NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ANOMALY DETECTION USING A VARIATIONAL AUTOENCODER NEURAL NETWORK WITH A NOVEL OBJECTIVE FUNCTION AND GAUSSIAN MIXTURE MODEL SELECTION TECHNIQUE

by

Brandon Bowman

June 2019

Thesis Advisor: Matthew Norton
Second Reader: Jonathan K. Alt

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2019	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
ANOMALY DETECTION USING A VARIATIONAL AUTOENCODER NEURAL NETWORK WITH A NOVEL OBJECTIVE FUNCTION AND GAUSSIAN MIXTURE MODEL SELECTION TECHNIQUE

5. FUNDING NUMBERS

6. AUTHOR(S) Brandon Bowman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Anomalies in data often convey critical information that can be leveraged in a variety of applications. For the military engaged in combat, this can amount to identifying threats early and preserving a lethal edge over an adversary. In other more benign cases it can corrupt data integrity and lead to ineffective application of other data analysis techniques. To tackle the problem of anomaly detection, there are several common methods provided in statistics and machine learning literature, including variational autoencoders (VAEs). Using a VAE, we develop a novel objective function to improve its performance detecting anomalies. Additionally, we introduce a modeling pipeline that works in the fully unsupervised context, where one does not know the true proportion of anomalies present in the data. To construct this pipeline, we fit reconstruction errors using a Gaussian mixture model (GMM) and select the model whose characteristics best match our performance metrics. Using our approach, we observe an increase in anomalies detected against a standard objective function, and we measure an average improvement of 0.4021 in F1 scores. We show our findings using four labeled benchmark data sets and apply our conclusions on an open-source, unlabeled data set taken from USASpending.gov.

14. SUBJECT TERMS anomaly detection, neural networks, variational autoencoder, Gaussian mixture model	**15. NUMBER OF PAGES** 87
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

ANOMALY DETECTION USING A VARIATIONAL AUTOENCODER NEURAL NETWORK WITH A NOVEL OBJECTIVE FUNCTION AND GAUSSIAN MIXTURE MODEL SELECTION TECHNIQUE

Brandon Bowman
Major, United States Marine Corps
BS, Purdue University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2019

Approved by: Matthew Norton
Advisor

Jonathan K. Alt
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Anomalies in data often convey critical information that can be leveraged in a variety of applications. For the military engaged in combat, this can amount to identifying threats early and preserving a lethal edge over an adversary. In other more benign cases it can corrupt data integrity and lead to ineffective application of other data analysis techniques. To tackle the problem of anomaly detection, there are several common methods provided in statistics and machine learning literature, including variational autoencoders (VAEs). Using a VAE, we develop a novel objective function to improve its performance detecting anomalies. Additionally, we introduce a modeling pipeline that works in the fully unsupervised context, where one does not know the true proportion of anomalies present in the data. To construct this pipeline, we fit reconstruction errors using a Gaussian mixture model (GMM) and select the model whose characteristics best match our performance metrics. Using our approach, we observe an increase in anomalies detected against a standard objective function, and we measure an average improvement of 0.4021 in F1 scores. We show our findings using four labeled benchmark data sets and apply our conclusions on an open-source, unlabeled data set taken from USASpending.gov.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AUC area under the curve

CVaR Conditional Value-at-Risk

CNN Convolutional Neural Network

DATA Digital Accountability and Transparency Act

DoD Department of Defense

DoN Department of the Navy

FFATA Federal Funding Accountability and Transparency Act

GMM Gaussian mixture model

KL Kullback–Leibler

KPI Key Performance Indicator

LSTM Long Short-Term Memory

MNIST Modified National Institute of Standards and Technology

NACIS North American Industry Classification System

ReLU Rectified Linear Unit

RIS Resource Information System

SGD Stochastic Gradient Descent

USFS United States Forest Service

USGS United States Geological Survey

VAE variational autoencoder

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

Anomalous data points can present themselves in a variety of ways, and their detection is a nontrivial task that could prove critically important for many reasons. To tackle the problem of anomaly detection, there are several common methods provided in the statistics and machine learning literature, including variational autoencoders (VAEs).

Using a VAE, we develop a novel objective function to improve its performance detecting anomalies. Additionally, we introduce a modeling pipeline that works in the fully unsupervised context, where one does not know the true proportion of anomalies present in the data. To construct this pipeline, we fit reconstruction errors using a Gaussian mixture model (GMM) and select the model whose characteristics best match our performance metrics. Using our approach, we observe an increase in anomalies detected against a standard objective function, and we measure an average improvement of 0.4021 in F_1 scores.

A VAE is a neural network which learns a low-dimensional encoding of input data and then also learns a decoder that reconstructs the original input from the low-dimensional encoding. In this sense, anomaly detection can be performed by analyzing the reconstruction error. The VAE will learn to encode and then reconstruct the "normal" inputs. Inputs (data points) that differ from "normal" inputs, however, will be reconstructed poorly, and their anomalous nature will be revealed by a large reconstruction error. We analyze these reconstruction errors using a GMM.

A GMM is a combination of multiple underlying Gaussian distributions taken together to form one continuous density function. The reconstruction errors from our VAEs form the underlying Gaussian distributions. We use three clusters to represent normal, high-loss normal, and anomalous reconstruction errors.

Our new objective function is based on a concept called Conditional Value-at-Risk (CVaR), coming from the risk management literature. Central to this objective function is the introduction of a new hyperparameter, $\alpha \in [0, 1]$, which determines what proportion of largest-loss training examples (per batch) will be ignored during training, or formally:

$$\min_{\theta} E[L(X, \theta)] - (1 - \alpha)\bar{q}_{\alpha}(L(X, \theta)). \tag{1}$$

Intuitively, we desire to train the VAE on only normal data, but in an unsupervised setting, we do not know what is normal and what is anomalous. Selecting the correct α for Equation 1 effectively ignores extremely high reconstruction errors during training and trains on only (presumably) normal data. Selecting α is determined by analyzing the GMMs fit from the reconstructed errors. The clusters with the largest mean for each GMM are compared based on their mean, variance, and weight. The most distant, dense, and smallest cluster is indicative of the best α and thus highest performing VAE for detecting anomalies, which we show using labeled benchmark data.

We used four labeled benchmark datasets (three unstructured and one image) to measure the VAE's performance. Information from their labels was not used in training and only gauged the effectiveness of our approach. We used a fifth unlabeled dataset for us to apply these methods in a truly unsupervised context. The proportion of anomalies in the benchmarks ranged from 0.39% to 7.14%, their features ranged from 9 to 768, and their sizes ranged from 49,097 to 567,497. The unlabeled dataset had 997 features and 292,853 data points.

Generally, we found that our new objective function outperformed the standard objective function (i.e., when $\alpha = 1.0$). The highest performing VAE had an α near the true proportion of normal points in the data, or $1 - \alpha$ anomalies. Additionally, these results matched our intuition when we fit the reconstruction errors using a GMM. The α associated with the highest performing VAE based on F_1 Score, also matched the GMM whose third cluster was closest to our ideal characteristics. These results present an anomaly detection process that can be applied in an unsupervised setting, which we demonstrate using our fifth unlabeled dataset. We show a lower bound and upper bound of anomalies between 0.74% and 6.35%, respectively.

Acknowledgments

I'd be remiss if I didn't first acknowledge my wife's efforts and support during my time at NPS, especially as I worked on this thesis. I can't thank you enough, Katrina, and I appreciate all the sacrifices you've made over the last 12 years. I love you. To my two daughters, Hannah and Eleanor, who always found a way to ensure I was included in playtime, I love you girls.

Dr. Norton, this definitely would not have been possible without your help. Thanks so much for starting me in the right direction with this thesis and keeping me on track along the way. I'm truly grateful for all of your ideas, comments, and explanations.

Finally, to the genesis of this thesis, thank you Dr. Alt. It was an interesting topic and one I learned a great deal about.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

In the age of Big Data, where datasets are becoming larger and data sources more ubiquitous, anomalous data points present themselves in a variety of ways. Their detection is a nontrivial task and can prove critically important for many reasons. Anomalies often translate to significant actionable information in a wide variety of domains (Chandola et al. 2009). For the military engaged in combat, this can amount to identifying threats early and preserving a lethal edge over an adversary. In other situations, it can be a sign of fraudulent financial activity, illicit computer network activity, or other aberrant behavior. Even seemingly benign anomalies, such as those arising from simple data entry errors, can corrupt data integrity and lead to ineffective application of other data analysis or pattern recognition techniques.

To tackle the problem of anomaly detection, there are several common methods provided in the statistics and machine learning literature. Some include distance-based, density-based, and rank-based techniques (Mehrotra et al. 2017), with examples including nearest neighbor methods, Naïve Bayes Networks, and applications of fuzzy logic. We briefly discuss some of these methods in Chapter 2. Recently, however, techniques based upon neural networks have gained popularity due to advancements in neural network technology, which include new mathematical tools for constructing and training neural networks in addition to relevant software and hardware improvements (Goodfellow et al. 2016). In particular, researchers have just begun to explore the use of unsupervised neural network approaches like variational autoencoders (VAEs) for anomaly detection. In this thesis, we propose a novel objective function to improve the performance of VAEs in anomaly detection. Additionally, we propose a modeling pipeline that works in the fully unsupervised context, where one does not even know the true proportion of anomalies present in the dataset.

A VAE neural network (Kingma and Welling 2014) is much like a typical autoencoder, which first learns a low-dimensional encoding of the input data and then second learns a decoder that reconstructs the original input from the low dimensional encoding. In this sense, anomaly detection can be performed by analyzing reconstruction error. The network will learn to encode and then reconstruct the "normal" inputs which comprise the majority

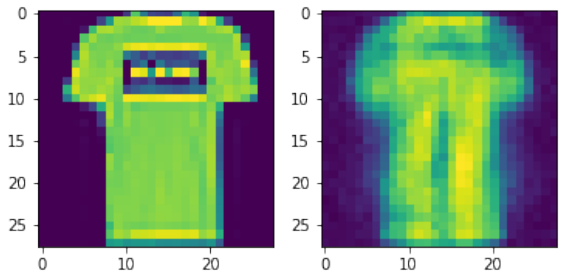
of the dataset. Inputs (data points) that differ from "normal" inputs, however, will be reconstructed poorly, and their anomalous nature will be revealed by the reconstruction error.

While an analysis of reconstruction errors seems like a simple and attractive approach, application of neural networks in this context is nontrivial due to their representational power. Neural networks have been shown, both theoretically and in practice, to be able to represent almost any function. Therefore, in the context of autoencoders, they are able to reconstruct any input with very little reconstruction error. Now, autoencoders aim to tackle this problem by creating a "bottleneck," forcing data through a low-dimensional bottleneck before reconstruction. However, this is often still insufficient in the anomaly context, and neural networks are still often able to learn to reconstruct rare training examples with high accuracy. We illustrate this in Figure 1.1 with a visual example involving a dataset with handwritten digits (normal) and articles of clothing (anomalies).

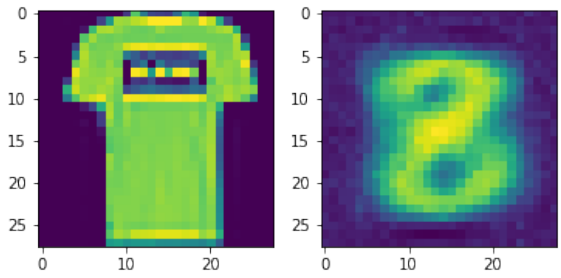
A primary goal of this work is to alter the objective function of a VAE so that it does not learn to reconstruct rare (or anomalous) inputs. We show that without alteration, it often commits errors on rare examples that are of the same magnitude as its worst reconstructions of normal examples. Our proposed alteration increases the gap between errors committed on rare examples and the worst errors committed on normal examples.

VAEs differ from traditional autoencoders as they are generative models defined via a probabilistic graphical model, combining variational inference with deep learning (An and Cho 2015). This allows a trained network to produce samples from the distribution of inputs. In the anomaly detection context, the desire is to have it learn the distribution of "normal" inputs, while committing large errors on anomalous examples. According to An and Cho (2015), VAEs also outperform standard autoencoder networks and principle component based methods with input reconstruction. This is particularly important for application in anomaly detection. Data unable to be reconstructed correctly after being processed through a VAE would be an indication of an anomaly in this context. In addition, being able to generate samples from the distribution of "normal" inputs can provide insight into what the network has learned to be considered "normal." We explore VAEs in detail in Chapter 2.

An additional challenge in anomaly detection is its unsupervised nature. Many anomaly detection tasks are in fact approached as a supervised learning problem, with a dataset of



(a) Anomaly (T-shirt) reconstructed with standard VAE



(b) Anomaly (T-shirt) reconstructed with our proposed VAE

Subfigure (a) illustrates the power of a VAE to accurately reconstruct data, regardless of class (normal or anomaly). This is undesirable for anomaly detection and what our method attempts to fix. Subfigure (b) shows our proposed method and the resulting high reconstruction error when processing anomalies. Notice that the anomalous T-shirt has been "interpreted" as the number 8, with high error, indicating a likely anomaly.

Figure 1.1. Fashion-MNIST Image Reconstruction

"normal" versus anomalous data points labeled and ready to be used for training supervised classification algorithms. Even in the unsupervised context, it is often really approached as a semi-supervised learning problem. Although, no labeled training data are available, it is assumed that one knows the approximate proportion of anomalies present in the dataset. However, real-world tasks are often fully unsupervised, with no labeled training data given and an unknown proportion of the dataset containing anomalous examples. In addition to altering the objective function of our VAE so that it works better to commit large errors on anomalies, we also propose a two step-procedure that allows it to perform strikingly well in a fully unsupervised context, where labels are now unknown and the proportion of anomalous training data is also unknown. Specifically, we utilize a Gaussian Mixture Model (GMM) to select the VAE with the most "preferable" distribution of reconstruction

errors for anomaly detection. Furthermore, the VAEs that we select from are all trained using the proposed novel objective function, differing only in the value of an important hyperparameter used to create the objective function.

This thesis seeks to design and leverage a VAE neural network as a means to identify anomalies within data. In Chapter 3, we further explore our methodology using a novel objective function and model-fitting technique to shape the distribution of reconstruction errors to better indicate anomalous data points. Additionally, we use four benchmark datasets as well as an open-source, unlabeled Department of the Navy (DoN) contract award dataset from USASpending (2019). Our choice of data enables us to take advantage of labels and build inferences about our method's performance as well as apply what we discover in a military context.

1.1 Problem Statement

In statistical regression, outliers are identified by their residual distances away from a fitted curve. The curve is produced from the relationships between a response variable and dependent variable(s). Anomaly detection is straightforward in this manner since there are parametric measures that can be used to distinguish normal from abnormal data like Cook's Distance.

In contrast to statistical regression, supervised machine learning techniques use labeled data to create and train and neural networks. These labels allow the network to receive feedback on its ability to accurately classify outcomes. This is useful when making future inferences on similar data that is unlabeled. Support Vector Machine Learning does exactly this and is useful for classification problems, such as anomaly detection, when a labeled training dataset is available.

Classifying unlabeled data is more challenging and is the primary domain in which this thesis focuses. Given the open-source, DoN contract award dataset, anomalous data points must be pulled out without any true indication of accuracy. This poses a problem and one that we attempt to solve using a VAE neural network.

1.2 Research Questions

The immediate contribution of this thesis is a new objective function for training VAEs in anomaly detection combined with a parameter selection technique based on GMMs that produces a fully unsupervised anomaly detection procedure where one needs zero labeled training data and no information about the proportion of anomalies present in the data. The techniques implemented are not tailored to any specific data type and can be applied to a variety of situations. Succinctly, this research seeks to answer the following questions:

- What is the best network architecture to effectively detect anomalies and how can we alter the objective function of the VAE to better detect them? Similarly, can we force the distribution of reconstruction errors to follow a desired distribution with anomalous data associated with large reconstruction errors that are far from the distribution of normal reconstruction errors (e.g., long-tail and bimodal)?
- How should we analyze the distribution of reconstruction errors to indicate anomalies (i.e., how large does the reconstruction error need to be for a point to be labeled anomalous)? Without ground-truth labels, how do we know if our algorithm is performing well?
- Can our proposed technique be generalized to perform effectively on many datasets? If so, how well?

1.3 Scope

Our primary effort in this thesis is developing a novel method that can be used for anomaly detection. We use various datasets to explore our approach. We first utilize labeled benchmark data to test the effectiveness of our approach. Note that we do not use these labels in training, but require them to test its actual detection performance compared to the standard VAE approach. We apply our network to both image and unstructured data and see that it is effective in both cases. Finally, we apply our method to DoN contract data, leaving a specific analysis of the anomalies detected to future work.

1.4 Thesis Organization

Chapter 1 provides an introduction to the topic and provides an overview of the problem. Chapter 2 examines anomalies, common detection techniques, and describes VAEs in detail.

It also reviews the primary literature used in writing this thesis as well as related work using VAEs used for anomaly detection. Chapter 3 explains the process of preparing the data for use as well as the methods used in creating the VAE. Chapter 4 discusses the results and provides analysis of our method's performance. Chapter 5 provides a summary of the work conducted and recommendations for further study.

CHAPTER 2:
Background

This chapter provides the requisite background information and theory necessary to understand the methodology used in this thesis. It begins by first defining an anomaly and its different types. Next, a broad overview of machine learning methods for detecting anomalies are discussed along with their relative strengths and weaknesses. Our chosen method, a VAE neural network, is then examined in detail. Basic mathematical concepts are included. Three cases are briefly discussed demonstrating the successful use of VAEs for detecting anomalies. Finally, the size and scope of the FY18 Department of the Navy (DoN) budget is provided to form an appreciation of the magnitude of total annual spending by the service as well as examples of misuse, neglect, and/or fraud cases within the entire military involving government contractors.

2.1 Anomalies Defined and Challenges with Detection

Observations within data that do not conform to well-defined normal behavior are said to be anomalies, and their detection refers to the problem of finding patterns in data that do not conform to expected behaviors (Chandola et al. 2009). There are three kinds of anomalies that are typically described in literature: point, collection, and contextual. For this research, we primarily concern ourselves with point anomalies but define each for thoroughness.

2.1.1 Point Anomalies

A point anomaly is a data instance that is labeled anomalous with respect to the rest of the data. It is the simplest type of anomaly and typically consist of a single, or small group of points, that is distant in some measurable way from the large majority of other points in the data. Figure 2.1(a) provides a visual representation this type.

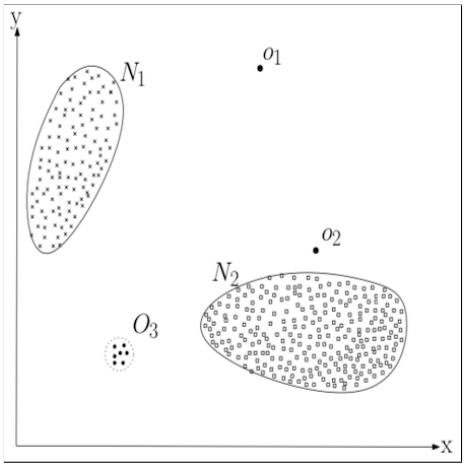
2.1.2 Collective Anomalies

A collective anomaly is one that involves a subset of the data with respect to the full set. Chandola et al. (2009) describes it as a collection of related data instances whose

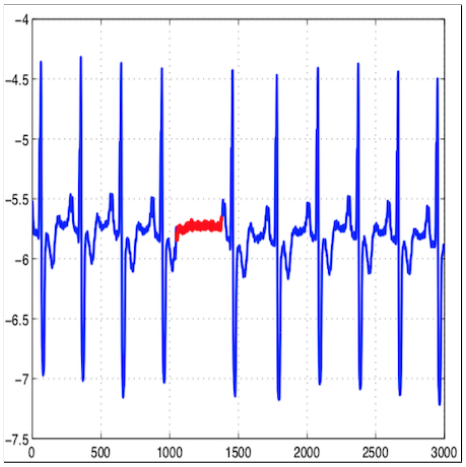
occurrence together is anomalous but when observed individually is not anomalous (Figure 2.1(b). Consider rocket attacks on a forward operating base. If an enemy typically fires two rockets before three salvos of 10 rockets apiece, then it would be considered anomalous if the forward operating base received a sequence of four dual-rocket salvos. While receiving a volley of only two rockets is normal, it is an anomaly if it occurs repeatedly in this case. The pattern is important.

2.1.3 Contextual Anomalies

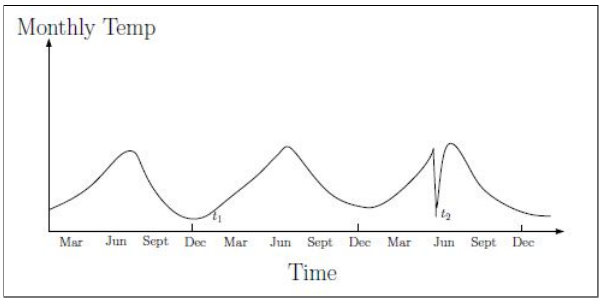
A contextual anomaly, also sometimes called a conditional anomaly (Song et al. 2007), is one that is dependent on the structure of the data it resides (Chandola et al. 2009). Time-series data is prime for exploring and detecting contextual anomalies, if they exist. An example would involve the effects of seasonality and air travel. High volumes of travelers are expected and considered normal for certain holidays, like Christmas; however, if the same volume occurred on some Wednesday in April then this could be considered anomalous. The volume of travelers with respect to their occurrence is what is unusual, not the volume travelers by itself. Figure 2.1(c) provides a visualization of a contextual anomaly.



(a) Point Anomalies



(b) Collective Anomalies



(c) Contextual Anomalies

These subplots represent the three different types of anomalies that occur. (a) represents a point anomalies. Data instances O_1, O_2, and O_3 are anomalies since they are not contained within N_1 or N_2. (b) represents a collective anomalous electrocardiogram. The red band is within the normal range; however, the duration of the signal is anomalous. (c) represents a context anomaly. t_1 and t_2 are the same temperature; however, t_1 occurs in the winter as expected while t_2 occurs in the summer making t_2 anomalous.

Figure 2.1. Three Types of Anomalies. Source: Chandola et al. (2009).

2.1.4 Challenges with Detection

Consider the often-used, simple example of anomaly detection in the credit card industry. Credit card companies collect large amounts of financial transaction information from their cardholders. Over time, these companies are able to model a cardholder's expected behavior or compare similar cardholders to one another. This forms the basis of normal data on their

cardholders. If a cardholder, or an entity pretending to be a cardholder, is making a purchase that does not conform to his or her well-defined normal behavior, then the purchase is labeled anomalous and an alert is transmitted for verification. Although this example is easy to understand, anomaly detection is a very challenging and nontrivial pursuit. One challenge with anomaly detection is the proportion of normal and anomalous data is often unknown. Chandola et al. (2009) presents four other factors relevant to this thesis that highlight the difficulties detecting anomalies:

- Defining a normal region is very difficult, and points near its boundary carry a higher chance of being mislabeled.
- Anomalies resulting from malicious activity are often masked to appear normal to avoid detection.
- Labeled data for model training and validation is often unavailable.
- Data often contains noise similar to actual anomalies making them hard to distinguish.

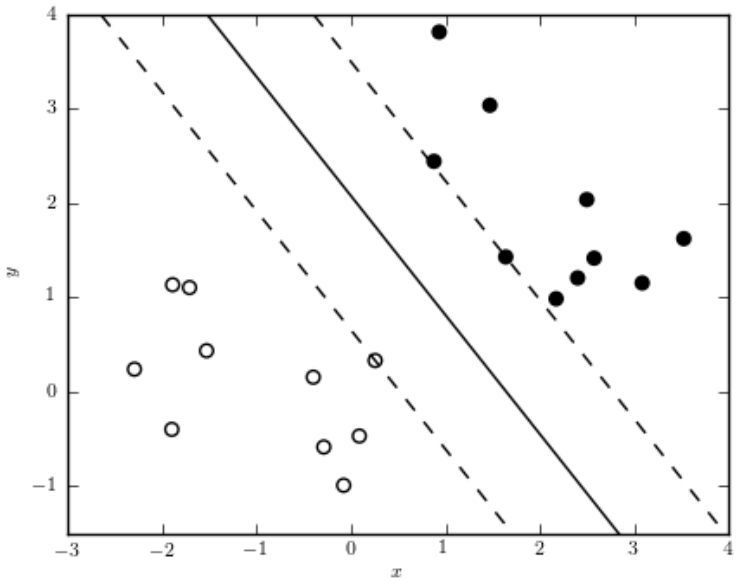
2.2 Common Anomaly Detection Techniques

In Machine Learning, training falls into one of three categories: supervised, semi-supervised, and unsupervised learning. With anomaly detection, supervised learning is when the training and testing data are both labeled as either normal or anomalous; however, it is rare to have data labeled in this fashion. Semi-supervised learning is the setting in which some of the data are labeled and some are unlabeled. Additionally, semi-supervised could be considered to be any setting in which "partial" information is available, such as the proportion of anomalies present in the data without explicit labels. Finally, unsupervised learning, which is the primary domain we concern ourselves with in this thesis, is when a model is trained using unlabeled data. Additionally, anomaly detection techniques can be generalized further into four broad areas.

2.2.1 Classification

Neural networks, support vector machines (see Figure 2.2), Bayesian networks, and rules-based techniques all fall within this area. For each, a model is trained on either one- or multi-class training data with the goal of learning a feature space that can differentiate between normal and anomalous points in test data. According to Upadhyaya and Singh

(2012), the advantages of this category are a fast testing phase since test data is input into a pre-computed model as well as the ability to utilize powerful computing algorithms for training. In contrast, Upadhyaya and Singh (2012) offer two primary disadvantages with this group of approaches. First, it requires accurate training labels which is often not feasible or even possible in most data. Second, test instances are classified as either normal or anomalous which may be too rigid when instances are near boundaries.



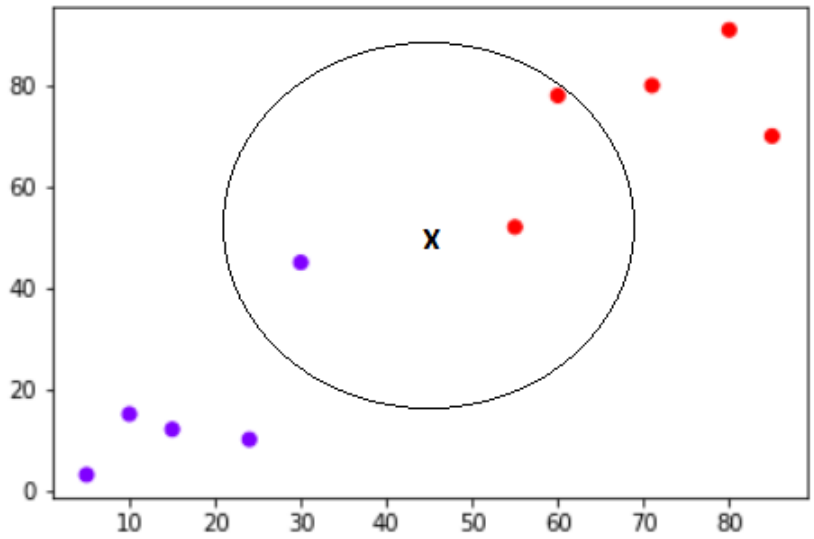
Two classes are depicted, filled and unfilled dots. The solid line represents the boundary between to classes and the dotted lines are the maximum distances to the nearest point in each class. The boundary distinguishes the classes.

Figure 2.2. Support Vector Machine. Source: Vanderplas et al. (2012).

2.2.2 Nearest-Neighbor

This area can be used for both classification and regression outputs, with its primary means of anomaly detection occurring through measurements in distance similarities. Data instances in some n-dimensional space are grouped according to their k nearest neighbors with the objective being to minimize the distances between neighbors (see Figure 2.3). Using this method, anomalies are detected when distances are unusually large between points. Upadhyaya and Singh (2012) state an advantage of using k nearest neighbors is

assumptions are not made regarding the distribution of the data; however, they further explain a significant disadvantage occurs if the normal data instances have distant neighbors and the anomalous data has close neighbors. This results in mislabeling.

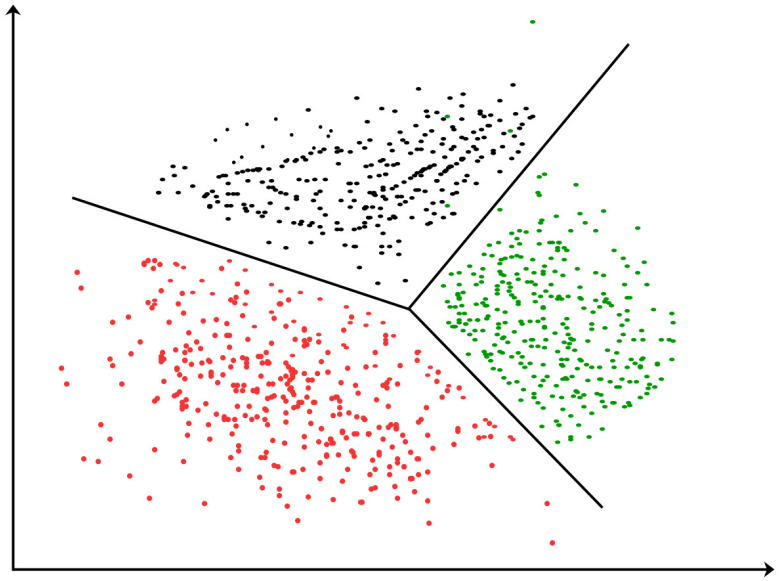


Point x is the unknown target point and the black circle surrounding it subsets the k nearest neighbors. In this case $k = 3$. The target point x can be one of two different classes: red or purple. Its class is determined by the majority class of the k nearest neighbors. For this example, x would be classified as belonging to the red class.

Figure 2.3. K-Nearest Neighbor. Source: Robinson (2018).

2.2.3 Clustering

The principal aim of clustering is to identify groups of similar instances within the data. Various algorithms attempt to identify the optimal number of clusters to accurately partition the data. Data instances that do not easily fall into any of the clusters are considered anomalous. Clustering can use several metrics for determining their centroids. K-means and k-mediod clustering are both common methods in literature. Chandola et al. (2009) describes one prime advantage of this technique is its fast computation in the testing phase. A significant disadvantage is anomaly detection by clustering is often a secondary result and thus this method is not optimized for this task. Figure 2.4 provides an example of 2-D clustering.



A scatterplot with three different clusters belonging to 3 different classes (red, green, and black) is partitioned.

Figure 2.4. Clustering. Source: Priy (2019).

2.2.4 Parametric Techniques

Using parametric statistical methods to detect anomalies falls into the realm of supervised learning. A response variable is selected for the data and predictors are fit according to some statistical approach (e.g., linear or logistic regression). For instance, linear regression would be suitable for detecting anomalies since the measure of residuals could be used to test outliers. The larger the residual value than the greater the possibility the associated data instance is an anomaly. Figure 2.5 shows a rudimentary analysis for anomalies using basic linear regression. A key advantage to using a statistical or modeling approach is the ability to use parametric techniques to identify anomalies, however, this only works if an appropriate response variable is chosen which can be difficult or unfeasible for some data.

Outlier, or anomalous points, are circled in red and are distance from the expected value, or fitted regression line.

Figure 2.5. Linear Regression with Outlier. Source: Laerd Statistics (2019).

2.3 Neural Networks and Variational Autoencoders

2.3.1 Neural Networks

Neural networks are a wide class of flexible nonlinear pattern recognition models (Sarle 1994) and are also loosely based on the biological brain in that they consist of interconnected neurons. These neurons can be turned on or off, in a sense, by an activation function. Weights are added to the values of each neuron's output. This process of weight summation and activation using a nonlinear activation function is repeated for each hidden layer in the neural network (see Figure 2.6). The final layer of a neural network is its output.

One primary benefit of neural networks is their unrivaled ability to adapt to the structure of the input data via their layer architecture. Much of the recent progress in neural network research has been pushed by novel constructions of network layers tailored to specific data structures. For example, if image data is the desired input, one only needs to change the layers to convolutional layers and performance will be increased dramatically, while most other modeling components stay generally the same (gradient descent, the objective function, etc.). Therefore, our proposed VAE can be applied to multiple input data types by only changing the network architecture, with our methodology remaining largely unchanged.

The genesis of neural networks was in 1943 by Warren McCulloch and Walter Pitts with

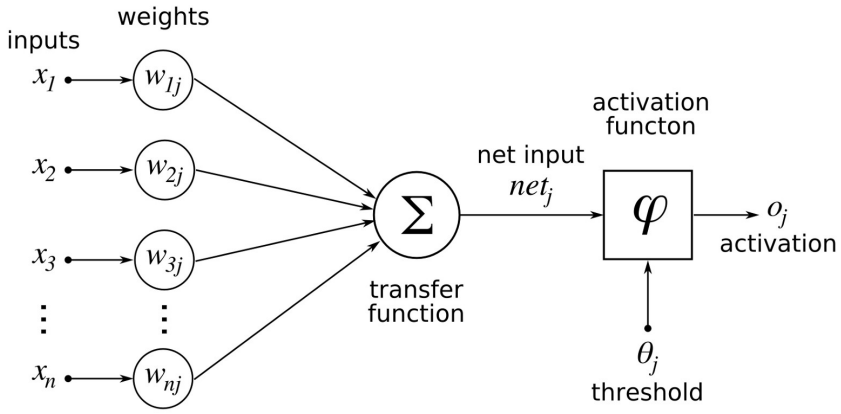
their creation of the Preceptron Model. Twenty years later, a technique to dynamically update parameter weights was developed (Rosenblatt 1961). From then, research in the field stalled until continuous activation functions were introduced in the 1980s (Rumelhart et al. 1986). While the concept of neural networks is over 75 years old, their analytic power is just now becoming apparent. According to Zhang (2000), four key advantages have helped neural networks become a suitable alternative to more conventional methods:

1. They are data-driven, self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model.
2. They are universal functional approximators in that neural networks can approximate any function with arbitrary accuracy.
3. They are nonlinear models, which makes them flexible in modeling real-world complex relationships.
4. They are able to estimate the posterior probabilities, which provides the basis for establishing classification rules and performing statistical analysis.

One key element of neural networks is the objective function. The objective function typically produces a loss value after each iteration from the input data. This value is the primary measure of the model's performance. By minimizing the objective function value through each iteration of the learning process, the weights of the model are adjusted dynamically through a process called back-propagation, also known as stochastic gradient descent when discussed in the more general optimization context. Back-propagation is the process in which a network's weights are updated by calculating their gradients. It is central in the application of neural networks. Refer to Goodfellow et al. (2016) for more detail about neural networks.

2.3.2 Variational Autoencoders

To understand a VAE it is important to first comprehend how an autoencoder is defined. An autoencoder is a neural network that is trained by unsupervised learning to produce reconstructions that are close to its original input (An and Cho 2015). In other words, an autoencoder is simply a process that seeks to produce outputs identical to its input. It uses unlabeled data for this task, which is important in our context of anomaly detection since



All inputs x_i are assigned a weight $w_{i,j}$ and summed denoted by Σ. The summed value net_j is activated by φ if it meets the threshold θ_j. This results in the output o_j, which continues to the next hidden layer in the process or terminates depending on the size of the neural network.

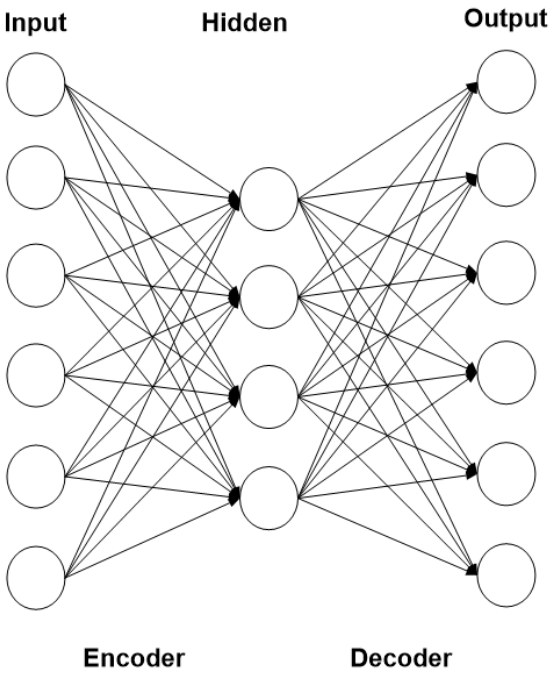
Figure 2.6. Components of a Neural Network–The Perceptron Model. Source: Goyal (2018).

normal and anomalous data points are rarely known in training sets.

An autoencoder's architecture can be simplified into two main parts: the encoder and decoder. Figure 2.7 shows a simple autoencoder model with its encoder and decoder labeled. Note, this is a fully connected autoencoder since each of the network's nodes are connected to all nodes in the adjacent layer. It is also worth noting that the encoder and decoder are both neural networks themselves.

For an autoencoder to work, the encoder portion receives the data input x and compresses it into a smaller dimension while feeding it forward to the next layer in the encoder. This can be accomplished for h layers, which are referred to as hidden layers. The final compression of the input occurs in the bottleneck of the autoencoder. The input's representation is now referred to as z, the latent representation of x. Next, the decoder takes the input's latent representation z and attempts to reconstruct the original input x by expanding it through the same number of hidden layers with identical number of corresponding neurons as the encoder.

Ideally, the output's result x' would be identical to the input x, and the autoencoder would learn a compressed (lower dimensional) version of the identity function. This is rarely, if



There is only one hidden layer in this fully-connected autoencoder.

Figure 2.7. An Autoencoder. Source: Galaxy Data Technologies (2019).

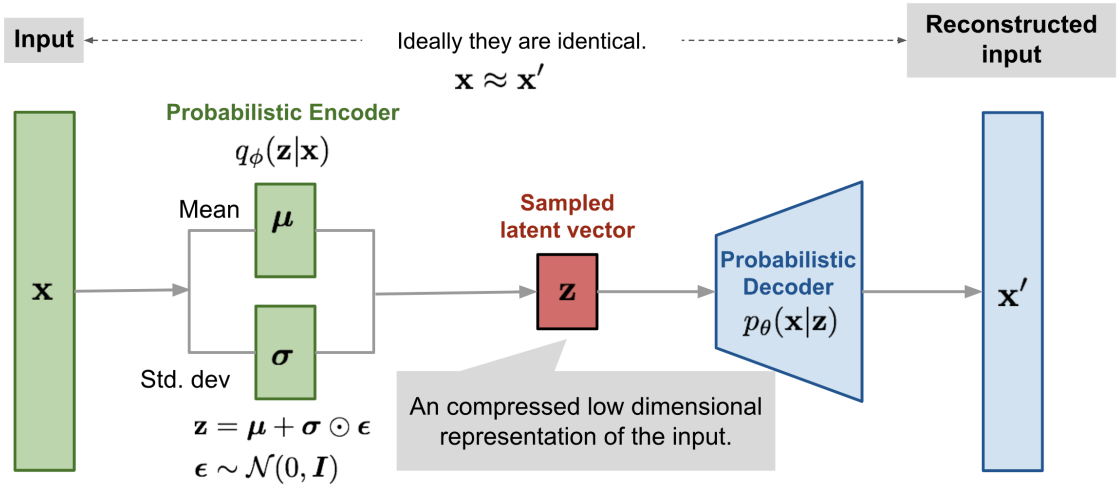
ever, the case and the subsequent difference between x and x' is called the reconstruction error. We exploit this quantity to detect anomalies.

In order to train the autoencoder to correctly reconstruct the inputs, an objective function is selected and minimized during the training process. This objective function represents the difference between the original input x and reconstructed input x'. There are dozens of different objective functions that could be used, and their selection depends on the autoencoder's data and objectives. One of the most common is the mean-square loss function. Another loss function is binary cross entropy. For a VAE (see Figure 2.8), the general expression of its objective function is the variational lowerbound of the marginal likelihood of the data, since the marginal likelihood is intractable (An and Cho 2015). Or more formally:

$$\log p_\theta(x^{(i)}) \quad = D_{KL}(q_\phi(z|x)||p_\theta(z)) + \mathcal{L}(\theta, \phi, x^{(i)}). \tag{2.1}$$

As An and Cho (2015) explain:

> $q_\phi(z|x)$ is the approximate posterior and $p_\theta(x|z)$ is the prior distribution of the latent variable z. The first term of the right-hand side of Equation 2.1 is the Kullback–Leibler (KL) divergence of the approximate posterior and the prior. The second term of the right-hand side of Equation 2.1 is the variational lowerbound on the marginal likelihood of the data point i.



A typical VAE with the encoder colored green, latent space colored red, and the decoder colored blue.

Figure 2.8. A VAE Neural Network. Source: Weng (2019).

What makes a VAE unique from a standard autoencoder is its bottleneck is a probabilistic distribution rather than a deterministic value. The posterior distribution, or encoder, $q_\phi(z|x)$ attempts to infer a distribution representing the input data. The final hidden layer of the encoder $q_\phi(z|x)$ produces two vectors, a vector of means and vector of standard deviations with dimensions m. The prior distribution, or decoder $p_\theta(x|z)$, now draws a random sample from a standard Gaussian distribution along with the vector of means and standard deviations

and feeds it through the decoder network $p_\theta(x|z)$ reconstructing the original input x. Again, like the autocoder, an objective function is selected and minimized to optimize the weight parameters ϕ, θ of the variational autoencoder. Doersch (2016) provides an excellent tutorial and is recommended for a deeper understanding of VAEs.

While VAEs seem much more complex than a standard autoencoder, they are extremely similar. For example, the lower-dimensional embedding learned by an autoencoder is simply treated as a mean vector of a multi-variate distribution. Additionally, the objective of the autoencoder seems more straightforward (e.g., the reconstruction error). However, for a Gaussian prior, we have that the objective of the VAE essentially reduces to a regularized version of the same reconstruction error.

A primary advantage of using a VAE over a standard autoencoder, and one we leverage in our research, is its robustness. A VAE is able to learn a robust interpretation of the latent representation since it randomly samples from a Gaussian distribution while decoding the vector of means and standard deviations. This sampling essentially acts as noise for the decoder to interpret, forcing it to learn many latent representations for a single data point while still producing the desired output. In a standard autoencoder, the decoder would always be presented with the same encoded latent representation z for each data point and essentially learn to overfit the data in time. Given our context of anomaly detection, a robust technique is desired due to the wide range of normal instances that may exist in data.

A second advantage of using a VAE rather than a standard autoencoder is its ability to generate unique outputs from sampling the latent space distribution. It is a generative model, meaning the decoder can create entirely new data from the latent distribution $p(z)$, if desired, and it will be as if it is from the same distribution as the actual data. This is particularly useful when using image data for training and testing. Images can be generated to show what the model is learning as representative of what the network thinks is "normal" data. We will explore this further in Chapter 4.

2.3.3 Issues with Standard VAE Approach

It is worth noting that standard VAEs and autoencoders have several shortcomings that can diminish their ability to detect anomalies as well as we would prefer. Both have significant representational power and can learn to reconstruct rare inputs or subsets of the data that are

abnormal. The default objective function of a VAE and autoencoder does not try to avoid this issue. In fact, it actually penalizes incorrect reconstructions of rare or anomalous data, which is counter to what we desire. After the VAE or autoencoder has been fit to the data, it can also be difficult to select an appropriate cut-off threshold, in terms of the reconstruction error, for determining anomalies. There is often no clear separation between "low" and "high" reconstruction errors. This is made even more difficult when the proportion of anomalies in the data are not known beforehand. Our approach remedies these problems and is discussed in detail in Chapter 3.

2.4 Related Work using Variational Autoencoders for Anomaly Detection

Although research in this field is relatively new, there are still several studies have been published leveraging the power of VAEs to detect anomalies. We briefly discuss three of these.

2.4.1 Segmentation in Brain MR Images

Baur et al. (2018) tested various autoencoder models to detect and delineate brain lesions from Multiple Sclerosis, tumors, and ischemias in magnetic resonance imaging of the human brain. They developed an approach they coined AnoVAEGAN which is a spatial VAE coupled with a Generative Adversarial Net (Goodfellow et al. 2016) and compared it to several different types of dense and spatial autoencoders and VAEs. AnoVAEGAN outperformed all other models they tested. It achieved an F_1 Score of just over 0.60 with a standard deviation of about 0.19. What is notable about their research is the spatial VAE performed nearly as well as the AnoVAEGAN model by achieving an F_1 Score of 0.59 with standard deviation 0.19. The dense VAE performed poorly, but this is not surprising since the task was detecting anomalies in images and Convolutional Neural Networks (Goodfellow et al. 2016) generally perform better in this situation.

2.4.2 Internet KPI Analysis

Xu et al. (2018) developed a model called *Donut* based on a VAE to detect anomalies in Key Performance Indicators (KPIs) for large Internet companies. KPIs are time-series data

metrics such as page clicks, number of reviews, etc. They focused their research on 18 business-related KPIs and conducted an experiment using three datasets, each with about 300,000 instances. The number of anomalies was between 6-7% in each dataset. They used F_1 Score as one of three metrics in determining the best performing model. Compared to a baseline VAE, their *Donut* model performed with an F_1 Score between 0.75-0.90 for all three datasets. Additionally, the smoothest dataset saw the highest score for *Donut* and lowest for the baseline VAE (around 0.4). For the least smooth dataset, both seem to perform well with an F_1 Score just below 0.8. Xu et al. (2018) concluded their *Donut* model, essentially an enhanced VAE, outperforms their non-VAE competitor *Opprentice*. Their result highlights the anomaly detection properties intrinsic to VAEs and their capacity to perform better when targeted modifications are applied.

2.4.3 Robotic Assisted Feeding

Researchers in the field of robotics recently tested the use VAEs to alert them of anomalous behavior in robots programmed to assist disabled individuals. A study conducted by Park et al. (2018) compared five different anomaly detection baseline methods to one they developed that utilizes a VAE coupled with a Long Short-Term Memory (LSTM) (Goodfellow et al. 2016) neural network. Their experimental setup was anchored around robotic assisted feeding. They recruited 24 participants and collected 1,556 feeding samples. Their LTSM-VAE model was pre-trained using 1,203 normal data instances and tested using 352 data instances (192 normal and 160 anomalous) which were all collected during the experiment. Seventeen features were included in the data, accounting for an equal number of robotic sensors. Their LSTM-VAE outperformed the five other detection methods. It achieved an area under the curve (AUC) value 0.0443 higher than the closest baseline model HMM-GP, 0.8564 and 0.8121, respectively. Again, like the previous two studies discussed, using a VAE or some altered version of a VAE for anomaly detection is a suitable choice and one worth exploring.

2.5 DoD Financial Anomalies

We show that VAEs are useful for detecting anomalies in a varying range of domains, from robotics to the medical community. Applying this technique to a relevant military application while simultaneously seeking to answer our research questions, we now begin

to shift our attention to Department of the Navy (DoN) budgeting and acquisitions for a single fiscal year. The DoN's annual budget provides a glimpse into the magnitude of total spending that occurs by the service each year and helps build context for our unlabeled dataset.

According to the Navy's Fiscal Year 2018 President's Budget summary, its baseline budget was $171.5 billion with $26.3 billion allocated for the Marine Corps. This amount includes expenses for operations and maintenance, personnel, procurement, etc. A large part of this, $124 billion, is paid to civilian entities across the country and around the globe for contracted services. While there is plenty of well-known oversight for the use of these funds, such as audits, history shows that *observations that do not conform to well-defined normal behavior*, or anomalies, still occur. We briefly provide three significant, non-conforming observations as it relates to relationships between the DoD writ large and its contractors.

Consider the now notorious Fat Leonard Scandal (Washington Post 2016). Leonard Glenn Francis was a businessman whose firm based in Singapore, Glenn Defense Marine Asia, managed to bribe dozens of officers with money, prostitutes, and lavish vacations for favorable government contracts. Beginning in the early 2000s until his arrest in 2013, Francis admitted to defrauding the DoN over $35 million while his company was awarded over $200 million over the same time.

Inchcape, a United Arab Emirates-owned DoN contractor, was the subject of fraud allegations that it routinely overbilled the Navy from 2005 to 2014 (Stars and Stripes 2018). At the time, the company carried $240 million in government contracts and supplied food and communications, in addition to other services, to ports across the globe. Inchcape did not admit to any wrong-doing in its settlement with the Justice Department in May 2018, but the company was fined $20 million as a result.

It is not just the DoN that has seen misuse of its treasure. The Air Force was the target of over-billing scheme by Northrup Grumman employees from 2010 to 2013 (U.S. Department of Justice 2018). Several employees routinely inflated the number of labor hours they billed the Air Force for services rendered on two awarded contracts. In total, over $5 million was stolen resulting in a $31.65 million settlement between the federal government and Northrup Grumman.

Given the DoD is not immune from illegal appropriations of its funds, whether from internal or external entities (and certainly not immune from clerical errors), we can apply the anomaly detection method we refine with our benchmark data to mine the DoN's open-source, contract award data for potential anomalies. We discuss these results in Chapter 4.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

In this chapter, we explain our approach and methodology using a VAE to detect anomalies. We first introduce our unique objective function and fitting technique for enhancing VAEs as an anomaly detector. Next, we explain in detail the architecture of the VAE and how it was tuned to process each dataset. Finally, we describe the data we used and how it was processed for use in our VAEs.

3.1 Percentile Objective Function

One of the primary drawbacks of autoencoder neural networks for anomaly detection is the ability of a single network to fit (reconstruct) multiple types of data at once. This ability is obviously advantageous in the context of learning highly flexible generative models or simply learning low-dimensional representations of complex data distributions. However, it is a drawback in the context of anomaly detection, since we want our model to learn to reconstruct only non-anomalous data inputs and to make large errors when reconstructing anomalous data points.

To fight this drawback, we introduce a novel objective function that is new to the autoencoder and VAE literature. Specifically, we utilize a concept called Conditional Value-at-Risk (CVaR), coming from the risk management literature, see Rockafellar and Uryasev (2002). While a full introduction to the CVaR concept, particularly in a risk management context, is beyond the scope of this thesis, it can be understood as a type of conditional expectation or tail-expectation. Specifically, given a continuously distributed real valued random variable L, CVaR at probability level $\alpha \in [0, 1]$ is given by,

$$\bar{q}_\alpha(L) = E[L|L > q_\alpha(L)], \tag{3.1}$$

where $q_\alpha(L) = \min\{t \mid P(L > t) \leq 1 - \alpha\}$ is the quantile of L. In other words, this simply gives the expected value of the right tail of the distribution of L, or the average of the largest $100 * (1 - \alpha)\%$ realizations of L. More general definitions and calculation formula exist for CVaR when L does not have a continuous distribution, see again Rockafellar and

Uryasev (2002). However, the important idea is simply that CVaR is the average of the largest $100 * (1 - \alpha)\%$ realizations of L.

Moving back to the context of VAEs, the typical objective function takes the form of the expected value of a random loss function $L(X, \theta)$ that depends on some random input X and neural network parameters θ. This loss is, for example, the reconstruction error of a random input X. Thus, the objective function seeks to solve:

$$\min_{\theta} E[L(X, \theta)].$$

This is then optimized via Stochastic Gradient Descent (SGD) in batch-wise fashion, where each iteration utilizes an empirical estimate of this expectation for gradient calculation given by

$$\frac{1}{|B|} \sum_{i=1}^{|B|} L(X_i, \theta),$$

where $B = \{X_1, \ldots, X_n\}$ is a batch of training examples.

3.1.1 A New Objective

As stated before, the VAE is powerful enough to fit even anomalous data distributions, and we would like to encourage the VAE to fit only non-anomalous data inputs while making large errors when reconstructing anomalous data. We accomplish this by using an objective of the following form, where $\alpha \in (0, 1)$ is a parameter chosen by the user to indicate what proportion of largest training losses should be ignored. Note, when $\alpha = 1.0$ we have the standard objective function.

$$\min_{\theta} E[L(X, \theta)] - (1 - \alpha)\bar{q}_{\alpha}(L(X, \theta)). \tag{3.2}$$

This new objective function effectively reduces to the following empirical variation over a sample batch $B = \{X_1, \ldots, X_n\}$, where we now denote loss realizations as $\ell_i := L(X_i, \theta)$ and denote an ordered permutation of samples as $\ell^{(1)} \leq \cdots \leq \ell^{(n)}$ and also let $k = \lfloor n(1 - \alpha) \rfloor$.

$$\frac{1}{n-k}\sum_{i=1}^{n-k}\ell^{(i)} \tag{3.3}$$

Equation (3.3) is the function used to calculate gradients for SGD for each batch. This new objective function ignores the k-largest losses from each training batch and only propagates errors for the smallest $n-k$ losses.

Intuitively, this new objective function helps to encourage the VAE to reconstruct non-anomalous examples which are assumed to produce smaller losses during training. Said another way, if the network receives an anomalous input and makes a large reconstruction error, a typical VAE objective function will propagate that error and encourage the VAE to learn how to reconstruct it. This is, of course, the opposite of what we want. Therefore, we construct an objective function that ignores these large errors during training.

In terms of the distribution of reconstruction errors which we use after training to detect anomalies, our objective function is meant to encourage the VAE to have a long right-tailed distribution so that it will be easier to identify the loss threshold that separates anomalies and non-anomalies.

3.1.2 Choosing α for the New Objective

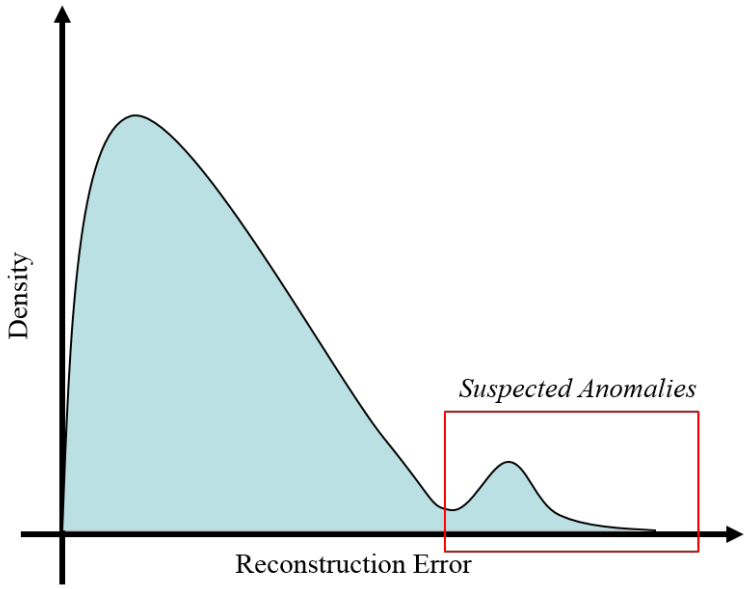
Our new proposed objective function introduces one new hyperparameter that needs to be selected; the parameter $\alpha \in [0, 1]$. This term determines what proportion of largest-loss training examples (per batch) will be ignored during training. Intuitively, the correct choice would be with $1-\alpha$ equal to the true proportion of anomalies present within the training set. Assuming training batches are sufficiently shuffled and randomly sampled, and assuming that indeed anomalous training points will produce one of the $k = n(1-\alpha)$ largest losses during training iterations, our objective function will ignore the reconstruction errors on anomalous points (approximately) throughout training if $1-\alpha$ is selected to equal the true proportion of anomalies in the training set.

If this information is known, we are then in a partially supervised learning setting. We show in the following sections that, indeed, if $1-\alpha$ is set to equal to the known proportion of anomalous data points in the training set, the learned VAE outperforms a traditionally

trained VAE.

The supervised learning setting, however, is not the setting we want to work in. We desire a method that works in the fully unsupervised setting, where no information about the training set (i.e., labels, proportion of anomalies) is known. In this setting, we need an automated procedure for selecting the best level α.

To gather intuition about how to distinguish between VAEs fit with "good" and "bad" choices of α, we can look at the resulting distributions of reconstruction errors when we know the true proportion of anomalous training examples. Indeed, we see that when α is chosen to match the true proportion of anomalies, the distribution of reconstruction errors begins to appear bi-modal with a gap between well-reconstructed examples and poorly-reconstructed examples (see Figure 3.1). This motivates us to use a clustering approach on the reconstruction errors to identify error distributions which have this natural clustering of large and small reconstruction errors.

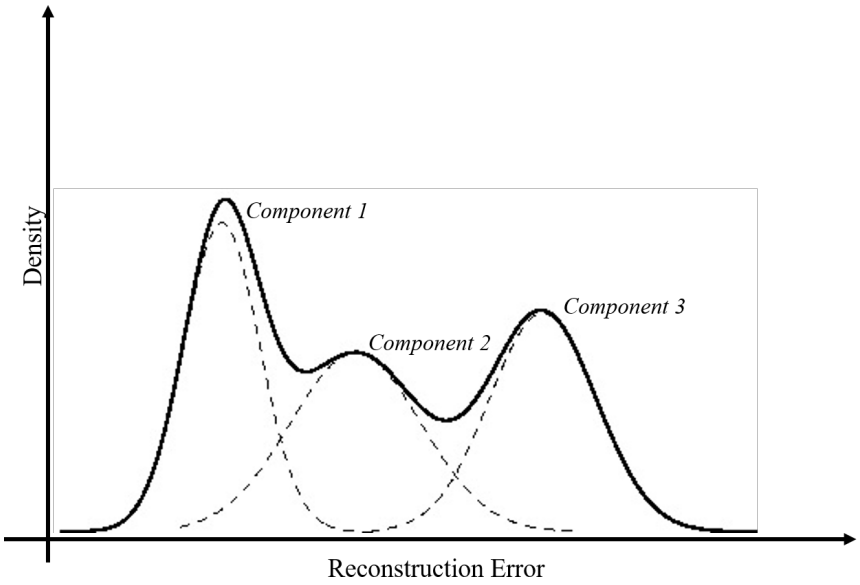


This bi-modal distribution of losses is the ideal case for detecting anomalies with the new objective function. The red box highlights the suspected region of anomalous reconstruction errors.

Figure 3.1. Desired Distribution of Reconstruction Errors

We utilize a Gaussian mixture model (GMM) to cluster the set of 1-D reconstruction errors

produced by our trained VAEs. A GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities (Reynolds 2015). In other words, it is a combination of multiple underlying Gaussian distributions taken together to form one continuous density function (see Figure 3.2). The reconstruction errors from our VAEs form the underlying Gaussian distributions. We choose three clusters to represent normal, high-loss normal, and anomalous reconstruction errors. Ideally, two clusters (normal and anomalous) would be sufficient, but this is not practical due to high-loss normal data we often encounter. In other words, normal data that has a larger than average reconstruction error must be accounted for in some way and is handled by fitting three clusters in our GMMs.



This GMM has 3 components represented by the dotted lines. The solid black line is the sum of the components and the GMM. For our purposes, components 1, 2, and 3 represent the reconstruction errors for normal, high-loss normal, and anomalous points, respectively.

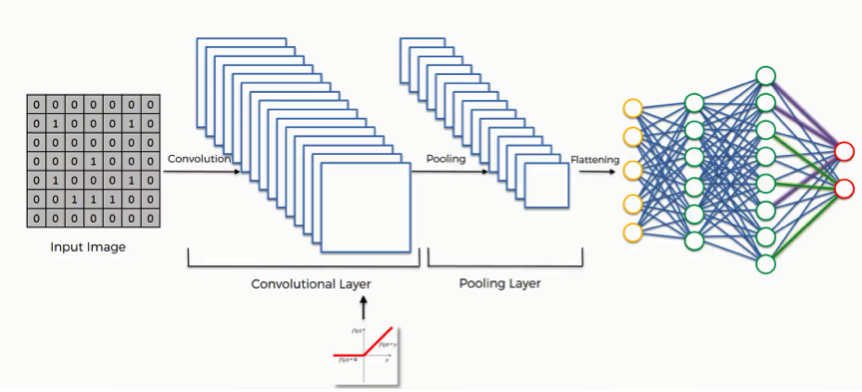
Figure 3.2. GMM with 3 Components (Clusters)

We train our VAE with multiple levels of α. Then, for each VAE, we fit a GMM on its errors. We then use the fit models to find which VAE (i.e., which choice of α) produced the most distinct clusters of reconstruction errors. For each GMM, we select the cluster with the highest mean. Focusing on only the third, or rightmost, components we compare their

attributes among all fit GMMs and finally select the GMM (and corresponding α from the VAE) that has the smallest weight w, where $w \in [0, 1]$, smallest variance σ, and largest mean μ. Put another way, of all distributions of reconstruction errors, we desire the one whose rightmost cluster has the highest mean and has the smallest weight and variance. Specific results using this approach are found in Chapter 4.2. Overall, we find that this approach, in combination with our proposed objective function, is highly effective and produces a suitable unsupervised approach for anomaly detection using VAE's when no labeled training data is given and the proportion of anomalies in the dataset is unknown.

3.2 Model Architecture

For each of the three non-image benchmark and USASpending datasets, we use fully-connected, or dense, VAEs, meaning each node is connected to every node in the previous and subsequent layers. For the Fashion-MNIST we utilize a Convolutional Neural Network (CNN) VAE (see Figure 3.3). CNNs are often used for processing image data, and their primary advantage over an ordinary, dense VAE is they maintain pixel location information and have shared weights among layers which leads to better results. We utilized the PyTorch (Paszke et al. 2017) library in Python for building and training all VAEs.



A basic CNN with Convolutional and Pooling Layers. Notice the input information is flattened and fed forward into a Fully Connected network.

Figure 3.3. Basic Convolutional Neural Network. Source: SuperDataScience Team (2018).

There are many parameters in a VAE that can be tuned, from learning rates to batch sizes. We select their values from either the default settings in the PyTorch software or from common

best practices. Our objective in designing the model's architecture is not to engineer the best VAE for identifying anomalies in the benchmark data but rather to use consistent parameter values that may provide clues for building a generalized VAE detector. When discussing hidden units specifically, Goodfellow et al. (2016) states:

> Predicting in advance which will work best is usually impossible. The design process consists of trial and error, intuiting that a kind of hidden unit may work well, and then training a network with that kind of hidden unit and evaluating its performance on a validation set.

His insights carry weight for the entire design process and are heeded here. When appropriate, parameter settings are kept constant for benchmark datasets unless otherwise specified. Table 3.1 provides their values.

Table 3.1. Summary of Unchanging Parameters

Parameter	Value
Epochs	5
Learning Rate	1e-3
Weight Decay	1e-5
DropOut Layer 1	0.2

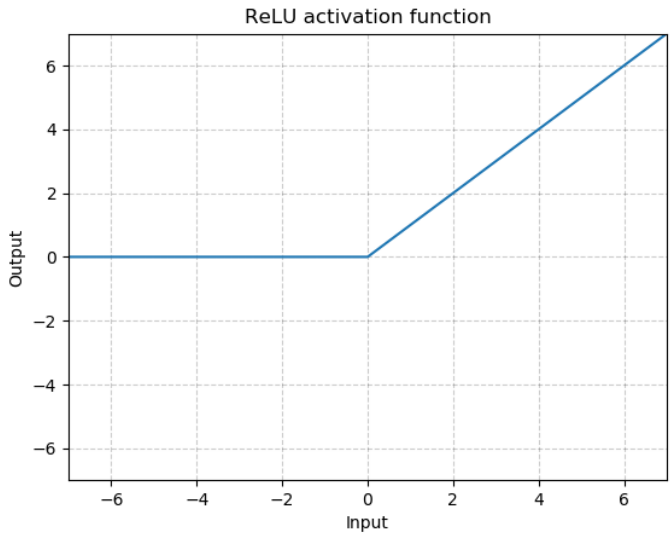
We initialize our VAEs with three hidden layers (see Table 3.2), including the bottleneck layer, and our choice of optimizer is Adam (Kingma and Ba 2015). We utilize a dropout layer (Srivastava et al. 2014) in the encoder $q_\phi(z|x)$ and as well as weight decay (Krogh and Hertz 1992) to regularize the network. Dropout is a technique that randomly removes nodes and their connections within a selected layer during training. This helps the network avoid over-fitting the training data, which is useful in our case since we do not want our VAE to learn to reconstruct the anomalous records in our data. Weight decay helps avoid overfitting as well. It is a penalty term applied to the weights of the network after each update. It causes them to exponentially decay to zero preventing them from growing arbitrarily large.

Our choice of activation function is Rectified Linear Units (ReLUs) and is shown in Figure 3.4. It is used to activate each neuron in all layers of the VAE with the exception of the final layer where we use a Sigmoid activation function. The final output is restricted from $[0, 1]$

Table 3.2. Initial Node Counts in Each Dense Layer of VAE

Layer	KDDCup99	Statlog Shuttle	CoverType	Fashion-MNIST	DoN Contracts
Input	48	9	54	784	997
Hidden 1	24	7	24	200	512
Hidden 2	6	4	6	20	48
Hidden 3	24	7	24	200	512
Output	48	9	54	784	997

in order to use Binary Cross Entropy as the reconstruction error portion of the objective function.



The ReLU activation function sets a negative input to 0 and is linear otherwise.

Figure 3.4. ReLU Activation Function. Source: Paszke et al. (2017).

Batch size is determined by dividing each dataset into 500 subsets which provides each epoch 500 iterations. Training batches are shuffled while test batches are not. Shuffling training batches helps regularize the VAE and not shuffling test batches provides consistent results that simplifies comparison.

Another distinction between training and testing with the VAE is the decoder does not sample from a Gaussian distribution during testing but simply uses the mean and standard

deviation vectors produced by the encoder. This eliminates variability in the final output and provides a measure of reproducibility in the network. Training is still conducted using random sampling from the latent space to the decoder.

As previously mentioned, the Fashion-MNIST data is trained using a CNN VAE. The key difference in architecture from the dense VAE is by the addition of two Convolutional and two Pooling Layers in the encoder preceding the dense hidden layers. We set the kernel size for the Convolutional and Pooling layers to 5 and 2, respectively.

3.3 Data and Pre-processing

In order to develop and tune a VAE for unsupervised anomaly detection, we first have to utilize labeled datasets so that we are able to evaluate its performance against ground truth labels. We use four benchmark datasets accessed from the University of California–Irvine Machine Learning Repository. This provides a feedback mechanism during training and enables exploration into how the VAE handles different data sizes and types. The fifth dataset we use is accessed from USASpending.gov and is unlabeled.

3.3.1 KDDCup99

Our primary benchmark dataset was used in the Knowledge Discovery and Data Mining Competition in 1999, and we will refer to it as KDDCup99. The original task of the competition using this data was "to build a network intrusion detector, a predictive model capable of distinguishing between *bad* connections, called intrusions or attacks, and *good* normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment" (Hettich and Bay 1999). We subset it to fit our requirements, similar to An and Cho (2015) and Williams et al. (2002). The original data contains 4,898,431 records, 41 features, and 1 label vector. Each record is a network login or an attempted network login. We only use records with positive (connected) network logins since we are attempting to determine if the login is normal or anomalous. This decreases the data to 703,066 records. Furthermore, we subset it by a feature called `service` which consists of network connection types (e.g., http, ftp, smtp). Using only the `http` service type, the data is further reduced to 567,497 records, of which 2,211 are intrusions. Table 3.3 shows the different classes along with

their quantities and proportion in the data. DOS, Probe, R2L, and U2R are all intrusions and considered to be anomalies.

Table 3.3. KDDCup99 Data

Class	Records	Percent
Normal	565,286	99.61%
DOS	2,203	0.39%
probe	4	~ 0%
R2L	0	0%
U2R	4	~ 0%
	567,497	100%

Next, all categorical features with more than two values are processed via one-hot encoding. This allows the VAE to interpret them as binary representations. For example, if one categorical feature has three possible values (e.g., sit, walk, run) then one-hot encoding will create three new features `sit`, `walk`, and `run`. If that record's original categorical variable value is `walk`, then the new vector will be (0,1,0). One-hot encoding is done frequently for categorical variables in this thesis and is common practice. After one-hot encoding `flag` and removing `protocol_type` and `service`, the final number of features in the data 48. Finally, all features are normalized to have mean zero and variance one.

3.3.2 Statlog Shuttle

Our second benchmark dataset is named Statlog Shuttle and can be found on the University of California–Irvine Machine Learning Repository (Dua and Graff 2017). It consists of 58,000 records, 9 features, and 1 label vector. There are 7 numeric classes in the label vector each corresponding to the following: 1 Rad Flow, 2 Fpv Close, 3 Fpv Open, 4 High, 5 Bypass, 6 Bpv Close, and 7 Bpv Open. Following Tan et al. (2011), we label classes 2, 3, 5, 6, and 7 as the anomaly classes due to their low frequency in the data. Class 4 is removed, and Class 1 is labeled normal since it is the majority class. Table 3.4 displays the number of records and percentages of each class within the data. All 9 features are numeric, so one-hot encoding is not required and the final dimension of the data remains at 9. Finally, similar to the KDDCup99 data, the Statlog Shuttle data is normalized to have mean zero and variance one.

Table 3.4. Statlog Shuttle Data

Class	Records	Percent
Rad flow	45,586	92.86%
Fpv Close	50	0.10%
Fpv Open	171	0.35%
Bypass	3267	6.65%
Bpv Close	10	0.02%
Bpv Open	13	0.03%
	49,097	100%

3.3.3 Forest Covertype

Our third benchmark dataset is Forest Covertype, again found at the University of California–Irvine Machine Learning Repository. Blackard (1998) created a database of seven different forest cover types in northern Colorado. His description of the data is thorough and reproduced here:

> [Predict] forest cover type from cartographic variables only (no remotely sensed data). The actual forest cover type for a given observation (30 x 30 meter cell) was determined from U.S. Forest Service (USFS) Region 2 Resource Information System (RIS) data. Independent variables were derived from data originally obtained from U.S. Geological Survey (USGS) and USFS data. Data is in raw form (not scaled) and contains binary (0 or 1) columns of data for qualitative independent variables (wilderness areas and soil types).
>
> This study area includes four wilderness areas located in the Roosevelt National Forest of northern Colorado. These areas represent forests with minimal human-caused disturbances, so that existing forest cover types are more a result of ecological processes rather than forest management practices.

The data contains 581,012 records, 54 features, and 1 label vector. Of the 54 features, 44 are binary and the remaining 10 are continuous. For our purposes, we subset the data similarly to Tan et al. (2011), but keep all features. There are 7 classes of covertypes: 1 Spruce/Fir, 2 Lodgepole Pine, 3 Ponderosa Pine, 4 Cottonwood/Willow, 5 Aspen, 6 Douglas-fir, and 7 Krummholz. The Lodgepole Pine and Cottonwood/Willow classes are

normal and anomalous, respectively, in our setup and all others are removed. Table 3.5 provides the number of records and proportions of each in our dataset. The dimensionality of the data remains 54. Finally, the data is normalized to have mean zero and variance one.

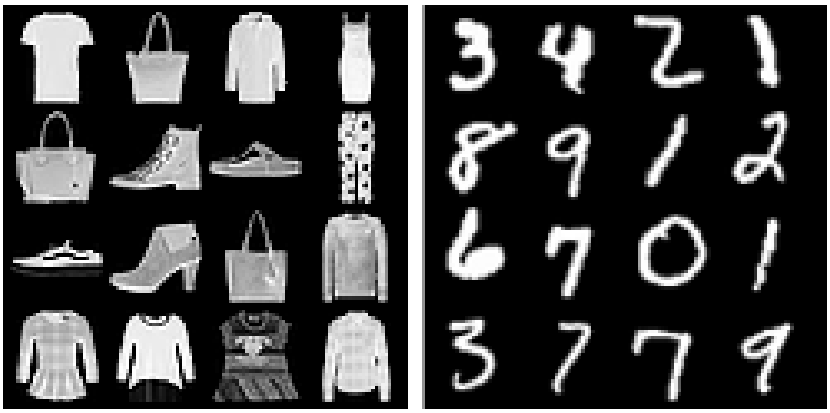
Table 3.5. Forest Covertype Data

Class	Records	Percent
Lodgepole Pine	283,301	99.04%
Cottonwood/Willow	2,747	0.96%
	286,048	100%

3.3.4 Fashion-MNIST

Our final benchmark dataset is created from the Fashion-MNIST (Xiao et al. 2017) dataset and the original handwritten digit MNIST (Lecun et al. 1998). Both consist of a 60,000 28x28 pixel gray-scale image training set and a 10,000 28x28 pixel gray-scale image test set. We discard the test sets for both and only work with the training sets.

Fashion-MNIST (Figure 3.5) has 10 classes of various articles of clothing and accessories: 0 T-shirt/top, 1 Trouser, 2 Pullover, 3 Dress, 4 Coat, 5 Sandal, 6 Shirt, 7 Sneaker, 8 Bag, and 9 Ankle boot. Similarly, MNIST has 10 classes of handwritten numbers, all digits from 0-9. We take the full set of 60,000 MNIST digits and merge them with a sample of 1,203 Fashion-MNIST records to create our dataset. The goal is for the VAE to identify the Fashion-MNIST records, our anomalies is this case, among the 60,000 MNIST handwritten digits. What makes this data unique from the other three benchmarks is it provides a visualization of the VAE's efforts to reconstruct anomalous points.



(a) Fashion MNIST (b) MNIST

Both the handwritten MNIST and a portion of the Fashion-MNIST are merged together to form the only image benchmark used for anomaly detection in this thesis.

Figure 3.5. Fashion-MNIST Data

3.3.5 DoN Contract Award Data

Our final dataset, DoN contract award data from FY14-FY18, is the only unlabled one we use in this research. It comes from an open-source, public database available on a U.S. government website (USASpending 2019). The organization hosting the database states its mission and background is:

> to show the American public what the federal government spends every year and how it spends the money. You can follow the money from the Congressional appropriations to the federal agencies and down to local communities and businesses.
>
> The Federal Funding Accountability and Transparency Act of 2006 (FFATA) was signed into law on September 26, 2006. The legislation required that federal contract, grant, loan, and other financial assistance awards of more than $25,000 be displayed on a publicly accessible and searchable website to give the American public access to information on how their tax dollars are being spent. In 2008, FFATA was amended by the Government Funding Transparency Act, which required prime recipients to report details on their first-tier sub-recipients

for awards made as of October 1, 2010.

The transparency efforts of FFATA were expanded with the enactment of the Digital Accountability and Transparency Act (DATA Act) Pub. L. 113-101 on May 9, 2014. The purpose of the DATA Act, as directed by Congress, is to:

- Expand FFATA by disclosing direct agency expenditures and linking federal contract, loan, and grant spending information to federal agency programs.
- Establish government-wide data standards for financial data and provide consistent, reliable, and searchable data that is displayed accurately.
- Simplify reporting, streamline reporting requirements, and reduce compliance costs, while improving transparency.
- Improve the quality of data submitted to USAspending.gov by holding agencies accountable.

Choosing to work with this data not only enables us to rigorously test our methodology but also provides military relevance to our work. As shown in Chapter 2, anomalies can present themselves in various domains and in various ways. The U.S. government, and in particular DoN financial activities, is not immune from erroneous, illicit, or negligent behavior. While it is not within our scope to identify *why* certain data instances are anomalous, we nonetheless make the distinction between normal and anomalous records in the data.

Due to the wide range of services and products that are awarded DoN contracts, we only analyze a portion of the entire dataset. We subset it based on the North American Industry Classification System (NAICS) codes and use sector 54 (Professional, Scientific, and Technical Services). This sector encompasses the majority of all DoN contracts awarded in the last five fiscal years. Processing the data in this fashion will make the VAE more likely to detect anomalies since contract awards will be similar to each other. This prevents the VAE from trying to learn "normal" awards across many NAICS categories (e.g., 23-Construction and 61-Education Services) . The same approach is done for the `service` feature in the KDDCup99 data.

In total, our DoN contract award dataset consist of 292,853 records (see Table 3.6). There are 6 continuous features (dollar and obligation amounts) and 126 categorical, which increases to a total of 997 after one-hot encoding.

Table 3.6. DoN Contract Awards for NACIS Sector 54

Fiscal Year	Records
2018	55,755
2017	57,741
2016	57,205
2015	59,592
2014	62,560

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results and Analysis

This chapter consists of three sections detailing our results and their analysis. We begin by discussing how the VAE performed with the benchmark datasets when the proportion of anomalies within each dataset is known. We show that our proposed VAE outperforms the traditional VAE, particularly when α is selected proportional to the known fraction of anomalies in the dataset, as discussed in Chapter 3.1. Next, ignoring labels and assuming the proportion of anomalies is unknown, we use the GMM methodology from Chapter 3.1 to choose the proper choice of α. Furthermore, we find that this outperforms the standard VAE approach and effectively separates anomalies from non-anomalies in our datasets. It additionally chooses the correct choice of α, matching the proportion of known anomalies when considering the labels after-the-fact. Finally, the DoN contract award data is analyzed using the same methods and conclusions made with respect to identifying anomalous records.

4.1 Performance with Benchmark Data and α Known

To test the effectiveness of the proposed objective function and to see whether the intuition laid out in Chapter 3.1 for choosing α is indeed correct, we begin by assuming we know the true proportion of anomalies and measure model performance for multiple values of α. We see that, indeed, model performance is optimal when α is set proportional to the known fraction of anomalies present in the dataset. In the next section, 4.2, we eliminate this assumption and utilize our GMM procedure for choosing α. We take advantage of knowing the true proportion of anomalies by using the benchmark labels in our analysis.

Scores for each model are determined by calculating the model's *precision*, *recall*, and F_1 *Score*, where

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$F_1Score = 2 * \frac{Precision * Recall}{Precision + Recall}.$$

We use F_1 Score as our primary measure of performance and test to see which α gives the best result.

In general, we discover our new objective function outperforms the non-altered baseline objective function. In other words, $\alpha < 1.0$ detects more anomalies in data than when $\alpha = 1.0$, with $\alpha = 1.0$ being equivalent to the non-altered baseline VAE objective function. Also, the optimum value for α is in the vicinity of the true proportion of anomalies within the data. This makes sense intuitively, since the VAE is ignoring the anomalies when training the model's ϕ, θ weight parameters. We explore the results of the VAE's performance by analyzing the results of our benchmark data. Each benchmark provides a useful insight we carry forward in our analysis and apply to our unlabeled data (DoN Contract Awards) in Chapter 4.3.

4.1.1 KDDCup99

Using the methodology described in Chapter 3, 10 values of α are compared for their performance detecting anomalies. The number of anomalies in the data is 2,211 or roughly 0.39%. Table 4.1 shows each α's corresponding F_1 Scores. The highest performing VAE is the one whose objective function uses $\alpha = 0.995$. Notice this value is nearly the same as the proportion of normal data within the dataset.

Inspecting the distribution of losses for each of the VAEs, we see the effects of the α hyperparameter increasing the anomalies' reconstruction errors. When comparing $\alpha = 1.00$ (no change to the objective function) to any other value of α, the distribution of losses associated with the anomalous classes increase. The separation in losses between the normal and anomalous classes indicates the VAE is learning exclusively from the data's normal records and ignoring the anomalies in training. Figure 4.1 displays this for the optimum value of α.

Table 4.1. KDDCup99 F_1 Scores at 99.61th Percentile of Losses

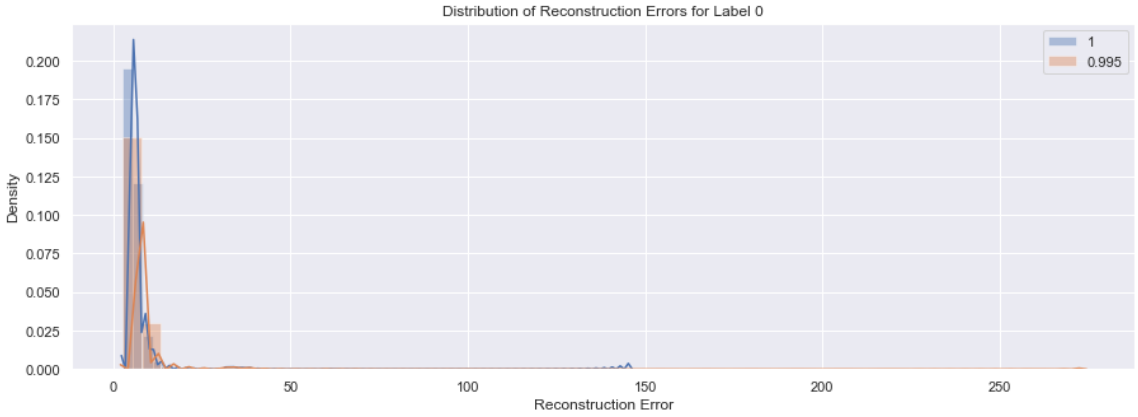
α	F_1 Score
1.00	0.4090
0.995	**0.9695**
0.99	0.8963
0.98	0.7837
0.97	0.6638
0.96	0.6929
0.95	0.6807
0.94	0.6255
0.93	0.6571
0.92	0.8443

4.1.2 Statlog Shuttle

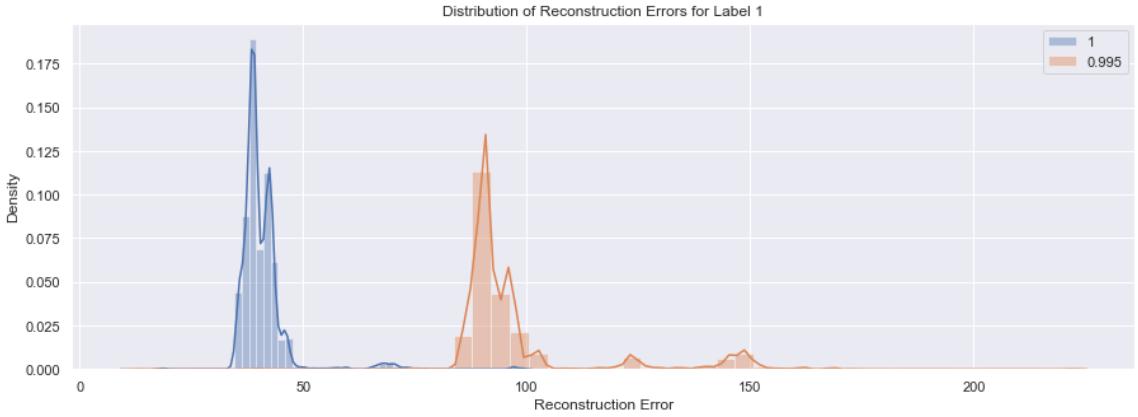
We find similar, but not identical, results with the Statlog Shuttle dataset (see Table 4.2). There are 49,097 records of which roughly 7% are anomalies. Inference from the KDD-Cup99 results suggest the best-performing VAE is the one whose $\alpha = 0.93$. This is roughly the true proportion of normal records within this dataset. While this is not exactly the case, we still get an α value relatively close to 0.93. We find the highest rate of detection when $\alpha = 0.90$. Moreover, the absolute differences in detection between the two ($\alpha = 0.93$ and $\alpha = 0.90$) is rather small when comparing each VAE's F_1 Score.

Table 4.2. Statlog Shuttle F_1 Scores at 92.85th Percentile of Losses

α	F_1 Score
1.00	0.7602
0.98	0.5990
0.96	0.7603
0.94	0.5480
0.93	0.7975
0.92	0.8015
0.91	0.7106
0.90	**0.9581**
0.89	0.9487
0.88	0.8610



(a) Normal Records Reconstruction Errors

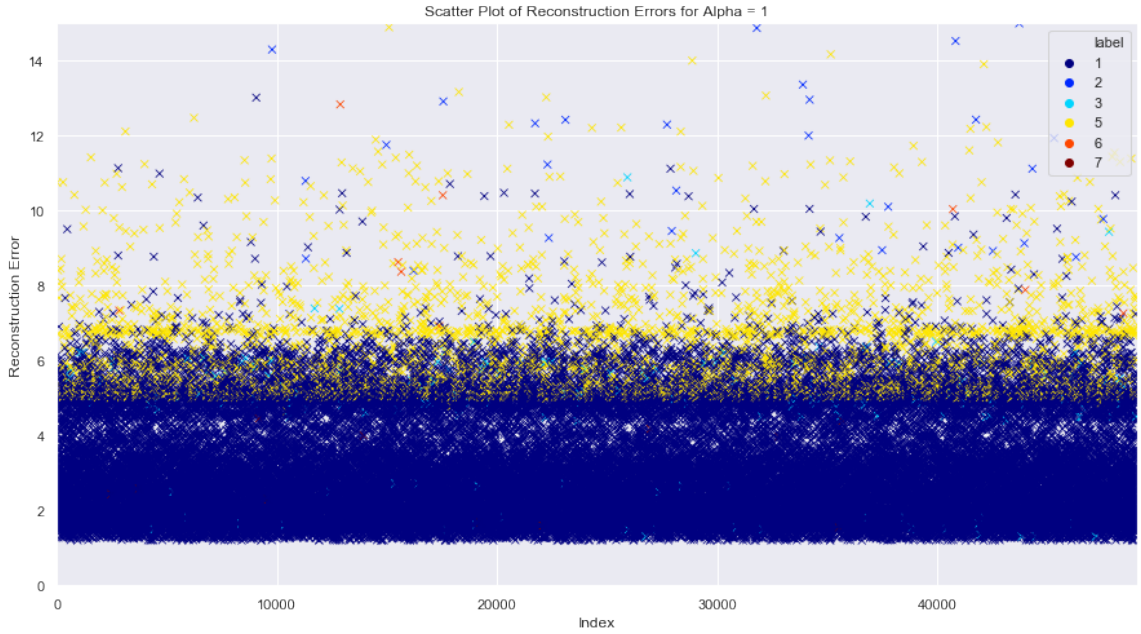


(b) Anomalous Records Reconstruction Errors

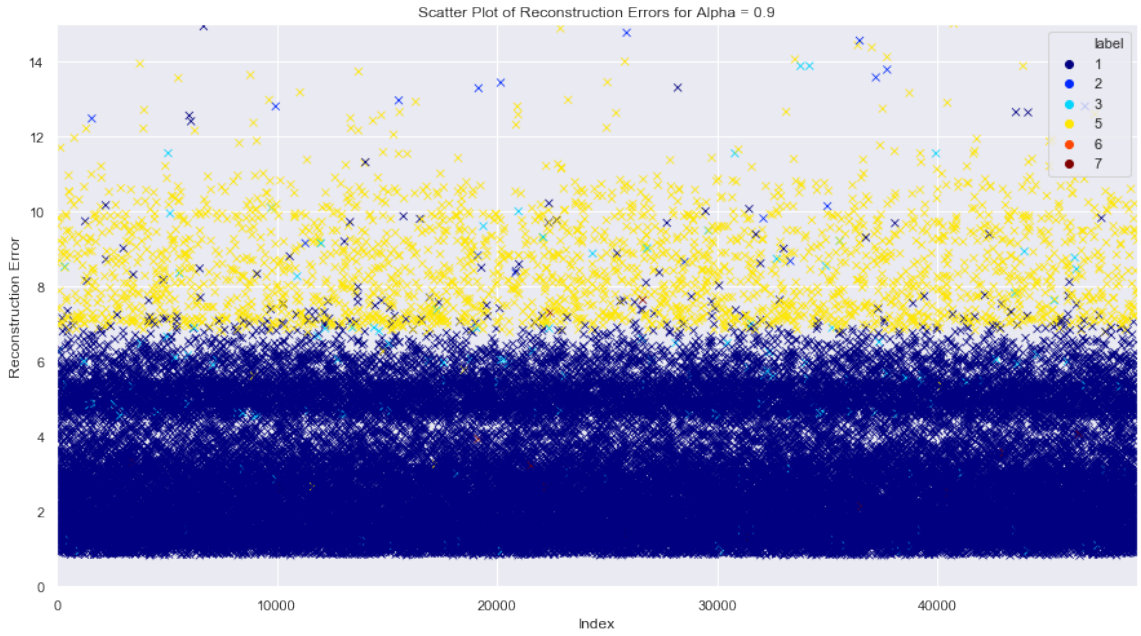
This figure illustrate the differences in reconstruction errors between (a) normal and (b) anomalous records in the KDDCup99 benchmark dataset. Notice the anomalous losses (b) increase for $\alpha = 0.995$ (orange).

Figure 4.1. KDDCup99 Distribution of Losses

The scatter plots of the reconstruction errors between the values of α provides an excellent visualization of the growing gap between normal and anomalous records. As the value of α changes and nears its optima for detecting anomalies, the gap expands and seems to stretch or rip the reconstruction errors into distinct chunks (see Figure 4.2). The effect of our objective function with the new hyperparameter α is clearly impacting our results in a favorable way, and choosing α is plausible by closely examining a scatter plot. While this method is admittedly subjective and could differ based on various interpretations, we explore our metric-based approach for α selection in Section 4.2.



(a) $\alpha = 1.0$



(b) $\alpha = 0.90$

This figure illustrates the differences in reconstruction error between normal (blue) and anomalous (yellow) points. Notice the anomalous losses seem to separate from the normal data when $\alpha = 0.90$, subfigure(b).

Figure 4.2. Statlog Shuttle Scatter Plots of Reconstruction Errors

4.1.3 CoverType

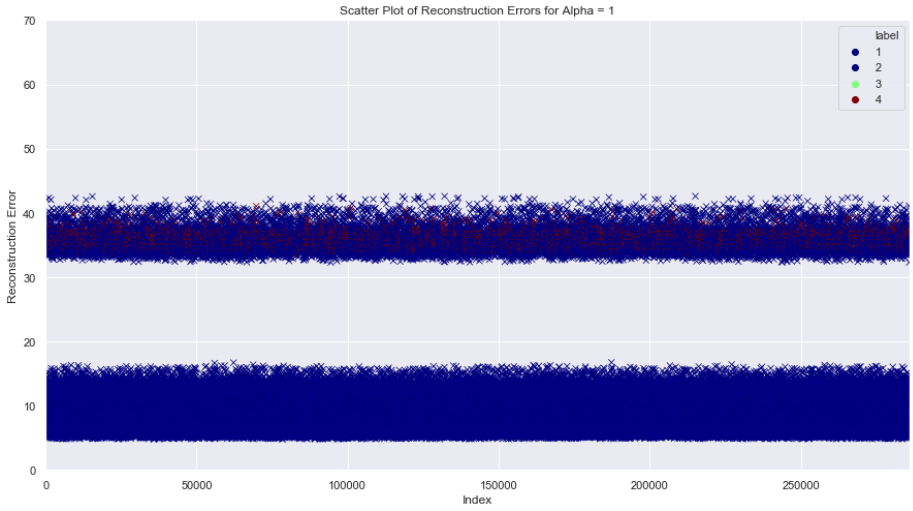
Our VAEs had the least amount of success of all the benchmark data detecting anomalies in the CoverType dataset. It consists of 286,048 records of which 2,747 are anomalies (0.96%), a slightly higher proportion than the KDDCup99 data. We only achieved a maximum F_1 Score of about 0.65 while the other benchmark data consistently produced results above 0.90 for the optimum α. Although anomaly detection was relatively low with this data, we still saw improved detection applying our new objective function (see Table 4.3). When $\alpha = 1.00$ and $\alpha = 0.97$ the F_1 Scores are 0.0662 and 0.6459, respectively. Figure 4.3 shows some of the difficulty achieving a high F_1 Score since there is significant overlap between normal and anomaly reconstructions, one of the challenges we discussed in Chapter 2. While the new objective function creates a clear separation between the majority of the normal and anomalous records, it cannot differentiate highest-loss normal records from the true anomalies well.

Table 4.3. CoverType F_1 Scores at 99.04th Percentile of Losses

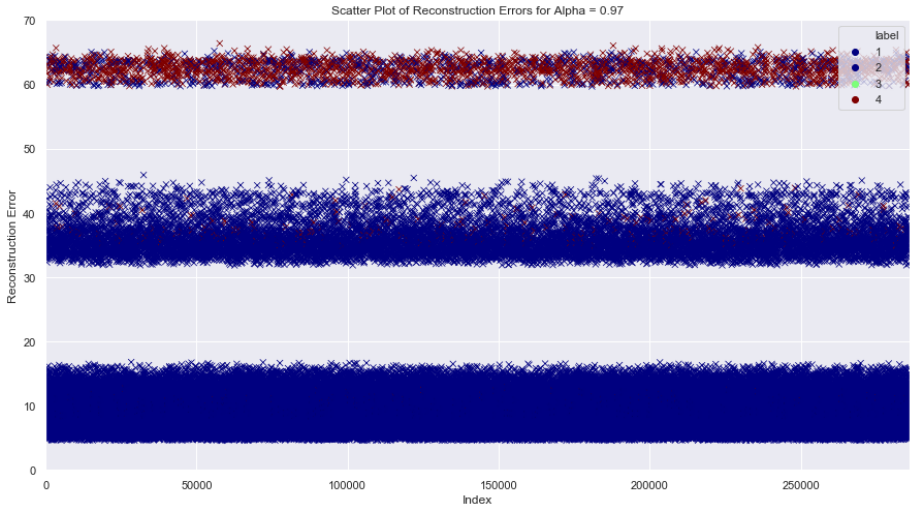
α	F_1 Score
1.00	0.0662
0.99	0.0962
0.98	0.1668
0.97	**0.6459**
0.96	0.6168
0.95	0.6376
0.94	0.6366
0.93	0.3474
0.92	0.6385
0.91	0.6286

Taking another look at the reconstruction losses, we examine the density distributions for the same α values (1.0 and 0.97). Figure 4.4(a) is a very close match to our desired distribution of reconstruction losses previously shown in Figure 3.1. It exhibits the desired bi-modal curve with clear separation between the two modes; however, this is deceptive since the anomalies are masked within normal records in this case (see Figure 4.3(a)). A third group, or cluster, emerges in Figure 4.4(b). The new objective function detects the majority of anomalies contained within this group and penalizes them more than the baseline VAE

($\alpha = 1.0$), thus creating a small, third group. This insight proved particularly useful when deciding to use three components for our GMM fitting, as introduced in Chapter 3.1.



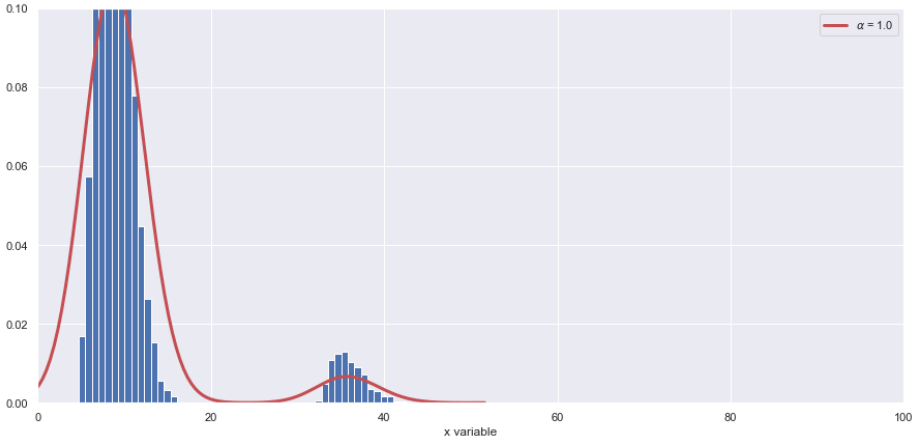
(a) $\alpha = 1.0$



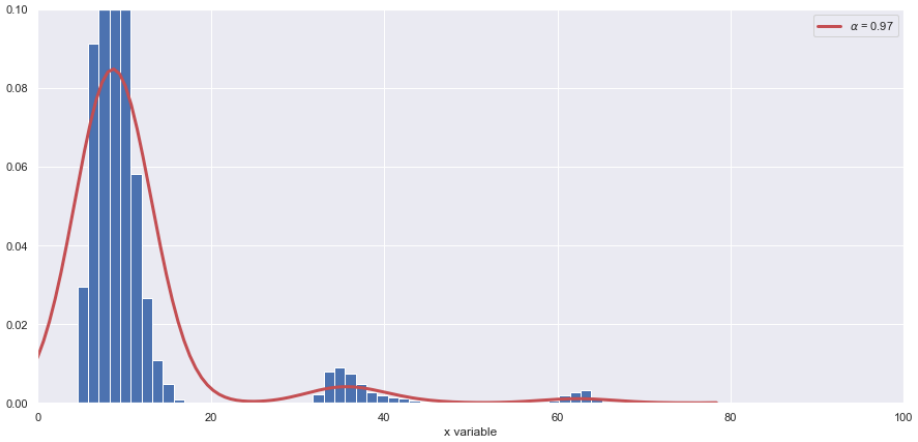
(b) $\alpha = 0.97$

This figure illustrate the differences in loss values between (a) normal and (b) anomalous records in the CoverType benchmark dataset. Notice the anomalous losses (b) seem to create a new cluster for $\alpha = 0.97$.

Figure 4.3. CoverType Scatter Plots of Losses



(a) $\alpha = 1.0$



(b) $\alpha = 0.97$

Density distribution of reconstruction errors. Note the rise of a third cluster.

Figure 4.4. CoverType Density Plots of Losses

4.1.4 Fashion-MNIST

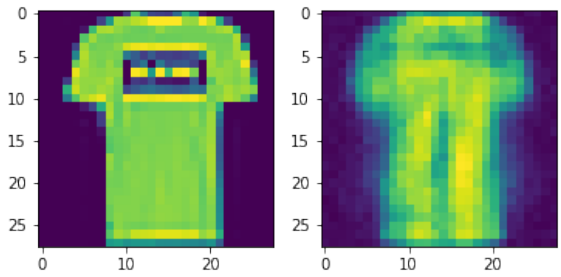
The Fashion-MNIST data is our only image benchmark. Along with the analysis of reconstruction errors, it provides visualization of the VAE's learned behavior (see Figure 4.5). The data contains 1,203 images of clothing and 60,000 images of handwritten digits. We test 10 values of α and expect the optimal value to be at or near 0.98, the true proportion of anomalies (clothing images). Our results show $\alpha = 0.97$ is the best choice and produces an F_1 Score of just over 0.98. This is also the identical value for $\alpha = 0.96$. In fact, all

F_1 Scores with $\alpha \leq 0.98$ are extremely high and within 0.01 of each other (see Table 4.4). This is clearly the most definitive indication that choosing any value of α less than or equal to the true proportion of anomalies is better than the baseline case ($\alpha = 1.0$).

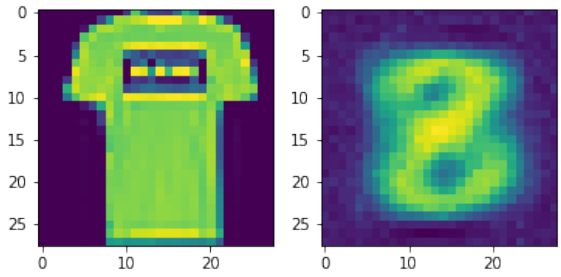
It is clear from Figure 4.5 that our novel objective function has a dramatic affect detecting anomalies. With very few samples in the data, a VAE with a standard objective function ($\alpha = 1.0$) is powerful enough to reconstruct the images of clothing contained within 60,000 images of handwritten digits, despite them being very different. Our objective function prevents the VAE from learning any information from the anomalous images. When the clothing images are fed into the VAE using our objective function, their reconstruction resembles handwritten digits! This results in a large reconstruction error and improves the likelihood of detection.

Table 4.4. Fashion-MNIST F_1 Scores at 98.03th Percentile of Losses

α	F_1 Score
1.00	0.7138
0.99	0.9135
0.98	0.9776
0.97	**0.9842**
0.96	0.9842
0.95	0.9817
0.94	0.9792
0.93	0.9784
0.92	0.9784
0.91	0.9776



(a) $\alpha = 1.0$



(b) $\alpha = 0.97$

This figure illustrates the differences in reconstruction by VAEs that use the baseline ($\alpha = 1.0$) and new ($\alpha = 0.97$) objective function. Notice in subfigure (b) the rightmost image is beginning to look like a number 8, indicating the VAE learned digits (normal data) better during training and cannot reconstruct the fashion articles (anomalies).

Figure 4.5. Fashion-MNIST Image Reconstruction

4.2 Performance with Benchmark Data and α Unknown

We have shown our new objective function, with $\alpha < 1.0$, improves anomaly detection in our benchmark data. Additionally, knowing their labels helped provide instant feedback on the VAE's performance and enabled us to choose α optimally and subsequently the best model. The task now is to continue selecting the optimum α when the data is unlabeled and the proportion of anomalies are unknown. We use the same benchmarks but remove their labels and focus only on the VAE's ability, or inability, to reconstruct data and their associated reconstruction error distributions.

In Chapter 3, we introduce GMMs as a way to fit the reconstruction errors into groups that can be used for analysis. Measuring the relative sizes, mean distance from the origin, and

their variability provides insight into how the VAE is working to reconstruct the data.

Using the same values of α as the previous section, we fit the models' reconstruction errors into a three-component GMM. Tables 4.5, 4.6, 4.7, and 4.8 show the mean, variance, and weight for the rightmost GMM component for each benchmark. We observe three key facts when fitting the GMM on the reconstruction errors for the third component:

1. The optimal α has a large mean
2. The optimal α has a small variance
3. The optimal α has a small weight

This reinforces our initial thoughts about the distribution of reconstruction errors described in Chapter 3.1. The α which returns the best results also has a small, dense, and high-mean group of reconstruction errors in the rightmost component. The Statlog Shuttle is the only exception to this since its weight is 0.0200, which ranks 7th among the other 10 choices. Despite this difference the trend is clear and provides a quantifiable metric for determining the best-performing VAE to utilize for anomaly detection with unlabeled data.

Table 4.5. KDDCup99 GMM 3rd (Rightmost) Component

α	Mean	Variance	Weight
1.00	29.78	221.70	0.0259
0.995	**88.41**	**569.03**	**0.0055**
0.99	80.19	610.84	0.0066
0.98	53.63	977.33	0.0317
0.97	45.50	1,049.56	0.0413
0.96	48.46	1,050.95	0.0373
0.95	42.55	1,038,51	0.0455
0.94	50.54	1,022.18	0.0358
0.93	42.17	1,049.36	0.0466
0.92	54.53	985.31	0.0593

Table 4.6. Statlog Shuttle GMM 3rd (Rightmost) Component

α	Mean	Variance	Weight
1.00	13.08	81.66	0.0037
0.98	11.34	61.08	0.0055
0.96	15.55	105.20	0.0027
0.94	12.39	86.27	0.0041
0.93	6.65	10.20	0.0830
0.92	7.74	13.38	0.0566
0.91	16.54	109.83	0.0030
0.90	**30.99**	**11.26**	**0.0200**
0.89	22.96	199.66	0.0019
0.88	32.37	15.49	0.0387

Table 4.7. CoverType GMM 3rd (Rightmost) Component

α	Mean	Variance	Weight
1.00	35.93	3.55	0.0594
0.99	36.20	4.68	0.0536
0.98	36.73	11.05	0.0541
0.97	**62.34**	**1.79**	**0.0114**
0.96	62.29	2.66	0.0113
0.95	62.79	2.56	0.0117
0.94	63.13	13.10	0.0117
0.93	63.03	2.95	0.0163
0.92	62.57	2.78	0.0174
0.91	62.56	2.33	0.0118

Table 4.8. Fashion-MNIST GMM 3rd (Rightmost) Component

α	Mean	Variance	Weight
1.00	167.36	4,784.12	0.1255
0.99	280.63	11,330.43	0.0284
0.98	376.24	14,029.17	0.0234
0.97	**442.80**	**7,556.40**	**0.0201**
0.96	454.23	11,602.98	0.0206
0.95	434.84	8,500.19	0.0204
0.94	444.83	12,068.55	0.0210
0.93	436.91	9,708.67	0.0208
0.92	428.47	10,455.35	0.0213
0.91	441.59	10,351.34	0.0209

4.3 Performance with DoN Contract Award Data

Given the method described in Chapter 3 and the results from Chapter 4.1 and 4.2, we apply what we discovered to our unlabeled data, DoN Contract Awards.

The data consists of 292,853 records and 997 features. It is compiled from USASpending (2019) and encompasses DoN contract awards from FY14-FY18 in NACIS sector 54 (Professional, Scientific, and Technical Services).

We perform a larger grid search for α than our benchmarks by training the VAE on 16 different possible values. Immediately, we notice our original network configuration used with our benchmark data is unstable and reconstruction error values diverge toward infinity. This could be due to the large number of binary features (991), but nonetheless a deeper network or smaller learning rate is required to remedy this problem. We add an additional hidden layer to both the encoder and decoder of the VAE and leave all other parameters untouched. Table 4.9 shows the updated VAE architecture used in this experiment. We note that this is one of the benefits of our methodology. Our novel objective function and GMM selection process can be used around any VAE model architecture. This is, in fact, one of the reasons we chose to focus on the application of neural networks to anomaly detection.

Table 4.9. Updated VAE Architecture for DoN Contract Award Data

Layer	Nodes
Input	997
Hidden 1	512
Hidden 2	256
Hidden 3	48
Hidden 4	256
Hidden 5	512
Output	997

Based on the analysis of the reconstruction error distributions, initial impressions indicate the best-performing VAE has an $\alpha = 0.92$ or $\alpha = 0.88$. If the true proportion of anomalies is around 10% then these values are plausible based on the findings using the labeled benchmark data. More generally, the VAE behave as expected as α changes. The right-tails of the reconstruction error distributions begin to lengthen and develop a second grouping. Figures 4.6 and 4.7 show this well. When the baseline $\alpha = 1.0$ reconstruction error distribution is compared with the ones with $\alpha = 0.92$ and $\alpha = 0.88$ we clearly see the right-tails emerge, which presumably consist of anomalies. Recalling the desired distribution from Chapter 3 (see Figure 3.1), we conclude the VAE with $\alpha = 0.88$ resembles this the best but are reluctant to dismiss $\alpha = 0.92$ as a possible optimum.

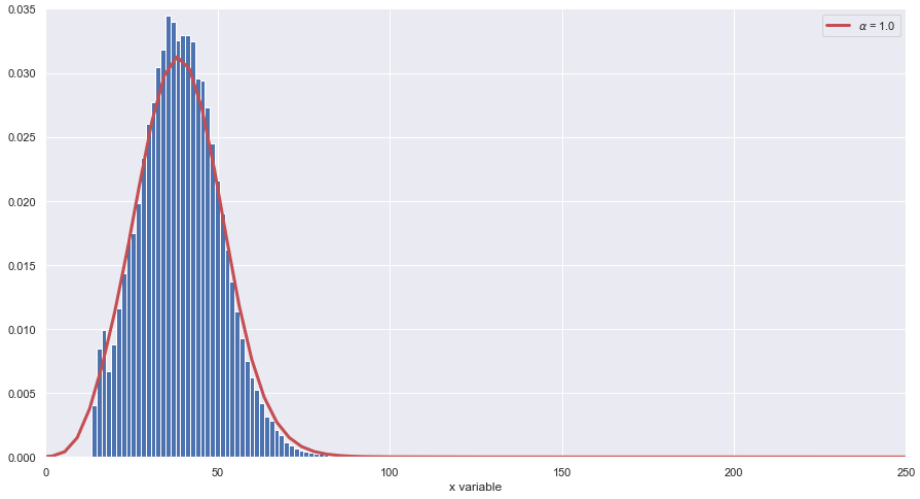
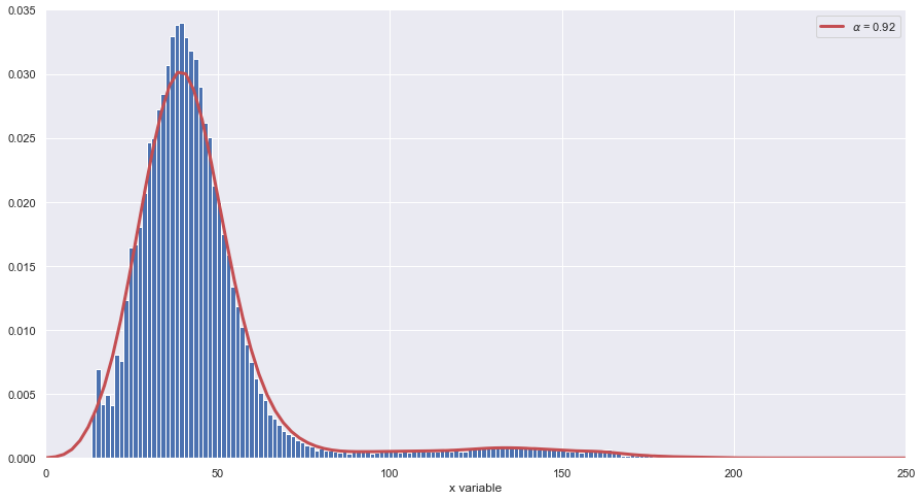
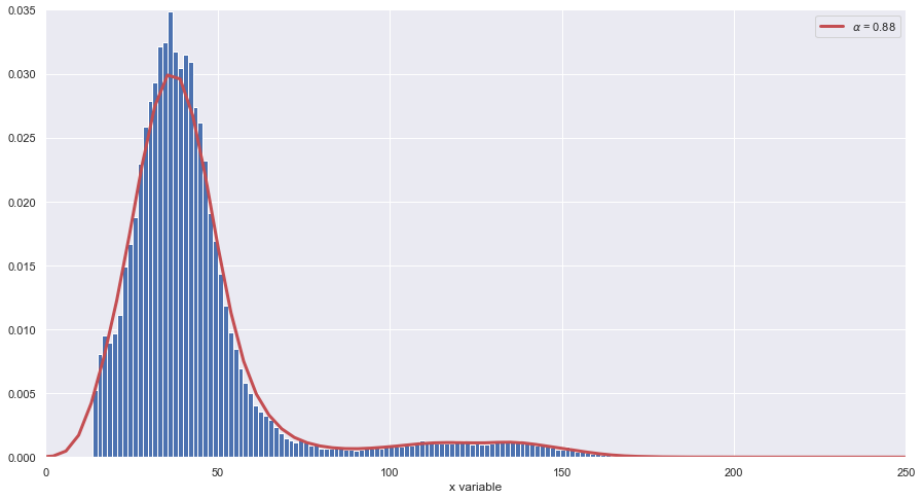


Figure 4.6. DoN Contract Award Reconstruction Error Distribution for $\alpha = 1.0$



(a) $\alpha = 0.92$



(b) $\alpha = 0.88$

Figure 4.7. DoN Contract Award Reconstruction Error Distributions for (a) $\alpha = 0.92$ and (b) $\alpha = 0.88$

With two possible candidate VAE models identified based on their reconstruction error distributions, we now fit a GMM to each of the 16 models to identify (or confirm) which performs the best detecting anomalous data. Table 4.10 displays the results. Based on mean, variance, and weight of the 3rd component of the GMM, both $\alpha = 0.92$ and $\alpha = 0.88$ perform the best; $\alpha = 0.92$ has the highest mean and $\alpha = 0.88$ has nearly the smallest variance. The variance for $\alpha = 1.0$ and $\alpha = 0.99$ are smaller, but we ignore those choices

of α based on their small and insignificant means relative to the other models. A preferred model is still elusive at this point, but we can safely assume it will be between $\alpha = 0.92$ and $\alpha = 0.88$

Table 4.10. DoN Contract Award GMM 3rd (Rightmost) Component

α	Mean	Variance	Weight
1.00	48.17	153.41	0.2720
0.99	48.64	329.20	0.1426
0.98	78.05	1,114.88	0.0270
0.97	80.38	844.05	0.0334
0.96	83.42	938.44	0.0400
0.95	88.24	863.68	0.0380
0.94	94.19	1,032.86	0.0608
0.93	113.34	1,310.58	0.0613
0.92	**121.75**	**1,032.86**	**0.0634**
0.91	108.27	888.10	0.0727
0.90	109.09	923.22	0.0800
0.89	116.63	870.91	0.0795
0.88	**114.28**	**731.99**	**0.0847**
0.87	110.27	996.31	0.1034
0.86	113.84	936.45	0.1086
0.85	111.71	932.72	0.1142

For thoroughness, we identify the worst 1,000 reconstructed records in each model's outputs and determine the fraction of them that are shared between models. In other words, we compare the worst records for $\alpha = 0.92$ against each α and count the number of times a particular record appears in both subsets. This provides a clue about the differences and similarities each model has in reconstructing these records. We find the 1,000 worst reconstructed outputs for $\alpha = 0.92$ comprise 69% of the 1,000 worst reconstructed outputs $\alpha = 0.88$. See Table 4.11 for a complete look at the similarities of $\alpha = 0.92$ with the other 15 values. The results of this comparison is interesting but does not assist in the overall goal of selecting the best α; however, we can conclude $\alpha = 0.92$ and $\alpha = 0.88$ produce the most similar results, confirming them as our best performing relative the the GMM selection criteria.

Two arguments can be extracted from the benchmark results for choosing between the two

Table 4.11. Similarity of 1,000 Worst Reconstructed Outputs between α = 0.92 and All Other α

α	Similarity
1.00	0.042
0.99	0.046
0.98	0.071
0.97	0.106
0.96	0.116
0.95	0.101
0.94	0.427
0.93	0.553
0.92	1.000
0.91	0.487
0.90	0.388
0.89	0.552
0.88	**0.691**
0.87	0.496
0.86	0.418
0.85	0.430

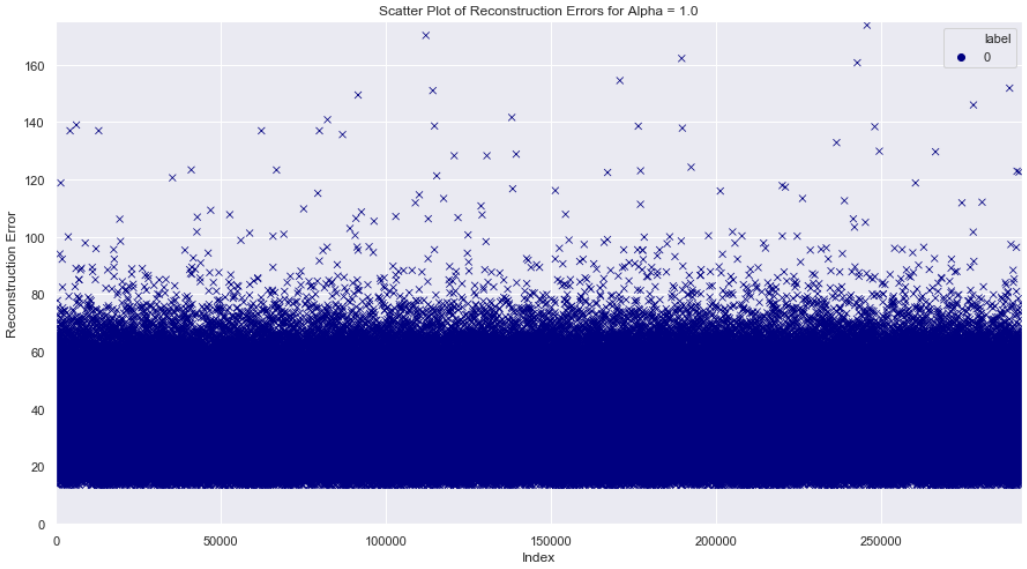
α. The conservative one would be to select $\alpha = 0.88$. Each model performed better using $\alpha < 1.0$, so in this case, simply choosing the one whose variance is small and has a relatively high mean would certainly be justified. On the other hand, we observe with the benchmarks the best performing VAE has an α that is no farther than 0.03 of the true proportion of anomalies. Presumably, the most likely number of anomalies in the DoN contract award data is smaller than 9-12%, so choosing $\alpha = 0.92$ may be ideal and supports a new assumption that the number of anomalies could be 5-8%.

After considering the results from the benchmarks, $\alpha = 0.92$ is arguably the best choice. It has the highest mean and 25% lower weight than $\alpha = 0.88$ in the GMM third component, although we concede its variance is the larger of the two. Additionally, the likelihood the number of anomalies is more than 8% seems low and provides a lower bound of about 5%, but again this is only an assumption on the true proportion of anomalies. Further, the weight of $\alpha = 0.92$ is 0.0634. Assuming there is some quantity of high-loss normal data within this GMM third group, we can also assume there is less than 6% anomalies (i.e., the total

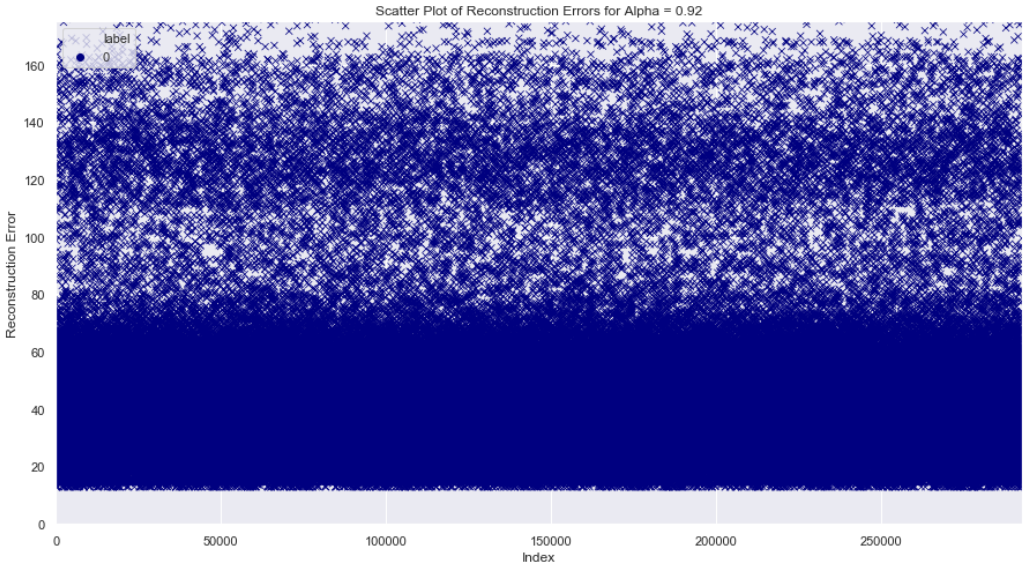
weight of third component). This is true with three of four results from the benchmark data (Shuttle Statlog did not have this characteristic in the results).

Determining the cutoff between normal and anomalous records, we start with the weight of the third group of the fitted GMM on $\alpha = 0.92$ model as the initial threshold. In other words, we mark the spot to "slice" the right-tail of the reconstruction error distribution at the 93.66th-percentile. Splitting here results in 18,567 anomalous records. Trying a Tukey Fence $[Q_3 + 3 * (Q_3 - Q_1)]$ instead results in 13,378 anomalies (4.57%), and if we look for natural breaks in the reconstruction error distribution we find two possible values (see Figure 4.8). Using a break point with a reconstruction errors greater than 100 and 150 result in 12,772 (4.36%) and 2,175 (0.74%) anomalies, respectively.

Ultimately, for the DoN contract award data, we conclude the best-performing VAE model uses our proposed percentile objective function with $\alpha = 0.92$, and the number of possible anomalous records is bounded between 2,175 – 18,567 records, or 0.74% – 6.34% of the data.



(a) $\alpha = 1.0$



(b) $\alpha = 0.92$

Figure 4.8. DoN Contract Award Reconstruction Scatter Plots for (a) $\alpha = 1.0$ and (b) $\alpha = 0.92$

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusion

With data collection becoming more ubiquitous in both the commercial and military sectors, the power to detect anomalies within that data has increasing value for decision makers. The impacts vary by domain, but the effects of anomalies can be overwhelming beneficial if detected or equally as devastating if missed. Our proposed method attempts to add another tool for analysts to use.

Returning to the research questions in Chapter 1, we address each individually throughout this thesis. Again, those questions are:

- What is the best network architecture to effectively detect anomalies and how can we alter the objective function of the VAE to better detect them? Similarly, can we force the distribution of reconstruction errors to follow a desired distribution with anomalous data associated with large reconstruction errors that are far from the distribution of normal reconstruction errors (e.g., long-tail and bimodal)?
- How should we analyze the distribution of reconstruction errors to indicate anomalies (i.e., how large does the reconstruction error need to be for a point to be labeled anomalous)? Without ground-truth labels, how do we know if our algorithm is performing well?
- Can our proposed technique be generalized to perform effectively on many datasets? If so, how well?

While network architecture varied slightly between data, we kept parameters as consistent as possible between different experiments and data. Ultimately, each dataset required its own unique VAE architecture based primarily on the size of its features. This was not a problem since our comparison was localized by varying the choice of α for the same dataset with the same architecture. Generalizing the network architecture for all five datasets was unfeasible and also unnecessary.

Altering our objective function was undeniably the most crucial and beneficial aspect for increasing anomaly detection using VAEs. Introducing the new hyperparameter α improved

detection in every benchmark dataset. Simply using *any* $\alpha < 1.0$ saw an improvement. Additionally, we achieved success forcing the distribution of reconstruction errors to follow our desired pattern (see Figure 3.1).

Using a GMM with three components to fit the distribution of reconstruction errors, we assign quantifiable metrics (mean, variance, and weight) to analyze the different choices of α for each VAE. Choosing the best-performing model, and associated α, is predicated on a large mean, small variance, and small weight in the GMM third component. Furthermore, in the DoN contract award data, we offer several different approaches for determining the cutoff between normal and anomalous (weight of 3rd GMM component, natural breaks in the distributions, and a Tukey Fence).

Generalizing our approach to work on many datasets is certainly feasible. While the VAE network architecture will be dependent on the data, using our proposed percentile objective function with $\alpha < 1.0$ will increase anomaly detection over the baseline VAE ($\alpha = 1.0$). The improvement will vary, but the benchmark data had an average increase of roughly 0.4021 in their F_1 scores.

While this research shows our method increases anomaly detection compared with a baseline VAE, it leaves areas of future work and study. Namely, altering the network architectures and parameters and capturing their effects. Implementing a Design of Experiments and choosing more than just the optimum α, but all optimum parameters, could further increase detection. This research deliberately focused on only studying the effects of the proposed objective function on anomaly detection, but a more in-depth and robust study could certainly strengthen this method.

Repeated experiments and automation are two additional areas for continued research. With anything stochastic, like VAEs, outcomes can vary between execution. Again, this thesis focused on determining how best to enhance anomaly detection with VAEs and a novel objective function, and now that this question is answered, knowing the bounds in which the results are contained could provide additional insights. In other words, running thousands of experiments with varying seeds could be fruitful for continued refinement of the method. Automating this entire process, rather than manually judging the relative merits of α values, like with the DoN contract award data, could provide a usable tool and quick means of assessment.

List of References

An J, Cho S (2015) Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture, December 27, SNU Data Mining Center, Seoul, South Korea. http://dm.snu.ac.kr/static/docs/TR/SNUDM-TR-2015-03.pdf.

Baur C, Wiestler B, Albarqouni S, Navab N (2018) Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. *MICCAI* (Granada, Spain), http://arxiv.org/abs/1804.04488.

Blackard JA (1998) Forest Covertype data. Accessed April 19, 2019, https://archive.ics.uci.edu/ml/datasets/covertype.

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: A survey. *ACM Comput. Surv.* 41(3), http://doi.acm.org/10.1145/1541880.1541882.

Doersch C (2016) Tutorial on variational autoencoders. *arXiv* abs/1606.05908, https://arxiv.org/abs/1606.05908.

Dua D, Graff C (2017) UCI machine learning repository. Acesssed January 15, 2019, http://archive.ics.uci.edu/ml.

Galaxy Data Technologies (2019) Introduction to autoencoders. Accessed May 17, 2019, https://galaxydatatech.com/2018/10/21/introduction-to-autoencoders/.

Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning*. http://www.deeplearningbook.org.

Goyal N (2018) Implementation of perceptron algorithm using Python. Accessed May 17, 2019, https://mlforanalytics.com/2018/04/29/implementation-of-perceptron-algorithm-using-python/.

Hettich S, Bay SD (1999) UCI machine learning repository. Accessed January 15, 2019, http://archive.ics.uci.edu/ml.

Kingma DP, Ba JL (2015) Adam: A method for stochastic optimization. *ICLR* (San Diego, CA), https://arxiv.org/abs/1412.6980.

Kingma DP, Welling M (2014) Auto-encoding variational Bayes. *ICLR* (Banff, Canada), https://arxiv.org/abs/1312.6114.

Krogh A, Hertz JA (1992) A simple weight decay can improve generalization. *NIPS* (Denver, CO), http://papers.nips.cc/paper/563-a-simple-weight-decay-can-improve-generalization.

Laerd Statistics (2019) Linear regression analysis using SPSS statistics. Accessed May 8, 2019, https://statistics.laerd.com/spss-tutorials/linear-regression-using-spss-statistics.php.

Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), https://doi.org/10.1109/5.726791.

Mehrotra KG, Mohan C, Huang H (2017) *Anomaly Detection Principles and Algorithms*. https://www.springer.com/us/book/9783319675244.

Park D, Hoshi Y, Kemp CC (2018) A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEE Robotics and Automation Letters* 3(3), https://doi.org/10.1109/LRA.2018.2801475.

Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in PyTorch. *NIPS* (Long Beach, CA), https://openreview.net/pdf?id=BJJsrmfCZ.

Priy S (2019) Clustering in machine learning. Accessed May 17, 2019, https://www.geeksforgeeks.org/clustering-in-machine-learning/.

Reynolds D (2015) *Encyclopedia of Biometrics*. https://doi.org/10.1007/978-1-4899-7488-4_196.

Robinson S (2018) K-nearest neighbors algorithm in Python and Scikit-Learn. Accessed May 17, 2019, https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/.

Rockafellar RT, Uryasev S (2002) Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance* 26(7), https://doi.org/10.1016/S0378-4266(02)00271-6.

Rosenblatt F (1961) Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. Technical Report VG-II96-G-8, Cornell Aeronautical Laboratory, Inc, Buffalo, NY, https://apps.dtic.mil/dtic/tr/fulltext/u2/256582.pdf.

Rumelhart DE, McClelland JL, PDP Research Group C (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations* (MIT Press, Cambridge, MA).

Sarle WS (1994) Neural networks and statistical models. *SAS Users Group International Conference* (Cary, NC), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.699.

Song X, Wu M, Jermaine C, Ranka S (2007) Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering* 19(5), https://doi.org/10.1109/TKDE.2007.1009.

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1), http://jmlr.org/papers/v15/srivastava14a.html.

Stars and Stripes (2018) Contractor inchcape to pay $20 million to settle fraud case over charges to navy (May 29), https://www.stripes.com/news/navy/contractor-inchcape-to-pay-20-million-to-settle-fraud-case-over-charges-to-navy-1.530039.

SuperDataScience Team (2018) The ultimate guide to convolutional neural networks (CNN). Accessed May 17, 2019, https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn.

Tan SC, Ting KM, Liu FT (2011) Fast anomaly detection for streaming data. *IJCAI* (Monash University, Australia), https://apps.dtic.mil/dtic/tr/fulltext/u2/a556329.pdf.

Upadhyaya S, Singh K (2012) Classification based outlier detection techniques. *International Journal of Computer Trends and Technology* 3(2), https://pdfs.semanticscholar.org/d2c6/975da8ebbdf3db01f6614b750aa75f6501fd.pdf.

US Department of Justice (2018) Northrop Grumman subsidiary agrees to pay $31.65 million for overbilling U.S. Air Force in civil and criminal settlements. Accessed May 8, 2019, https://www.justice.gov/usao-sdca/pr/northrop-grumman-subsidiary-agrees-pay-3165-million-overbilling-us-air-force-civil-and.

USASpending (2019) Accessed April 19, 2019, https://www.usaspending.gov/#/about.

Vanderplas J, Connolly A, Ivezić Ž, Gray A (2012) Introduction to astroml: Machine learning for astrophysics. *CIDU* (Boulder, CO), https://arxiv.org/abs/1411.5039.

Washington Post (2016) The man who seduced the 7th fleet (May 27), https://www.washingtonpost.com/sf/investigative/2016/05/27/the-man-who-seduced-the-7th-fleet/?noredirect=on&utm_term=.56384fc79997.

Weng L (2019) From autoencoder to beta-VAE. Accessed May 17, 2019, https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html.

Williams G, Baxter R, He H, Hawkins S, Gu L (2002) A comparative study of RNN for outlier detection in data mining. *ICDM* (Maebashi City, Japan), https://ieeexplore.ieee.org/document/1184035.

Xiao H, Rasul K, Vollgraf R (2017) Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. Accessed April 30, 2019, https://arxiv.org/pdf/1708.07747.pdf.

Xu H, Chen W, Zhao N, Li Z, Bu J, Li Z, Liu Y, Zhao Y, Pei D, Feng Y, Chen J, Wang Z, Qiao H (2018) Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. *CoRR* abs/1802.03903, http://arxiv.org/abs/1802.03903.

Zhang GP (2000) Neural networks for classification: A survey. *IEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews* 30(4), https://doi.org/10.1109/5326.897072.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California