy-tailed distribution over the inferred components. To this end, in 4.6 we plot the component weight posterior expectations, $\langle \varpi_c(v) \rangle$, of the fitted PYP-EBF model, where we employed a truncation threshold $C = 10$. As we observe, our model yields two dominant components, and another three components with much lower weights. The remainder half of the initially postulated components effectively remain empty. This is an important outcome, as it corroborates both the usefulness of the power-law property of our model, as well as its capacity to infer how many components it actually needs, irrespectively of how big the truncation threshold is.

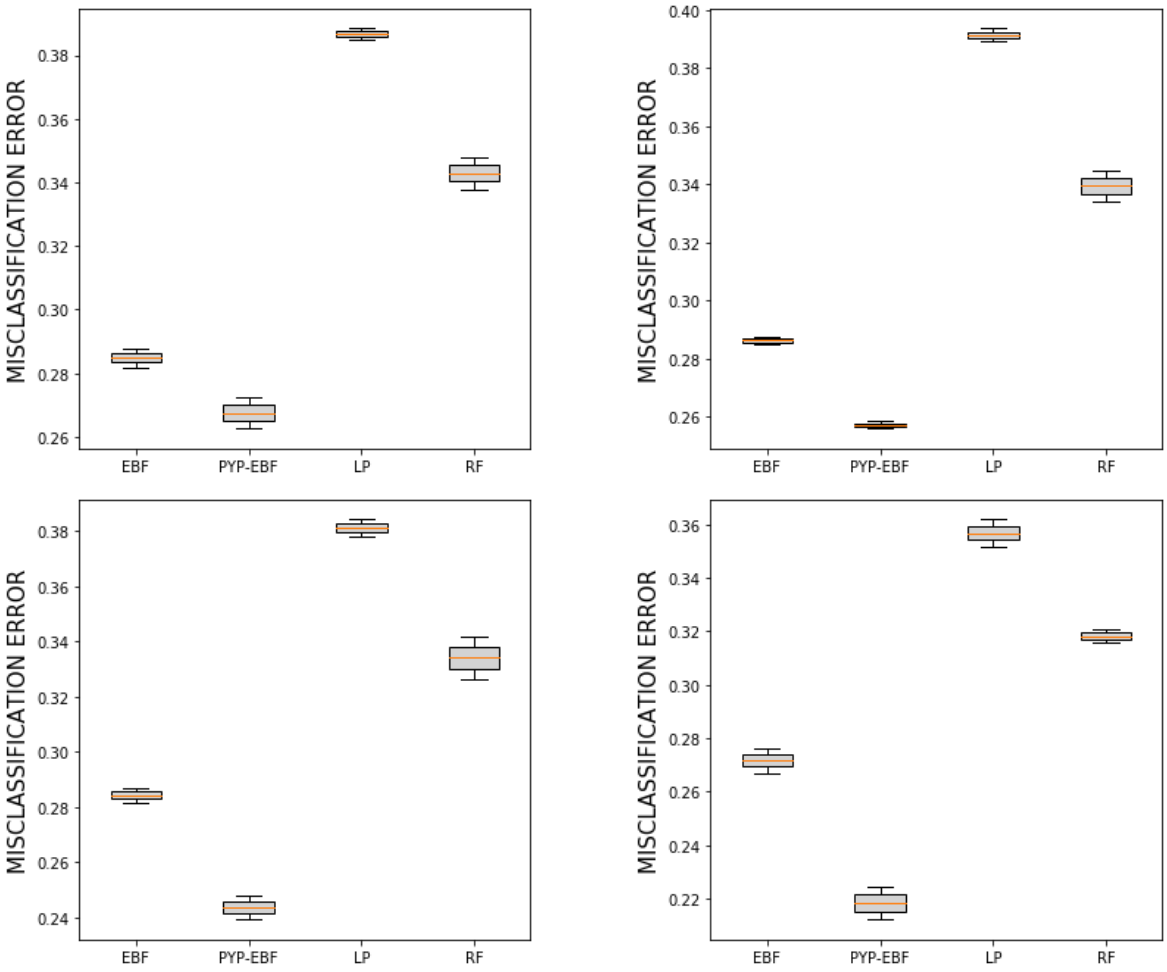


Figure 4.1: Misclassification Errors: Base Scenario + 48%, 32%, 16%, 0%, respectively.

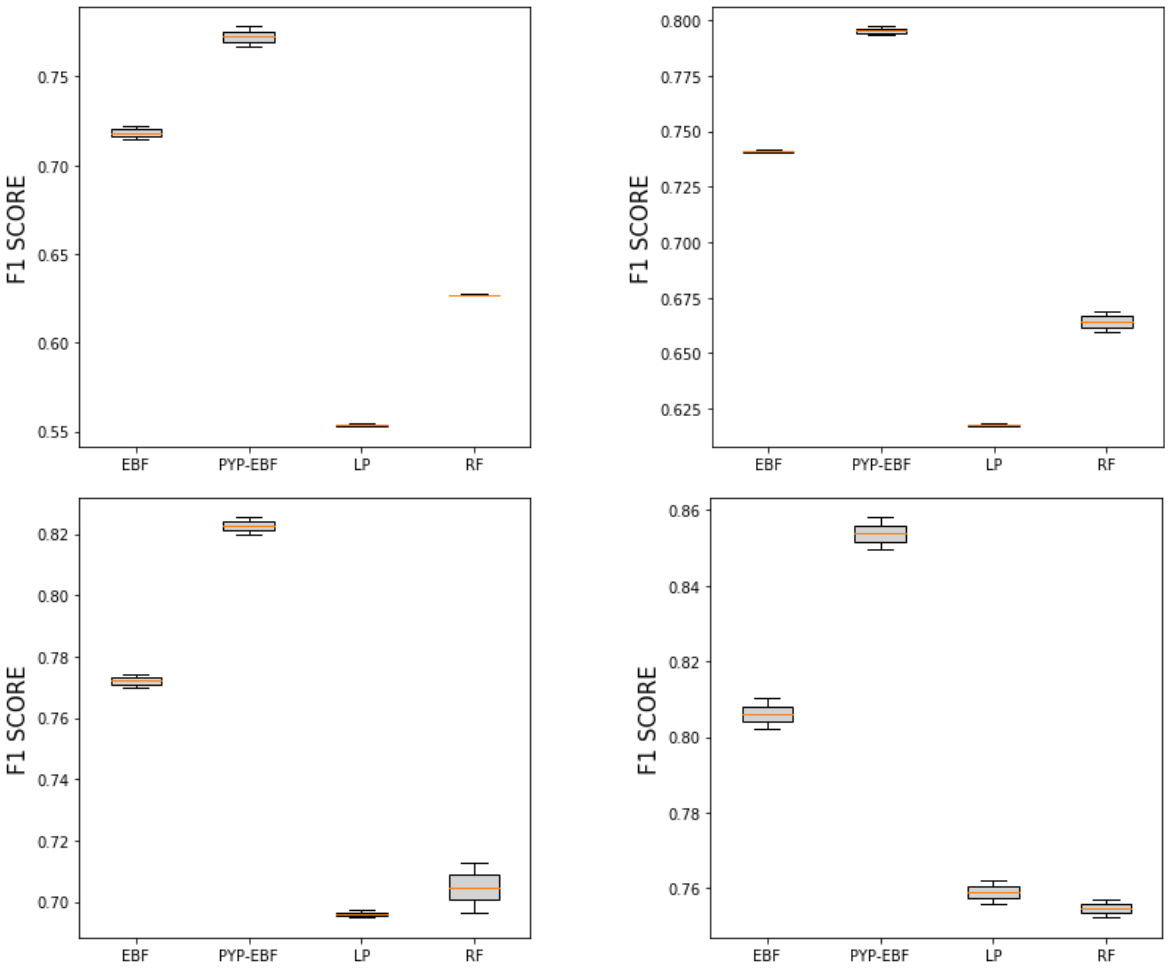


Figure 4.2: F1 Scores: Base Scenario + 48%, 32%, 16%, 0%, respectively.

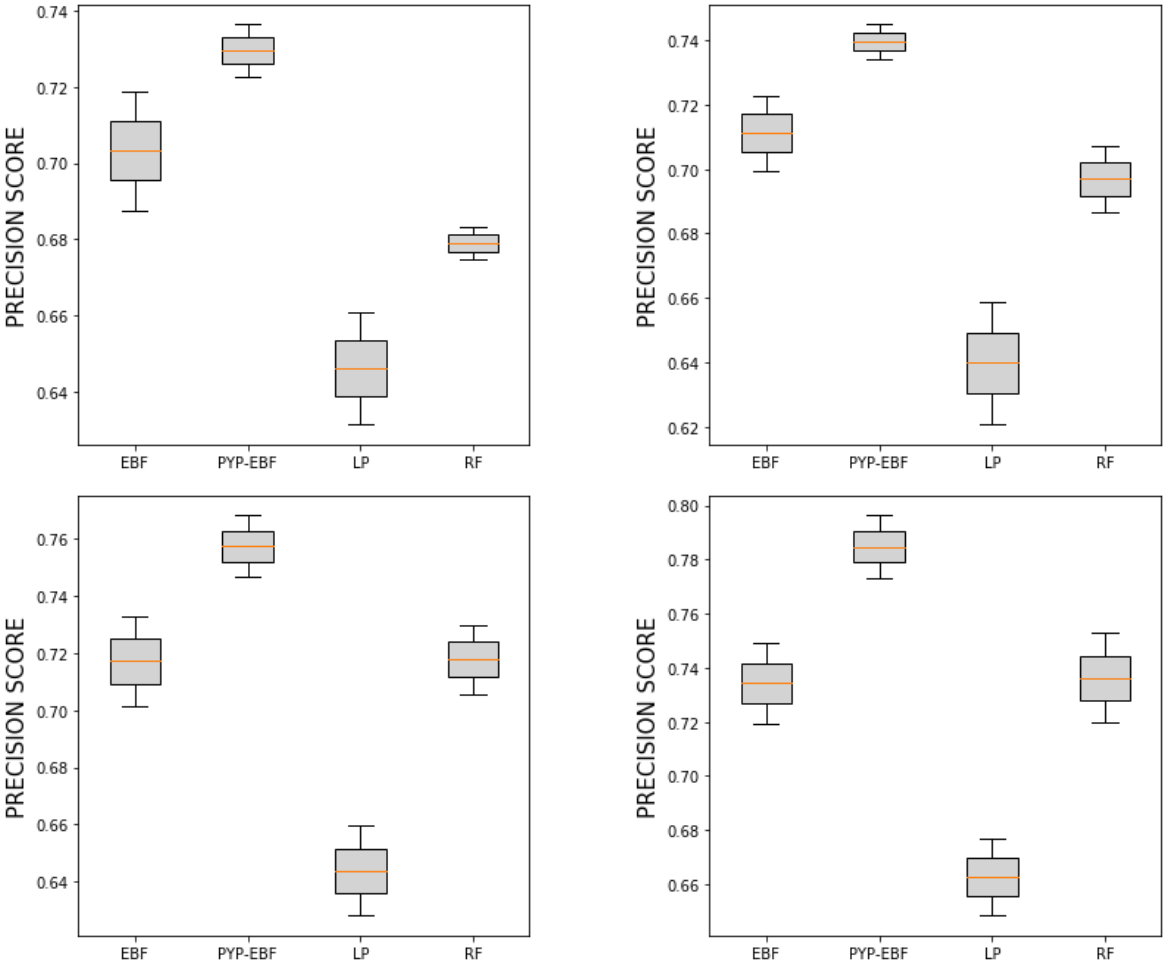


Figure 4.3: Precision Scores: Base Scenario + 48%, 32%, 16%, 0%, respectively.

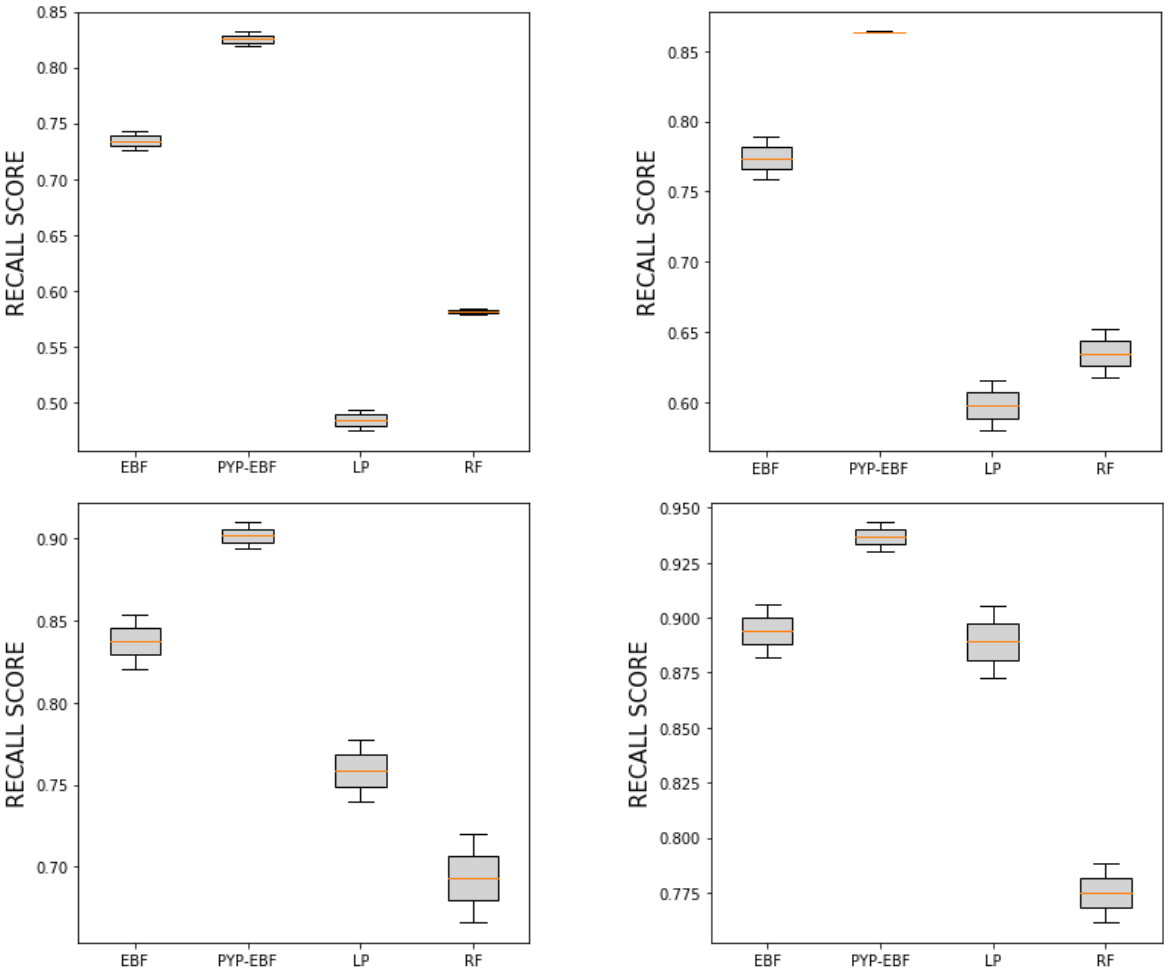


Figure 4.4: Recall Scores: Base Scenario + 48%, 32%, 16%, 0%, respectively.

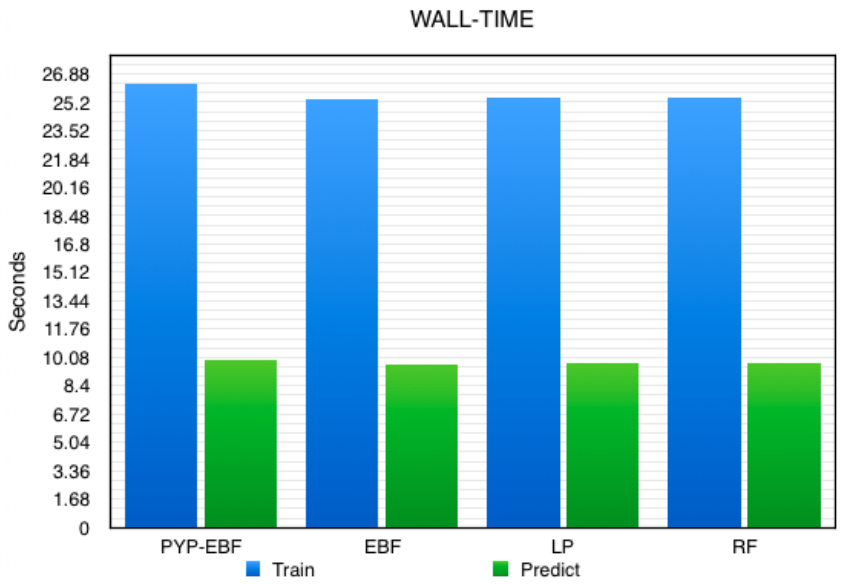


Figure 4.5: Evaluated Methods: Wall-Times of Model Training and Prediction Generation

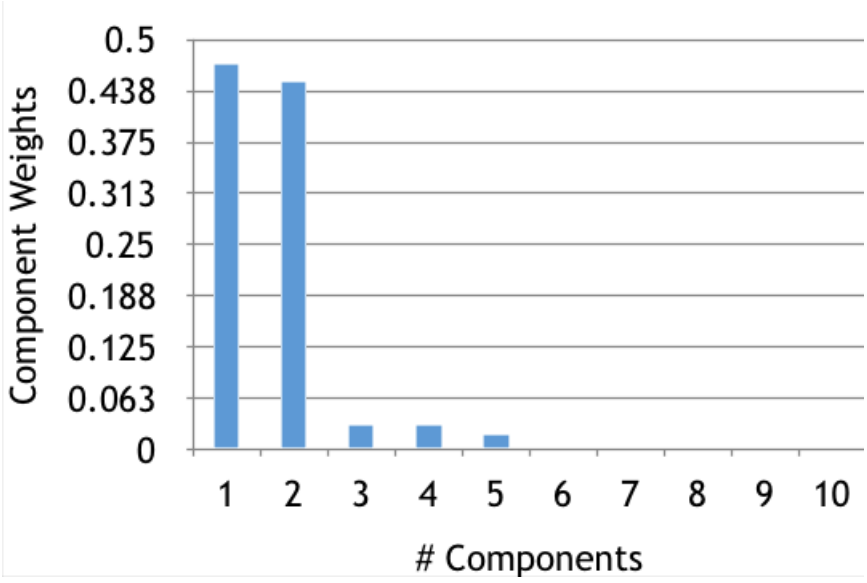


Figure 4.6: Component weight posterior expectations, $\langle \varpi_c(v) \rangle$, of the fitted PYP-EBF model.

# Chapter 5

# Gated Mixture Variational Autoencoders for Value Added Tax Audit Case Selection

## 5.1 Introduction to Deep Generative Models

Generative models learn the joint probability of data and latent variables $p_\theta(x,z)$ and are able to generate new data x . An auto-encoder is an example of generative model where the data is transformed into an abstract representation and then transformed back to the initial data. The Variational Auto Encoder (VAE), described in [77], is an example where approximate Bayesian inference can efficiently train unsupervised data. This is achieved with the help of inference networks and stochastic back-propagation. To give a closer look lets observe the components used. First there is a centered isotropic multivariate Gaussian prior over the latent variables

$$p_\theta(z) = N(z;0,I) \tag{5.1}$$

and a multivariate Gaussian with diagonal covariance for the variational approx-

imate posterior

$$\log q_{\varphi}(z|x^{(i)}) = \log N(z;\mu^{(i)},\sigma^{2(i)}) \tag{5.2}$$

where $\mu^{(i)},\sigma^{(i)}$ multilayer perceptrons outputs (inference network) and i a single datapoint. In order to achieve stochastic back-propagation they sample from the variational approximate posterior with $z^{(i,l)} \sim q_{\varphi}(z|x^{(i)})$ by using the reparameterization trick.

$$z^{(i,l)} = \mu^{(i)} + \sigma^{(i)}\varepsilon^{(l)} \tag{5.3}$$

where

$$\varepsilon^{(l)} \sim N(0,I).$$

Due to the specific choice of prior and variational posterior the KL divergence in the lower bound can be computed and differentiated without the use of sampling. Achieving the following lower bound

$$\mathscr{L}(\theta,\varphi;x^{(i)}) \simeq \frac{1}{2}\sum_{i=1}^{J}(1+\log((\sigma_j^{(i)})^2)-(\mu_j^{(i)})^2-(\sigma_j^{(i)})^2)+\frac{1}{L}\sum_{l=1}^{L}\log p_{\theta}(x^{(i)}|z^{(i,l)}) \tag{5.4}$$

where $p_{\theta}(x^{(i)}|z^{(i,l)})$ a Gaussian or Bernoulli multi layered perceptron. Bernoulli for binary and Gaussian for real valued data. VAE training is performed with the use of one sample. In [78] Importance Variational Auto Encoder was introduced, an extension to VAE that uses multi samples for training. To efficiently train

and deal with the high variance arisen from the samples they use important sampling.

### 5.1.1 Semi-Supervised Learning

In the real world problem of selecting high yield taxpayers for audit there is an increasing need for semi-supervised learning. This is a result of the abundance of tax returns (unlabeled data) and the minimal number of audited tax returns (labeled information), because of the limited number of tax auditors the number of tax returns that can be audited annually is finite. Semi-supervised learning uses labeled and unlabeled data to train the model to identify with high accuracy taxpayers with high yield in case of a tax audit. Unsupervised learning with Deep Generative Models (DGM) can be adapted to include labeled data as well, leading to semi-supervised learning. As was described by [79] with some simple modifications we can have three variations of the VAE. The first is to use the unlabeled data to train a similar to VAE approach in order to get a good abstract latent representation of the data and use it to train a classifier. This was called a latent-feature discriminative model M1. The second approach is to use the labels (target y) as an extra latent variable in line with the z latent variables. At training time the unlabeled data will be consider a latent variable and the labeled as values. This was called generative semi-supervised model M2. And finally the last approach is the stacked generative semi supervised model which stacks the two previous approaches i.e the M2 model is now infused with extra latent parameters from the M1 model. As was described by [79] the variational lower

bound for M1 is:

$$\mathscr{L}(x) = E_{q_\varphi(z|x)}[\log p_\theta(x|z) + \log p_\theta(z) - \log q_\varphi(z|x)] \tag{5.5}$$

For the M2 model the variational lower bound is extended to:

$$\mathscr{L}(x,y) = E_{q_\varphi(z|x,y)}[\log p_\theta(x|z,y) + \log p_\theta(y) + \log p_\theta(z) - \log q_\varphi(z|x,y)] \tag{5.6}$$

for the labeled data and

$$\mathscr{L}(x) = E_{q_\varphi(z|x,y)}[\log p_\theta(x|z,y) + \log p_\theta(y) + \log p_\theta(z) - \log q_\varphi(y,z|x)] \tag{5.7}$$

for the unlabeled data.

For the unlabeled lower bound it exists a classifier $q_\varphi(y|x)$ which can be used to test our model in unseen data. However this term exists only on the unlabeled data training. In order to alleviate this rable property of having a classifier trained only on the unseen data [79] introduced a classification loss in the lower bound for the labeled data.

## 5.2 Proposed Approach

### 5.2.1 Motivation

This thesis takes a different route in the effort of addressing the limited labeled training data availability that plagues the application of supervised machine

learning models to automated VAT audit selection. Specifically, we pioneer the utilization of the semi-supervised machine learning paradigm, with strong inspiration from recent developments in the field of deep learning [80].

Semi-supervised learning in its simplest form assigns predicted labels to the unlabeled data and incorporates them in the training set [81]. A number of repetitions is performed until a preset convergence criterion is met. However, this procedure may result in poor predictions being reinforced. More advanced procedures that ameliorate this risk employ graph-based methods that create a graph connecting similar observations; when a minimum energy configuration is found, the label information is propagated between labelled and unlabeled nodes [82]. The inherent limitation of this paradigm is limited scalability [80]. Another example is the Transductive Support Vector Machine (TSVM) [83] semi-supervised classification model; this enhances basic SVM's so as to use the minimum number of predicted output labels which are near the margin.

A groundbreaking paradigm that bears great promise towards resolving these issues is deep learning. Specifically, deep networks can be simultaneously trained under both the supervised and unsupervised learning paradigms. For instance, autoencoders [77] are deep network configurations that are typically trained in an unsupervised fashion, on the basis of an observation representation (encoding) and reconstruction error criterion. However, they can also be used as an intricate part of a supervised deep classifier, so as to facilitate training of the network intermediate layers by exploiting vast amounts of unlabeled data [84].

Semi-supervised learning has already been successfully applied to fraud detection tasks. For instance, Zhang in [85] proposed a binary classification of tax declarations (fraudulent/not fraudulent) using unlabeled and expert-marked data to fine-tune weights of a deep network. On a different vein, two subsets of credit card transactions were used in [86] to identify suspicious transactions. However, VAT audit selection has never been addressed before.

These facts constitute a major source of inspiration.

### 5.2.2 Model Formulation

We attempt to answer the following fundamental question: **"Can tax administrations leverage non-audited filed VAT returns to accurately predict whether a prospective audit will achieve high or low yield?"** To obtain a convincing answer, we develop a tailor-made deep learning model, whereby we cast the problem into classification as cases of high or low potential audit yield. Then, we address the introduced problem by leveraging the latest advances in the field of autoencoder deep networks, namely variational autoencoders (VAEs) [87, 77, 88].

Initially, we process the raw data described in Section 4.4 of quarterly VAT returns to obtain the observations presented to the network. Hence, the obtained measurements comprise: (i) economic activity type, classified according to the Eurostat NACE classification[1]; (ii) district codes; (iii) type of taxpayer (physical, legal); (iv) *raw* declared amounts, including VAT due/local sales, VAT

[1] https://ec.europa.eu/eurostat/documents/3859598/5902521/KS-RA-07-015-EN.PDF

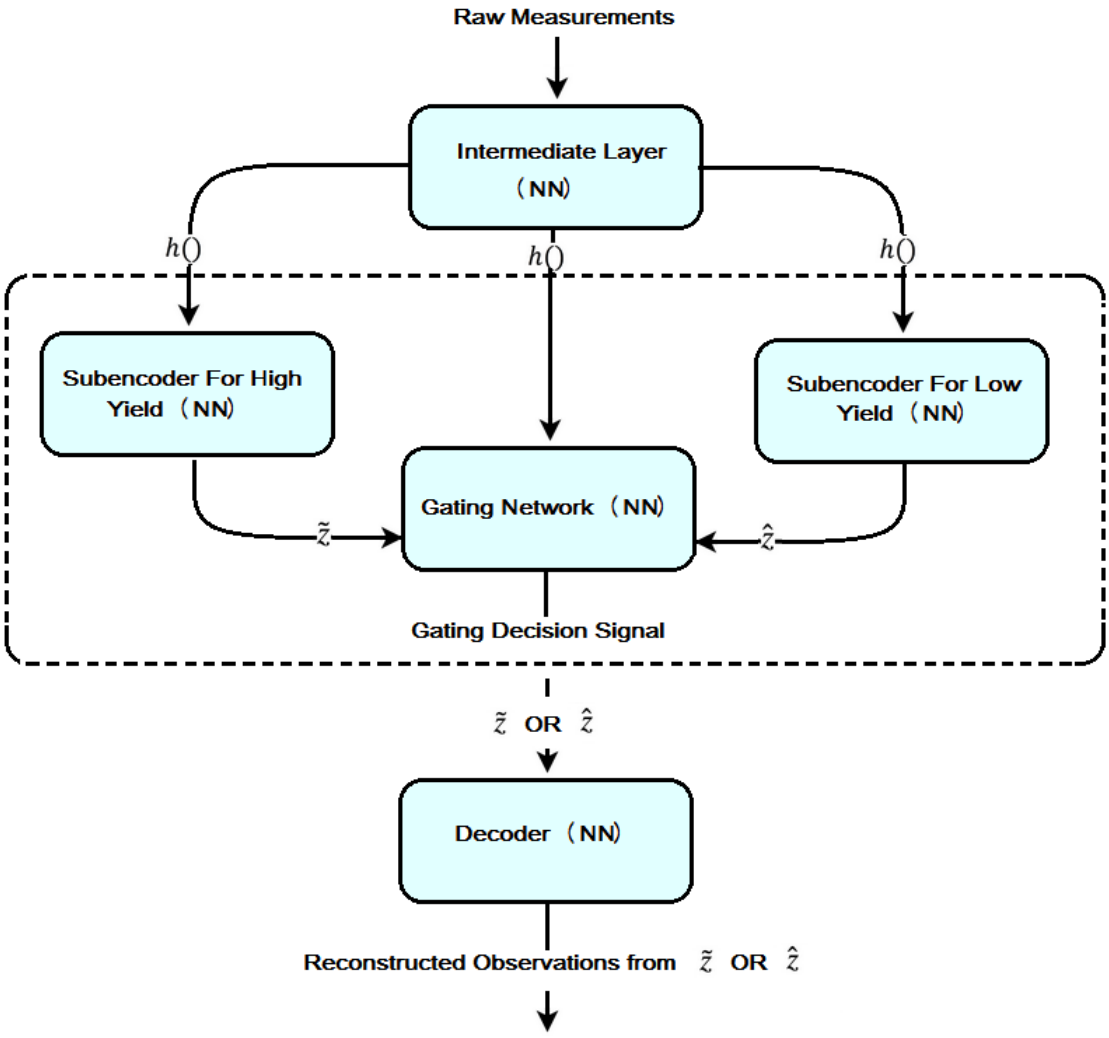


Figure 5.1: Overview of the proposed model

due/EU purchases, VAT refundable (purchases), VAT payable, net value of sales, net value of purchases, value of zero-rated sales, value of purchases from EU (goods and services), and value of sales to EU (goods and services). These raw measurements used to quantitatively describe our data were selected based on the advice of experienced field auditors, who have devised the heuristic rules currently used by the Cyprus Tax Department.

Eventually, we end up with a total of 47 raw measurements that constitute the observed data fed into the devised model. As labels associated with these observations, we use the corresponding audit outcomes, if an audit has been performed. Apparently, since only a small fraction of the filed returns are audited, most of the available data points are unlabeled.

The so-obtained observations are presented to an *encoder* network; this splits into two parts, with the first being an *intermediate* dense layer that comprises 40 ReLU units. Drawing from the recent advances in the field of variational autoencoders, e.g. [87], this encoder network facilitates the modeling process by learning to infer a high-level representation of the observed measurements. This representation is more useful for the classification process compared to the measurements themselves [87, 88]. As shown in figure 5.1, the intermediate layer of the encoder network is followed by a second part comprising two distinct *subencoders* that work in tandem. This is a radically novel modeling selection adopted in our work, which differentiates it from the existing literature. Both these subencoders are presented with the 40-dimensional output of the intermediate layer, and generate a final 20-dimensional (latent) vector, again obtained

from ReLU nonlinearities. These 20-dimensional latent vector representations (encodings) are propagated to the subsequent parts of the proposed model.

The rationale behind this novel configuration of the encoder of the devised model is motivated by a key observation; the two modeled classes (high/low yield) are expected to entail significantly different patterns of latent underlying dynamics. Hence, it is plausible that each class can be adequately and effectively modeled by means of distinct, and different, encoder distributions. We posit that learning these two distinct distributions may be best facilitated by using two subencoders. The distinct subencoder parts allow for differentiation, while the common anterior encoder part enforces our expectation that the two learned encoding distributions share some correlation.

At this point, we introduce another key modeling principle of our method. We consider that the output units of the subencoders are of a stochastic nature; specifically, we consider stochastic outputs, say $\tilde{z}$ and $\hat{z}$, with Gaussian (posterior) densities. This assumption renders our model a variational autoencoder (as opposed to a conventional autoencoder model). We strategically select to adopt the variational inference framework in developing our autoencoder model, as it is well-understood to allow for significantly improved generalization capacity and reduced overfitting tendencies [87]. Hence, what the postulated subencoders actually compute are the means, $\tilde{\mu}$ and $\hat{\mu}$, as well as the (diagonal) covariance matrices, $\tilde{\sigma}^2$ and $\hat{\sigma}^2$, that parameterize these Gaussian posteriors. On this basis, the actual subencoder output vectors, $\tilde{z}$ and $\hat{z}$, are sampled each time from the corresponding (inferred) Gaussian posteriors.

Under this mixture model formulation, we need to establish an effective mechanism for inferring which observations (i.e., analyzed VAT returns) are more likely to match the learned distribution of each component subencoder. In layman terms, this can be considered to be analogous to a (soft) classification mechanism differentiating between audit cases of high and low potential yield. This mechanism can be obtained by computation of the posterior distribution of mixture component membership (also known as ”responsibility” in the literature of finite mixture models [89]). This is also needed for effectively selecting between the samples of $\tilde{z}$ or $\hat{z}$, at the output of the encoding stage of the devised model, that will be propagated to the subsequent model components.

To allow for inferring this posterior distribution, in this work we postulate a gating network. This is a dense-layer network, presented with the same 40-dimensional intermediate representation, $h()$, as the two postulated subencoders, and using a sigmoid activation function. It is trained alongside the rest of the model, and it is the only part of the model that requires availability of labeled data for its effective training. Thus, under this model construction, the needs of our approach in labeled data availability are considerably reduced.

To conclude the formulation of the proposed model, we need to postulate an appropriate decoder distribution, and a corresponding network that infers it. In this work, we opt for a simple dense-layer neural network, which is fed with the (sampled) output of the postulated finite mixture model encoder, and attempts to reconstruct the original raw measurements. Specifically, we postulate a network comprising one hidden layer with 40 intermediate ReLU units.

Let us denote as $x_n$ the set of observable measurements pertaining to the $n$th available VAT return. Then, based on the above description, the encoder distribution of the postulated model reads

$$q(z_n|x_n) = q(\tilde{z}_n|x_n)^{q(c_n=1|x_n)} q(\hat{z}_n|x_n)^{q(c_n=0|x_n)} \tag{5.8}$$

Here, $z_n$ is the output of the encoding stage of the proposed model that corresponds to $x_n$, $\tilde{z}_n$ is the output of the first subencoder, corresponding to the high yield class, $\hat{z}_n$ is the output of the second subencoder, corresponding to the low yield class, and $c_n$ is a latent variable indicator of whether $x_n$ belongs to the high yield class or not. We also postulate

$$q(\tilde{z}_n|x_n) = \mathcal{N}(\tilde{z}_n|\tilde{\mu}(x_n;\tilde{\theta}), \operatorname{diag}\tilde{\sigma}^2(x_n;\tilde{\theta})) \tag{5.9}$$

$$q(\hat{z}_n|x_n) = \mathcal{N}(\hat{z}_n|\hat{\mu}(x_n;\hat{\theta}), \operatorname{diag}\hat{\sigma}^2(x_n;\hat{\theta})) \tag{5.10}$$

Here, the $\tilde{\mu}(x_n;\tilde{\theta})$ and $\tilde{\sigma}^2(x_n;\tilde{\theta})$ are outputs of the deep neural network that corresponds to the high yield class subencoder, with parameters set $\tilde{\theta}$. Similarly, the $\hat{\mu}(x_n;\hat{\theta})$ and $\hat{\sigma}^2(x_n;\hat{\theta})$ are outputs of the deep neural network that corresponds to the low yield class subencoder, with parameters set $\hat{\theta}$.

The posterior distribution of mixture component allocation, $q(c_n|x_n)$, which is parameterized by the aforementioned gating network, is a simple Bernoulli distribution that reads

$$q(c_n|x_n) = \text{Bernoulli}(\varpi(h(x_n);\varphi)) \tag{5.11}$$

Here, $\varpi(h(x_n);\varphi) \in [0,1]$ is the output of the gating network, with trainable parameters set $\varphi$. This infers the probability of $x_n$ belonging to the high yield

class.

Lastly, the postulated decoder distribution reads

$$p(x_n|z_n) = \mathcal{N}(x_n|\mu(z_n;\phi), \text{diag}\ \sigma^2(z_n;\phi)) \tag{5.12}$$

where the means and diagonal covariances, $\mu(z_n;\phi)$ and $\sigma^2(z_n;\phi)$, are outputs of a deep network with trainable parameters set $\phi$, configured as described previously.

### 5.2.3 Model Training

Let us consider a training dataset $X = \{x_n\}_{n=1}^N$ that consists of $N$ filed VAT returns. A small subset, $X^l$, of size $M$ of these samples is considered to be labeled, with corresponding labels set $Y = \{y_m\}_{m=1}^M$. That is, these VAT returns triggered an audit, which may have generated a high or low audit yield ($y_m = 1$ and $y_m = 0$, respectively). Then, following the VAE literature [77], model training is performed by maximizing the evidence lower bound (ELBO) of the model over the parameters set $\{\tilde{\theta}, \hat{\theta}, \varphi, \phi\}$. The ELBO of our model reads:

$$\begin{aligned} \log p(X) \geq & \mathscr{L}(\tilde{\theta}, \hat{\theta}, \varphi, \phi|X) = -\sum_{n=1}^{N} \text{KL}\big[q(z_n|x_n)||p(z_n)\big] \\ & + \gamma \sum_{n=1}^{N} \mathbb{E}[\log p(x_n|z_n)] + \sum_{x_m \in X^l} \log q(c_m = y_m|x_m) \end{aligned} \tag{5.13}$$

Here, $\text{KL}\big[q||p\big]$ is the KL divergence between the distribution $q(\cdot)$ and the distribution $p(\cdot)$, while $\mathbb{E}[\cdot]$ is the (posterior) expectation of a function w.r.t. its entailed random (latent) variables. Note also that, in the ELBO expression (5.6), the introduced hyperparameter $\gamma$ is a simple regularization constant, employed to ameliorate the overfitting tendency of the postulated decoder networks,

$p(x_n|z_n)$. We have noticed that this simple trick yields a significant improvement in generalization capacity.

In Eq. (5.6), the posterior expectation of the log-likelihood term $p(x_n|z_n)$ cannot be computed analytically, due to the nonlinear form of the decoder. Hence, we must approximate it by drawing Monte-Carlo (MC) samples from the posterior (encoder) distributions (5.2)-(5.3). However, MC gradients are well-known to suffer from high variance. To resolve this issue, we utilize a smart re-parameterization of the drawn MC samples. Specifically, following the related derivations in [77], we express these samples in the form of a differentiable transformation of an (auxiliary) random noise variable $\varepsilon$; this random variable is the one we actually draw MC samples from:

$$\tilde{z}_n^{(s)} = \tilde{\mu}_n + \tilde{\sigma}_n \cdot \varepsilon_n^{(s)},\ \varepsilon_n^{(s)} \sim \mathcal{N}(0, I) \tag{5.14}$$

$$\hat{z}_n^{(s)} = \hat{\mu}_n + \hat{\sigma}_n \cdot \varepsilon_n^{(s)} \tag{5.15}$$

Hence, such a re-parameterization reduces the computed expectations into averages over samples from a random variable with low (unitary) variance, $\varepsilon$. This way, by maximizing the obtained ELBO expression, we yield low-variance estimators of the sought (trainable) parameters, under some mild conditions [77]. We perform the maximization process of $\mathscr{L}(\tilde{\theta}, \hat{\theta}, \varphi, \phi|X)$ by resorting to AdaGrad [90].

### 5.2.4 Prediction generation

To predict the class (high/low yield) of a VAT audit case (filed return data), $x_n$, we compute the mixture assignment posterior distribution $q(c_n|x_n)$, inferred via the postulated gating network, $\varpi(h(x_n);\varphi)$. On this basis, assignment is performed to the high-yield class if $\varpi(h(x_n);\varphi) > 0.5$.

## 5.3 Method Deployment

### 5.3.1 Development process

The motivating force of this work has been the pressing need to reliably automate the VAT audit selection process for the Cyprus Tax Department. As such, development of the devised Gated Mixture Variational Autoencoder was performed with their close collaboration. Specifically, we gathered over 1,000,000 filed VAT returns as unlabeled data and over 10,000 audited VAT returns as labeled data[2]. These constitute nearly all the VAT returns of the last six years. Following the instructions of the Tax Department, and to best facilitate their needs, we have considered three *alternative model configurations*: (i) learning to detect potential audit yields exceeding €100; (ii) exceeding €75; (iii) exceeding €67; and (iv) exceeding €50.

We used this dataset to both train and evaluate our model and the considered competitors. Specifically, training was performed using the whole set of unlabeled data, and a fraction of the labeled ones under a 4-fold stratified cross-

[2]Note that, since actual VAT returns and VAT audit results from the Cyprus Tax Department are used, we are restricted from disclosure of the used data and codes, as they constitute privileged information.

validation rationale; the rest of the available labeled data was used for model evaluation (in each iteration of the 4-fold cross-validation process).

The proposed approach was implemented in Python, using the TensorFlow library [91]. The developed models were run on a Desktop PC hosting an off-the-shelf NVIDIA 10 series Graphic Processing Unit. To perform model training, we used $S = 10$ drawn MC samples, $\varepsilon^{(s)}$; we found that increasing this value does not yield any statistically significant accuracy improvement, despite the associated increase in computational costs.

To enable automatic determination of the optimal selection of model hyperparameters, which in the case of deep networks includes the number of hidden layers, the number of units in each layer, the employed nonlinearities, the used batch-size, and the selection of the Dropout and learning rates, we resorted to Neural Architecture Search (NAS) [92] which is now the state-of-the-art paradigm in Machine Learning for hyperparameter selection. Model training was performed via Adagrad.

### 5.3.2 The disappointment of a simple Dense Network alternative

Initially, we examined the efficacy of a state-of-the-art alternative to our approach. Specifically, we considered a conventional deep network which constitutes a supervised learning *alternative* to our approach. We used the available labeled data points to train this deep learning alternative, and resorted to NAS to determine its optimal configuration; this yielded two dense hidden layers with 40 and 20 ReLU units, respectively, regularized via Dropout [93] with rate 0.2.

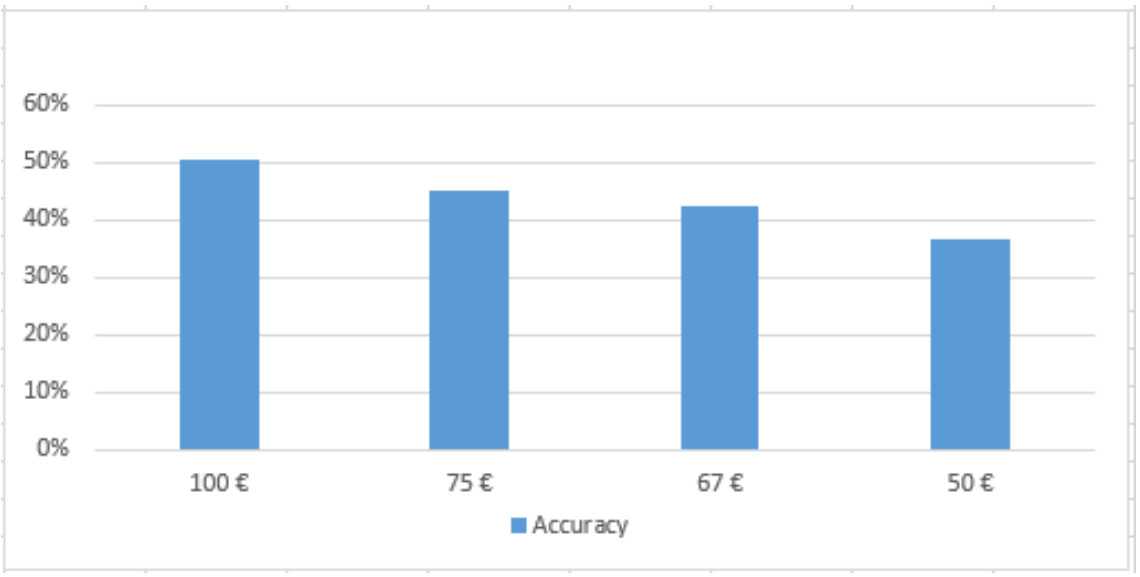


Figure 5.2: Supervised Model: Obtained accuracy for the audit yield outcomes most typically considered by the Cyprus Tax Authority.

As we illustrate below, the obtained results were far from encouraging; specifically, they were close to the random performance model accuracy (5.2) across all the four tested model configurations (€100, 75, 67 and 50). The confusion matrices (5.3) across all model configurations were also disappointing, as the outcomes are clearly imbalanced. This proves that, with this limited availability of labeled samples, a state-of-the-art supervised model fails to learn any meaningful classification pattern.

### 5.3.3 The promise of semi-supervised deep learning models

Subsequently, we proceeded to implement and deploy our proposed Gated Mixture Variational Autoencoder, using the full available dataset (both labeled and unlabeled data points). To obtain a statistically significant evaluation outcome, we performed 4-fold stratified cross-validation, as previously. In addition, to

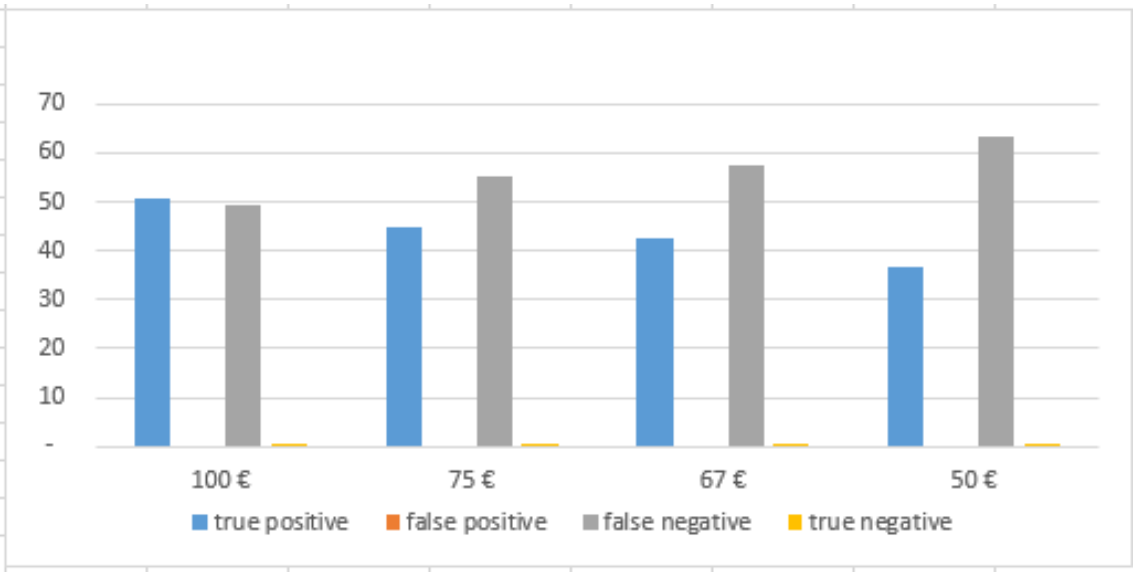


Figure 5.3: Supervised Model: Confusion matrices for the audit yield outcomes most typically considered by the Cyprus Tax Authority.

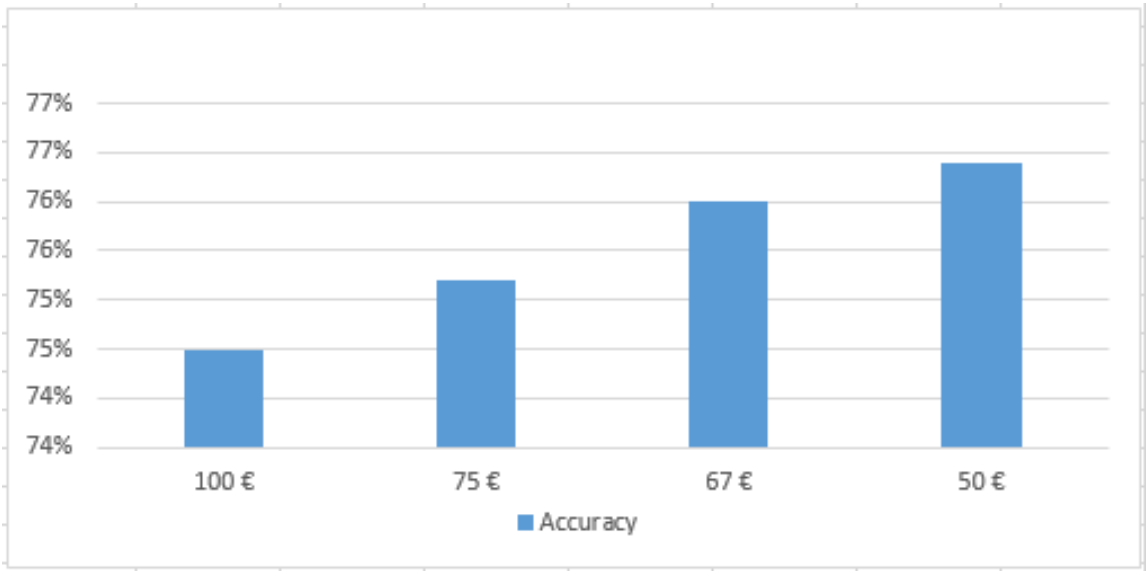


Figure 5.4: Proposed System: Obtained accuracy for the audit yield outcomes most typically considered by the Cyprus Tax Authority.

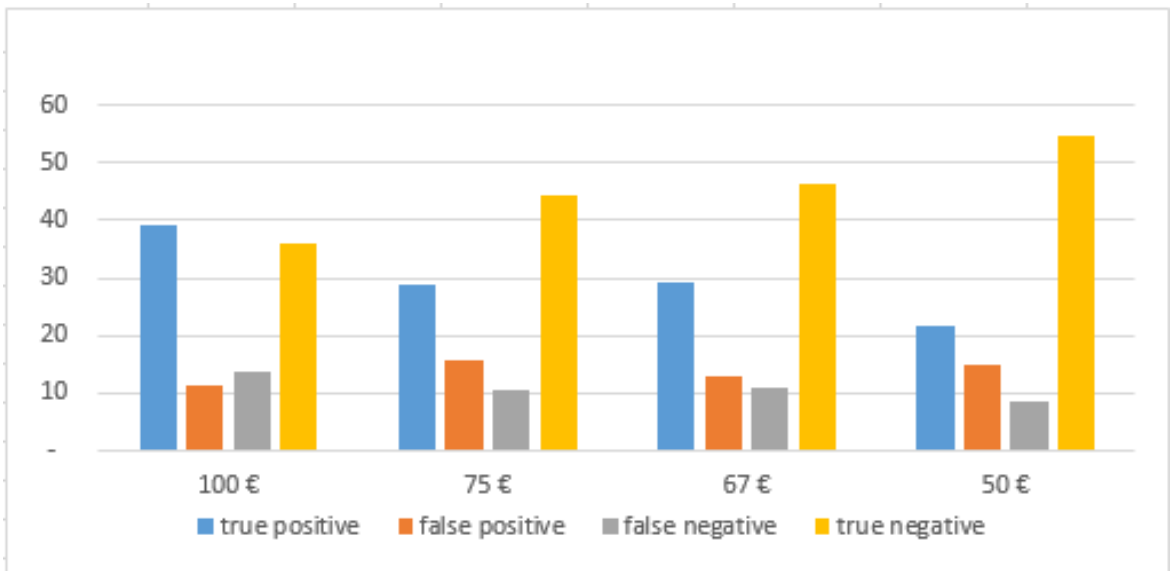


Figure 5.5: Proposed System: Confusion matrices for the audit yield outcomes most typically considered by the Cyprus Tax Authority.

obtain some *comparative results*, we also developed and deployed an existing state-of-the-art *competitor*, namely the M1+M2 semi-supervised deep learning model introduced in [94]. This model comprises a variational autoencoder with dense-network encoder and decoder, combined with a softmax classification layer; it has been shown to greatly and consistently outperform all popular semi-supervised classification alternatives, including the popular TSVM [83]. NAS yielded a M1+M2 configuration comprising 40 intermediate units and 20-dimensional latent vectors; exactly the same configuration NAS obtained for our approach.

Figure 5.4 depicts the detection accuracy obtained by our proposed system for the audit yield outcomes most typically considered by the Cyprus Tax Authority; figure 5.5 shows the corresponding confusion matrices. As we observe, despite the limited availability of labeled samples, our approach yields quite a high ac-

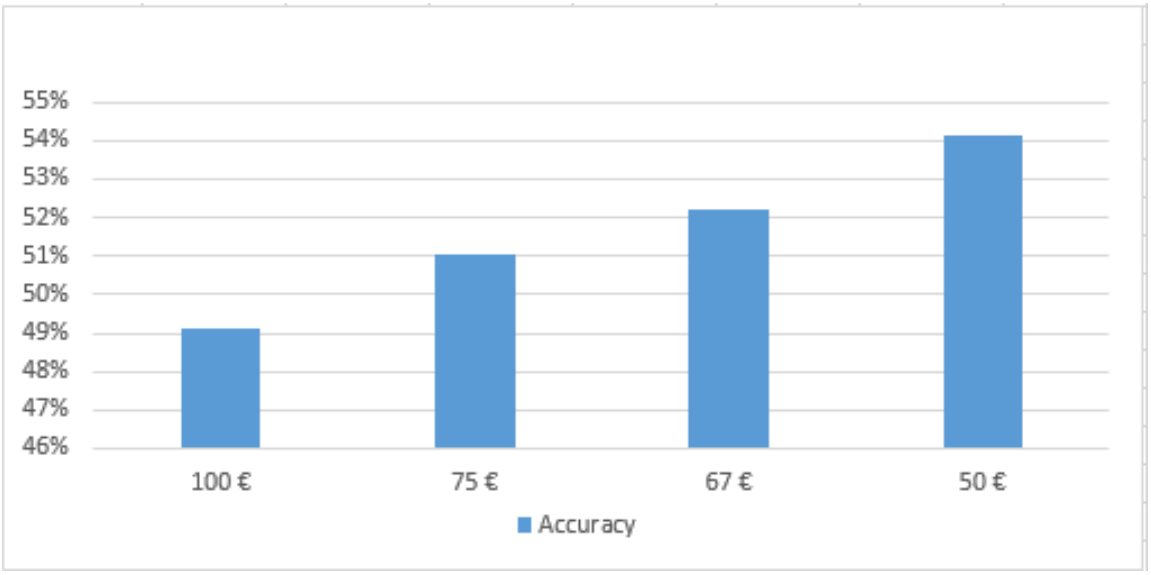


Figure 5.6: M1+M2 Model: Obtained accuracy for the audit yield outcomes most typically considered by the Cyprus Tax Authority.

curacy level across all the considered scenarios. This represents a dramatic improvement over the supervised learning alternative, providing strong evidence of the efficacy of our proposed approach and the importance of appropriately leveraging unlabeled data in the context of our addressed problem.

Further, we provide the corresponding evaluation outcomes pertaining to the considered M1+M2-based alternative. These are shown in figures 5.6 and 5.7, respectively. It becomes apparent that the M1+M2 algorithm is incapable of yielding any meaningful performance outcome, as it has barely managed to exceed 50% in all scenarios. This provides indisputable evidence of the superiority of our modeling approach, including both the proposed split of the encoder module, as well as the use of the gating network (classifier) as an integral part of the variational autoencoder. Therefore, we deduce that resorting to a state-of-the-art semi-supervised learning algorithm does not guarantee effective exploitation of

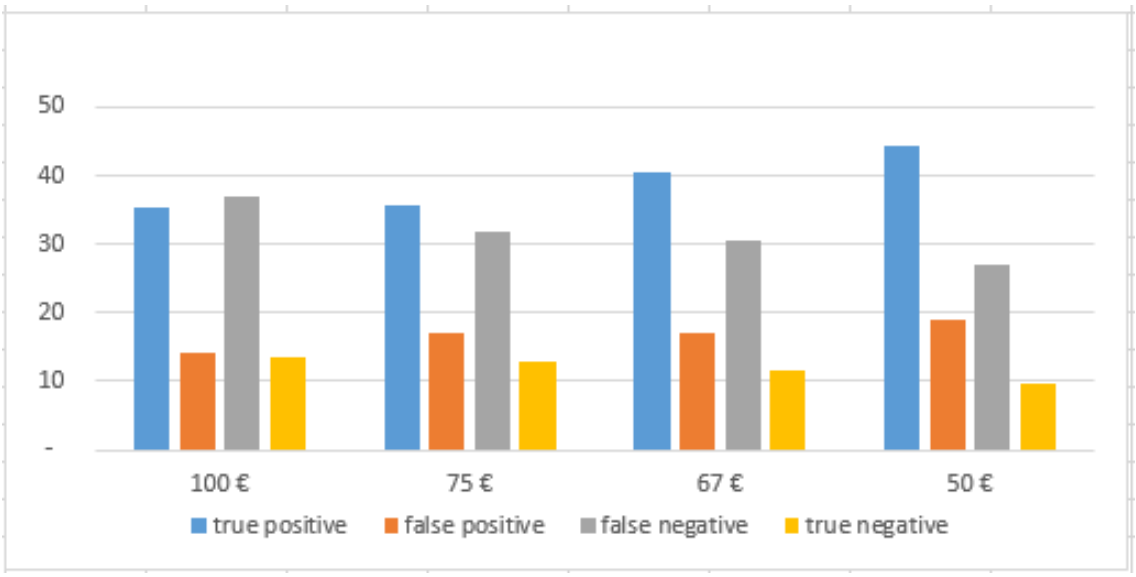


Figure 5.7: M1+M2 Model: Confusion matrices for the audit yield outcomes most typically considered by the Cyprus Tax Authority.

unlabeled data. Addressing the task at hand requires significant expertise and understanding of the problem, combined with the capacity to build upon and extend the state-of-the-art in machine learning.

### 5.3.4 Ablation study

Finally, to obtain a deeper understanding of how unlabeled training data availability facilitates the modeling performance of the proposed Gated Mixture Variational Autoencoder model, we performed an extensive ablation study. Focusing on the target audit yield of €100 outcome, we repeated our evaluation by reducing the number of used unlabeled training data points. Specifically, we examined three different test cases, where we used a randomly sampled fraction of the unlabeled data points comprising 500K, 250K and 100K samples, respectively.

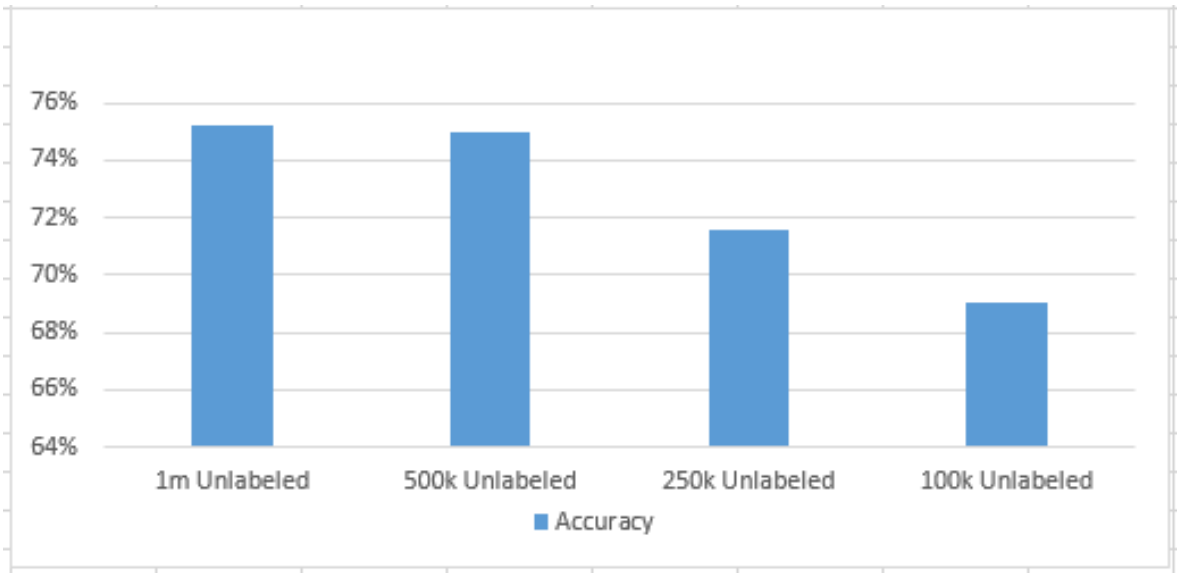


Figure 5.8: Proposed system: Accuracy variation by altering the number of used unlabeled data points (€100 audit yield detection).

The obtained results are provided in figure 5.8 (accuracy) and figure 5.9 (confusion matrices). We observe that performance remains robust as we decrease the number of unlabeled data points by 50% (500K unlabeled data points), but deteriorates if we reduce these even further. Characteristically, when only a 10% of the originally available unlabeled data is used, the accuracy drops by 7 percentage points. However, it remains profoundly better than the M1+M2 model and the evaluated supervised alternative. This constitutes conspicuous empirical evidence of the solid methodological foundation and versatility of the devised solution to the addressed problem of VAT audit case selection.

### 5.3.5 System adoption

The previous results strongly support the efficacy of the proposed system. As the €100 baseline is the targeted audit yield threshold for the rule-based systems

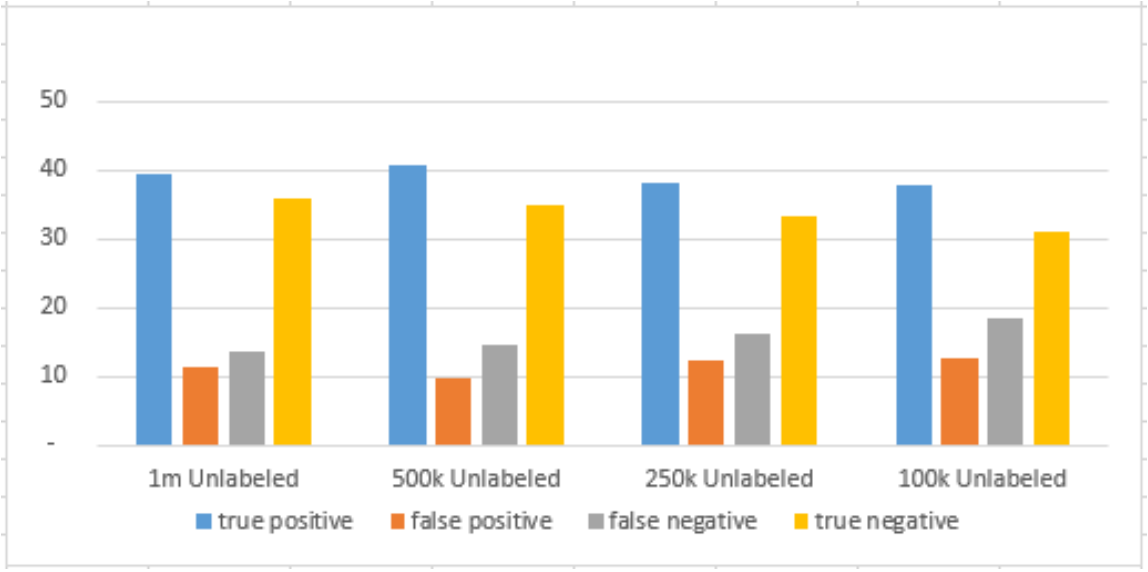


Figure 5.9: Proposed system: Confusion matrix variation by altering the number of used unlabeled data points (€ 100 audit yield detection).

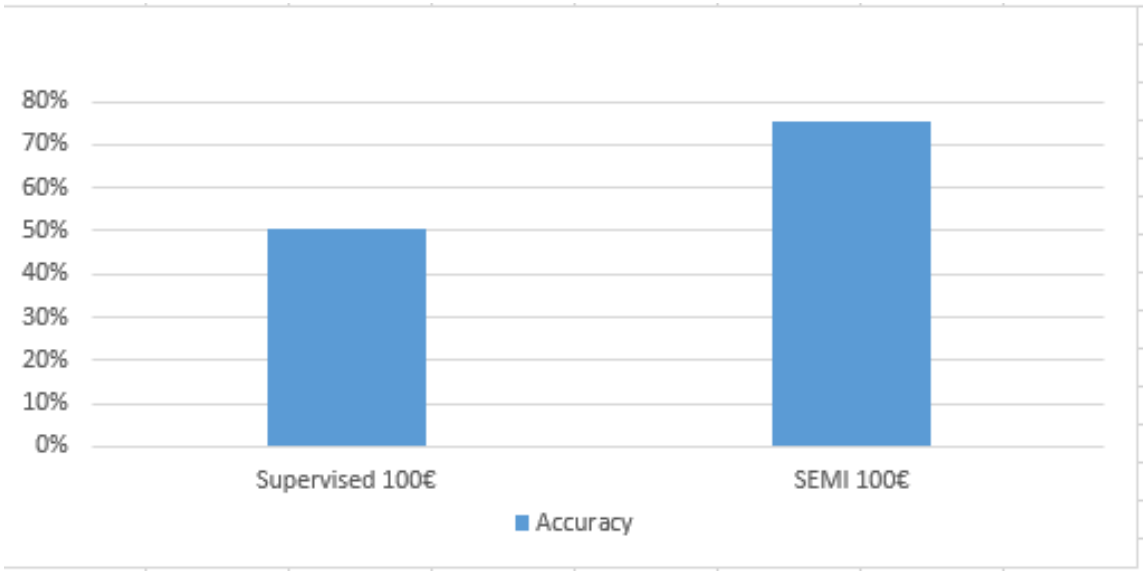


Figure 5.10: Accuracy: Supervised Vs Semi-supervised Model.

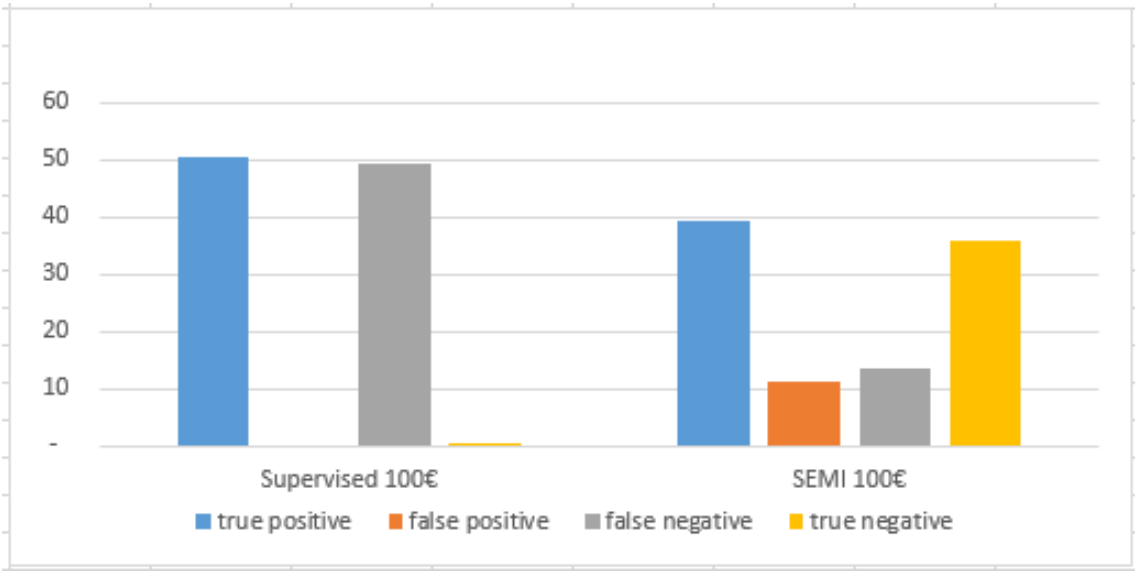


Figure 5.11: Confusion Matrices: Supervised Vs Semi-supervised Model.

currently used by the Cyprus Tax Department, it is important to stress that the obtained performance outcome offers an unprecedented level of reliability for tax auditors. Figures 5.10 and 5.11 summarize how strong the improvement of our approach is over supervised techniques. In addition, we emphasize that the *currently used rules-based systems* developed by the Cyprus Tax Department for assisting the VAT audit case selection process achieve a success rate that *fluctuates between 60-65%* (depending on seasonality effects). Note also that prediction generation using our model requires only feedforward computation encompassing the anterior part of the encoder and the gating network; as such, predictions are obtained momentarily. Hence, our thesis represents a giant leap-forward towards the goal of more effective and targeted VAT audit selection. Its full deployment, which remains open to further (longer-term) performance confirmation, is expected to eventually catalyze a significant reduction in Cyprus VAT-gap.

# Chapter 6

# Conclusions

## 6.1 Discussion on the Thesis Outcomes

The mission of the Cyprus Tax Department is the *"consistent application of the laws, ensuring fair taxation in a way that enhances the confidence of the taxpayer, the minimization of tax evasion and the effective collection of tax revenues of the state with the least possible cost."* [95]. As VAT is one of the major sources of tax evasion [2], this thesis has provided an effective automated solution to the problem of VAT audit case selection. It is expected to greatly facilitate the Tax Department in its effort to reduce tax evasion, by utilizing its limited resources (experienced auditors) to target cases with high audit yield.

Our goal was to develop a full methodological pipeline that obviates the need for tax experts to create hundreds of detailed rules; a procedure extremely time-consuming, costly, and disturbingly imprecise. At the same time, our approaches were designed to make the most out of the available audit data, taking under consideration that their availability is too limited for a supervised learning algorithm

to achieve satisfactory performance.

We developed and deployed experimental prototypes of our systems by making use of more than one million quarterly VAT returns filed in the last years, as well as 10,000 associated audit outcomes. Eventually, not only did our approaches significantly surpass the high-yield audit case selection success rate of the currently used rules-based systems, which stands at 60-65% of the audited cases. Even more profoundly, they completely outperformed popular alternative machine learning algorithms, including state-of-the-art deep networks, RFs, and Transductive SVM's.

Finally, it is worth to note that the greatest achievement of this project was the stimulation of interest within the Cyprus Tax Department for developing in-house state-of-the-art deep learning tools. Indeed, the success of our project has fostered a pro-research culture, which is especially favorable to further investment in machine learning, in close collaboration with the Academia.

## 6.2 Directives for Future Work

From the preceding discussion, it becomes apparent that the outcomes of our work bare conspicuous advantages over the existing practice and academic state-of-the-art. These facts have prodded the Tax Department to perform a set of follow-up evaluation cycles for performance verification purposes. The ultimate vision is to fully integrate the system into the Department's standard VAT audit selection practices, replacing the rules-based systems currently used.

On another vein, we also aim to pursue the examination of how our methods can be leveraged to address other sources of tax evasion. Indeed, it is common practice for tax administrations to cross-validate and reconcile items declared in the tax returns of corporation tax and VAT, like revenue; a taxpayer who filed substantially different revenue amounts should expect an enquiry from the tax authorities. Therefore, taxpayers who under-declare revenue in their VAT returns are also expected to under-declare revenue for direct taxation purposes, and vice versa, so as to avoid attracting the scrutiny of tax authorities. Since VAT evasion and direct tax evasion are correlated, a model that combines raw data from both VAT returns and direct tax returns and performs joint audit case selection for both should yield higher accuracy compared to models addressing VAT and direct taxes separately. This remains to be confirmed in the context of our future research endeavors.

# Bibliography


[1] European Commission Directorate-General for Taxation and Customs Union. *Taxation Trends*. 2018.

[2] European Commission Directorate-General for Taxation and Customs Union. "Study and Reports on the VAT-gap in the EU-28 Member States:2018 FinalReport". In: *TAXUD/2015/CC/131* (2018).

[3] Christopher M Bishop. "Pattern recognition". In: *Machine Learning* 128 (2006).

[4] Sergios Theodoridis. *Machine learning: a Bayesian and optimization perspective*. Academic Press, 2015.

[5] Christian Robert. "Machine Learning, a Probabilistic Perspective". In: *CHANCE* 27.2 (2014), pp. 62–63.

[6] Jing-Hao Xue and D. M. Titterington. "Comment on "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes"". In: *Neural Processing Letters* 28.3 (2008), pp. 169–187. DOI: 10.1007/s11063-008-9088-7. URL: http://dx.doi.org/10.1007/s11063-008-9088-7.

[7] Andrew Y. Ng and Michael I. Jordan. "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes". In: *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, Decem-*

*ber 3-8, 2001, Vancouver, British Columbia, Canada]*. Ed. by Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani. MIT Press, 2001, pp. 841–848. URL: http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.

[8] Y. Dan Rubinstein and Trevor Hastie. "Discriminative vs Informative Learning". In: *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, California, USA, August 14-17, 1997*. Ed. by David Heckerman, Heikki Mannila, and Daryl Pregibon. AAAI Press, 1997, pp. 49–53. URL: http://www.aaai.org/Library/KDD/1997/kdd97-008.php.

[9] Bradley Efron. "The efficiency of logistic regression compared to normal discriminant analysis". In: *Journal of the American Statistical Association* 70.352 (1975), pp. 892–898.

[10] Terence J. O'neill. "The General Distribution of the Error Rate of a Classification Procedure with Application to Logistic Regression Discrimination". In: *Journal of the American Statistical Association* 75.369 (1980), pp. 154–160. DOI: 10.1080/01621459.1980.10477446. eprint: http://www.tandfonline.com/doi/pdf/10.1080/01621459.1980.10477446. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1980.10477446.

[11] John Aldrich. "R.A. Fisher and the making of maximum likelihood 1912-1922". In: *Statist. Sci.* 12.3 (Sept. 1997), pp. 162–176. DOI: 10.1214/ss/1030037906. URL: http://dx.doi.org/10.1214/ss/1030037906.

[12] Michael A Babyak. “What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models”. In: *Psychosomatic medicine* 66.3 (2004), pp. 411–421.

[13] Igor V. Tetko, David J. Livingstone, and Alexander I. Luik. “Neural network studies, 1. Comparison of overfitting and overtraining”. In: *Journal of Chemical Information and Computer Sciences* 35.5 (1995), pp. 826–833. DOI: 10.1021/ci00027a006. URL: http://dx.doi.org/10.1021/ci00027a006.

[14] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning”. In: *Nature* 512 (2015), pp. 436–444.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.* Ed. by Peter L. Bartlett et al. 2012, pp. 1106–1114. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

[16] Clément Farabet et al. “Learning Hierarchical Features for Scene Labeling”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35.8 (2013), pp. 1915–1929. DOI: 10.1109/TPAMI.2012.231. URL: http://dx.doi.org/10.1109/TPAMI.2012.231.

[17] Jonathan J. Tompson et al. “Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation”. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani et al. 2014, pp. 1799–

1807. URL: http://papers.nips.cc/paper/5573-joint-training-of-a-convolutional-network-and-a-graphical-model-for-human-pose-estimation.

[18] Christian Szegedy et al. “Going deeper with convolutions”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594. URL: http://dx.doi.org/10.1109/CVPR.2015.7298594.

[19] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. “Multi-column deep neural networks for image classification”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*. IEEE Computer Society, 2012, pp. 3642–3649. DOI: 10.1109/CVPR.2012.6248110. URL: http://dx.doi.org/10.1109/CVPR.2012.6248110.

[20] George E Dahl et al. “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition”. In: *IEEE Transactions on audio, speech, and language processing* 20.1 (2012), pp. 30–42.

[21] Tomáš Mikolov et al. “Strategies for training large scale neural network language models”. In: *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE. 2011, pp. 196–201.

[22] Geoffrey Hinton et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[23] Tara N Sainath et al. “Deep convolutional neural networks for LVCSR”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE. 2013, pp. 8614–8618.

[24] Ronan Collobert et al. “Natural Language Processing (Almost) from Scratch”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2493–2537. URL: http://dl.acm.org/citation.cfm?id=2078186.

[25] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani et al. 2014, pp. 3104–3112. URL: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.

[26] Junshui Ma et al. “Deep Neural Nets as a Method for Quantitative Structure-Activity Relationships”. In: *Journal of Chemical Information and Modeling* 55.2 (2015), pp. 263–274. DOI: 10.1021/ci500747n. URL: http://dx.doi.org/10.1021/ci500747n.

[27] Michael K. K. Leung et al. “Deep learning of the tissue-regulated splicing code”. In: *Bioinformatics* 30.12 (2014), pp. 121–129. DOI: 10.1093/bioinformatics/btu277. URL: http://dx.doi.org/10.1093/bioinformatics/btu277.

[28] T Ciodaro et al. “Online particle detection with Neural Networks based on topological calorimetry information”. In: *Journal of Physics: Conference Series* 368.1 (2012), p. 012030. URL: http://stacks.iop.org/1742-6596/368/i=1/a=012030.

[29] Marvin Minsky and Seymour Papert. *Perceptrons - an introduction to computational geometry*. MIT Press, 1987. ISBN: 978-0-262-63111-2.

[30] Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar. “Kernel Regression for Image Processing and Reconstruction”. In: *IEEE Trans. Image*

*Processing* 16.2 (2007), pp. 349–366. DOI: 10.1109/TIP.2006.888330. URL: http://dx.doi.org/10.1109/TIP.2006.888330.

[31] Johan AK Suykens and Joos Vandewalle. "Least squares support vector machine classifiers". In: *Neural processing letters* 9.3 (1999), pp. 293–300.

[32] Terrence S Furey et al. "Support vector machine classification and validation of cancer tissue samples using microarray expression data". In: *Bioinformatics* 16.10 (2000), pp. 906–914.

[33] Simon Tong and Daphne Koller. "Support vector machine active learning with applications to text classification". In: *Journal of machine learning research* 2.Nov (2001), pp. 45–66.

[34] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. "The Curse of Highly Variable Functions for Local Kernel Machines". In: *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*. 2005, pp. 107–114. URL: http://papers.nips.cc/paper/2810-the-curse-of-highly-variable-functions-for-local-kernel-machines.

[35] David H Hubel and Torsten N Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of physiology* 160.1 (1962), pp. 106–154.

[36] Daniel J Felleman and David C Van Essen. "Distributed hierarchical processing in the primate cerebral cortex." In: *Cerebral cortex (New York, NY: 1991)* 1.1 (1991), pp. 1–47.

[37] Charles F Cadieu et al. “Deep neural networks rival the representation of primate IT cortex for core visual object recognition”. In: *PLoS computational biology* 10.12 (2014), e1003963.

[38] Yoshua Bengio et al. “Greedy layer-wise training of deep networks”. In: *Advances in neural information processing systems*. 2007, pp. 153–160.

[39] Christopher Poultney, Sumit Chopra, Yann L Cun, et al. “Efficient learning of sparse representations with an energy-based model”. In: *Advances in neural information processing systems*. 2007, pp. 1137–1144.

[40] Pierre Sermanet et al. “Pedestrian detection with unsupervised multi-stage feature learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 3626–3633.

[41] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.

[42] Roung-Shiunn Wu et al. “Using data mining technique to enhance tax evasion detection performance”. In: *Expert Systems with Applications* 39.10 (2012), pp. 8769–8777.

[43] Daniel de Roux et al. “Tax fraud detection for under-reporting declarations using an unsupervised machine learning approach”. In: *Proc. ACM KDD*. 8. New York, NY, USA: ACM, 2018, pp. 215–222.

[44] Chengwei Liu et al. “Financial Fraud Detection Model: Based on Random Forest”. In: *International Journal of Economics and Finance* 7 (2015), pp. 178–188.

[45] D. Cleary. “Predictive Analytics in the Public Sector: Using Data Mining to Assist Better Target Selection for Audit”. In: *Electronic Journal of e-Government* 9.2 (2011), pp. 132–140.

[46] Dan Vesset et al. “Worldwide Big Data and Analytics Software 2017 Market Shares: Healthy Growth Across the Board”. In: *IDC’s Worldwide Big Data and Analytics Software Taxonomy* US42353216 (2017).

[47] Michael Braun and Jon McAuliffe. “Variational inference for large-scale models of discrete choice”. In: *Journal of the American Statistical Association* 105.489 (2010), pp. 324–335.

[48] Amr Ahmed and Eric P. Xing. “Seeking The Truly Correlated Topic Posterior - on tight approximate inference of logistic-normal admixture model”. In: *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS 2007, San Juan, Puerto Rico, March 21-24, 2007*. Ed. by Marina Meila and Xiaotong Shen. Vol. 2. JMLR Proceedings. JMLR.org, 2007, pp. 19–26. URL: http://www.jmlr.org/proceedings/papers/v2/ahmed07a.html.

[49] TS Jaakkola and MI Jordan. “Bayesian logistic regression: a variational approach”. In: *Proceedings of the 1997 Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL*. 1997.

[50] David M. Blei and John D. Lafferty. “Dynamic topic models”. In: *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*. Ed. by William W. Cohen and Andrew Moore. Vol. 148. ACM International Conference Proceeding Series. ACM, 2006, pp. 113–120. DOI: 10.1145/1143844.1143859. URL: http://doi.acm.org/10.1145/1143844.1143859.

[51] Mohammad Emtiyaz Khan et al. “Variational bounds for mixed-data factor analysis”. In: *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing*

*Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.* Ed. by John D. Lafferty et al. Curran Associates, Inc., 2010, pp. 1108–1116. URL: http://papers.nips.cc/paper/3947-variational-bounds-for-mixed-data-factor-analysis.

[52] Antti Honkela and Harri Valpola. "Unsupervised Variational Bayesian Learning of Nonlinear Models". In: *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*. 2004, pp. 593–600. URL: http://papers.nips.cc/paper/2564-unsupervised-variational-bayesian-learning-of-nonlinear-models.

[53] Joshua Clinton, Simon Jackman, and Douglas Rivers. "The statistical analysis of roll call data". In: *American Political Science Review* 98.02 (2004), pp. 355–370.

[54] Chong Wang and David M. Blei. "Variational inference in nonconjugate models". In: *Journal of Machine Learning Research* 14.1 (2013), pp. 1005–1031. URL: http://dl.acm.org/citation.cfm?id=2502613.

[55] Danilo Jimenez Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by Francis R. Bach and David M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 1530–1538. URL: http://jmlr.org/proceedings/papers/v37/rezende15.html.

[56] Laurent Dinh, David Krueger, and Yoshua Bengio. “NICE: Non-linear Independent Components Estimation”. In: *CoRR* abs/1410.8516 (2014). URL: http://arxiv.org/abs/1410.8516.

[57] Leo Breiman. “Random Forests”. In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[58] Matt Taddy, Chun-Sheng Chen, and Jun Yun. “Bayesian and empirical Bayesian forests”. In: (Feb. 2015).

[59] S. Walker et al. “Bayesian nonparametric inference for random distributions and related functions”. In: *J. Roy. Statist. Soc. B* 61.3 (1999), pp. 485–527.

[60] R. Neal. “Markov chain sampling methods for Dirichlet process mixture models”. In: *J. Comput. Graph. Statist.* 9 (2000), pp. 249–265.

[61] P. Muller and F. Quintana. “Nonparametric Bayesian data analysis”. In: *Statist. Sci.* 19.1 (2004), pp. 95–110.

[62] C. Antoniak. “Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems.” In: *The Annals of Statistics* 2.6 (1974), pp. 1152–1174.

[63] D. Blei and M. Jordan. “Variational methods for the Dirichlet process”. In: *21st Int. Conf. Machine Learning*. New York, NY, USA, July 2004, pp. 12–19.

[64] T. Ferguson. “A Bayesian analysis of some nonparametric problems”. In: *The Annals of Statistics* 1 (1973), pp. 209–230.

[65] D. Blackwell and J. MacQueen. “Ferguson distributions via Pólya urn schemes”. In: *The Annals of Statistics* 1.2 (1973), pp. 353–355.

[66] J. Pitman and M Yor. “The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator”. In: *Annals of Probability*. Vol. 25. 1997, pp. 855–900.

[67] J. Sethuraman. “A constructive definition of the Dirichlet prior”. In: *Statistica Sinica* 2 (1994), pp. 639–650.

[68] Y. W. Teh. “A hierarchical Bayesian language model based on Pitman-Yor processes”. In: *Proc. Association for Computational Linguistics*. 2006, pp. 985–992.

[69] David M. Blei and Michael I. Jordan. “Variational Inference for Dirichlet Process Mixtures”. In: *Bayesian Analysis* 1.1 (2006), pp. 121–144.

[70] Yuting Qi, John William Paisley, and Lawrence Carin. “Music Analysis Using Hidden Markov Mixture Models”. In: *IEEE Transactions on Signal Processing* 55.11 (2007), pp. 5209–5224.

[71] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

[72] M.I. Jordan et al. “An introduction to variational methods for graphical models”. In: *Learning in Graphical Models*. 1998.

[73] D. Chandler. *Introduction to Modern Statistical Mechanics*. New York: Oxford University Press, 1987.

[74] S. Chatzis, D. Kosmopoulos, and T. Varvarigou. “Signal modeling and classification using a robust latent space model based on *t* distributions”. In: *IEEE Trans. Signal Processing* 56.3 (2008).

[75] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[76] Yoshua Bengio et al. “Efficient nonparametric function induction in semi-supervised learning”. In: *In AISTAT*. 2005, pp. 96–103.

[77] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *CoRR* abs/1312.6114 (2013). URL: http://arxiv.org/abs/1312.6114.

[78] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. “Importance Weighted Autoencoders”. In: *CoRR* abs/1509.00519 (2015). URL: http://arxiv.org/abs/1509.00519.

[79] Diederik P. Kingma et al. “Semi-Supervised Learning with Deep Generative Models”. In: *CoRR* abs/1406.5298 (2014). URL: http://arxiv.org/abs/1406.5298.

[80] Yair Weiss Rob Fergus and Antonio Torralba. “Semi-Supervised Learning in Gigantic Image Collections.” In: 522-530. Jan. 2009.

[81] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. “Semi-Supervised Self-Training of Object Detection Models”. In: *Proc. IEEE Workshops on Application of Computer Vision*. 2005, pp. 29–36.

[82] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. “Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions.” In: *Proc. ICML*. Jan. 2003, pp. 912–919.

[83] Thorsten Joachims. “Transductive Inference for Text Classification Using Support Vector Machines”. In: *Proc. ICML*. Aug. 2001.

[84] Jason Weston, Frédéric Ratle, and Ronan Collobert. “Deep Learning via Semi-supervised Embedding”. In: *Proc. ICML* (2008).

[85] Kehan Zhang, Aiguo Li, and Baowei Song. “Fraud Detection in Tax Declaration Using Ensemble ISGNN”. In: *Computer Science and Information Engineering, World Congress on*. 2009.

[86] Vladimir Zaslavsky and Anna Strizhak. “Credit Card Fraud Detection Using Self-Organizing Maps”. In: *Information & Security: An International Journal* 18 (2006), pp. 48–63.

[87] Harris Partaourides and Sotirios P. Chatzis. “Asymmetric Deep Generative Models”. In: *Neurocomputing* 241 (2017), pp. 90–96.

[88] Lars Maaløe et al. “Auxiliary Deep Generative Models”. In: *ICML*. 2016.

[89] G. McLachlan and D. Peel. *Finite Mixture Models*. 2000.

[90] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive subgradient methods for online learning and stochastic optimization”. In: *Journal of Machine Learning Research* 12.Jul (2011), pp. 2121–2159.

[91] Martıin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: http://tensorflow.org/.

[92] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. “Neural Architecture Search: A Survey”. In: *Journal of Machine Learning Research* 20 (2019), pp. 1–21.

[93] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *JMLR* (2014).

[94] Durk P Kingma et al. “Semi-supervised Learning with Deep Generative Models”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 3581–3589. URL: http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf.

[95] Cyprus Tax Department. *Vision / Mission*. 2019. URL: http://www.mof.gov.cy/mof/tax/taxdep.nsf/page04_en/page04_en?opendocument.