Error detection in blood work

A comparison of self-supervised deep learning-based models

PAUL VINELL

Master's Programme, Computer Science, 120 credits
Date: June 20, 2022

Supervisor: Pawel Herman
Examiner: Arvind Kumar
School of Electrical Engineering and Computer Science
Host company: Werlabs AB
Swedish title: Felupptäckning i blodprov
Swedish subtitle: En jämförelse av självbevakade djupinlärningsmodeller

Abstract

Errors in medical testing may cause serious problems that has the potential to severely hurt patients. There are many machine learning methods to discover such errors. However, due to the rarity of errors, it is difficult to collect enough examples to learn from them. It is therefore important to focus on methods that do not require human labeling. This study presents a comparison of neural network-based models for the detection of analytical errors in blood tests containing five markers of cardiovascular health. The results show that error detection in blood tests using deep learning is a promising preventative mechanism. It is also shown that it is beneficial to take a multivariate approach to error detection so that the model examines several blood tests at once. There may also be benefits to looking at multiple health markers simultaneously, although this benefit is more pronounced when looking at individual blood tests. The comparison shows that a supervised approach significantly outperforms outlier detection methods on error detection. Given the effectiveness of the supervised model, there is reason to further study and potentially employ deep learning-based error detection to reduce the risk of errors.

Keywords

anomaly detection, outlier detection, error detection, machine learning, deep learning, blood work, blood tests

Sammanfattning

Fel i medicinska tester kan orsaka allvarliga problem som har potential att allvarligt skada patienter. Det finns många maskininlärningsmetoder för att upptäcka sådana fel. Men på grund av att felen är sällsynta så är det svårt att samla in tillräckligt många exempel för att lära av dem. Det är därför viktigt att fokusera på metoder som inte kräver mänsklig märkning. Denna studie presenterar en jämförelse av neurala nätverksbaserade modeller för detektering av analytiska fel i blodprov som innehåller fem markörer för kardiovaskulär hälsa. Resultaten visar att feldetektering i blodprov med hjälp av djupinlärning är en lovande förebyggande mekanism. Det har också visat sig att det är fördelaktigt att använda ett multivariat tillvägagångssätt för feldetektering så att modellen undersöker flera blodprov samtidigt. Det kan också finnas fördelar med att titta på flera hälsomarkörer samtidigt, även om denna fördel är tydligare när modellen tittar på individuella blodprov. Jämförelsen visar att ett övervakat tillvägagångssätt avsevärt överträffar metoder för detektering av extremvärden vid feldetektering. Med tanke på effektiviteten av den övervakade modellen finns det anledning att studera tillvägagångssättet vidare och eventuellt använda djupinlärningsbaserad feldetektering för att minska risken för fel.

Nyckelord

felupptäckning, extremvärden, maskininlärning, djupinlärning, blodprov

Acknowledgments

I would like to thank my supervisor Dr. Pawel Herman for his guidance and feedback. With an excellent eye for both language and methodology, without his help this report would not be what it is today. Likewise, I want to thank my industrial supervisor Jens Stjernström whose excellent ideas and questions have made me rethink and reinvent this project and my approach multiple times. I also extend my gratitude to Werlabs AB for providing me with the opportunity to work with this unique and exciting dataset.

I have been fortunate to have several other great contacts at Werlabs. I want to thank fellow master's student Felix Gudéhn for his support and feedback. I also want to thank physicians Dr. Johan Weidenhaijn and Dr. Lovisa Krantz for helping me with their knowledge and expertise. Without it, this work would not be possible as it has crucially shaped the evaluation approach, error simulation.

Lastly, I want to thank friends and family that have supported me during this process. I am grateful to have a great support system that backs me up in any and all efforts.

Stockholm, June 2022
Paul Vinell

Contents

Chapter 1

Introduction

1.1 Background

Blood is one of the most common samples tested in a clinical laboratory. In every blood test one or more health markers (blood test parameters) are measured. Since these results can be the basis for diagnoses or lifestyle change recommendations, lab results should be accurate. If they are not, mistakes can be made that have serious consequences. False positives can result in possibly dangerous interventions, and false negatives can result in neglect. One way to combat this is known as quality control samples [1]. When tested, these samples are expected to give a certain value and are thus used to verify equipment calibration and function. However, errors can still occur due to operator errors or equipment malfunction. If, for instance, quality control samples are forgotten after a recalibration of some instrument, then analytical errors (i.e. measurement errors) can go unnoticed. Due to the risk and possible severity of consequences, the need for extra lines of defense is well motivated.

One such idea for an extra line of defense is to regularly analyze the values produced by the laboratories (exemplified in Figure 1.1). This is to some degree already done by any physician or other educated personnel that has the opportunity to observe any of the produced values. Since there is some degree of expectation about what the value should be, any significant deviations will stand out as suspicious. If multiple measurements produced by the same equipment follow a similar trend, there is reason to suspect the presence of systematic errors. This is systematized by outlier detection methods, techniques that measure deviations from the norm. Outliers do not necessarily imply errors. An anomalous blood test may just represent a sick or unhealthy person. But when what would once be outliers seems to become the norm, there is reason to investigate. If the threshold for classifying blood tests as errors is set too low, the warnings from the model runs the risk of being ignored by the laboratory staff after one too many false alarms. Conversely, if the threshold is set too high it runs the risk of missing errors that go on to affect patient outcomes. However, assuming that it functions well, automated outlier detection would be a powerful intervention.

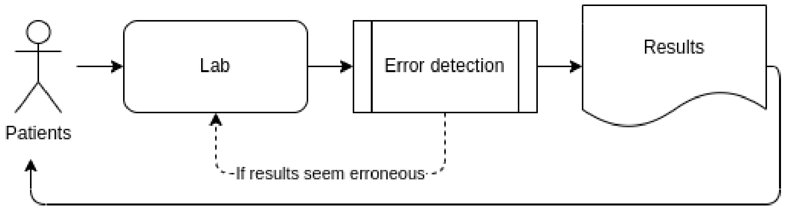

Figure 1.1: Simplified diagram. Error detection is the process that lets a lab know that measurements are failing and prevents patients from getting erroneous results.

Today, outlier detection is performed through a variety of approaches. A first general approach is to detect deviations from the norm in a univariate fashion [2]. Significant deviations can, for instance, be defined using statistical means or by distance measurements. Then there are multivariate approaches to identify deviations from the norm. Although a univariate analysis may suffice, it can leave room for improvement. Physicians working with blood tests know this as they may spot errors, not entirely motivated by the erroneous health marker itself, but also based on the value they would expect given the other measured health markers. The family of multivariate approaches includes methods such as distance-based methods, clustering, and artificial neural networks [3, 4]. The latter will be the subject of analysis in this study since it has proven itself capable of learning features of the data and complex relations in it automatically, a desirable property when working with data such as blood tests, since markers may have complex or subtle dependencies and relations [5].

1.2 Problem

1.2.1 Original problem and definition

While errors are in general rare, the most commonly occurring type of error in blood tests is analytical errors. These are errors that in their cause are marked by equipment failure or operator errors (e.g. forgetting to calibrate the equipment). In appearance, they can be recognized by a constant error factor. The distribution of these erroneously measured blood tests will deviate from the ordinary distribution. The details of this deviation will differ depending on the cause of the error and the circumstances surrounding that cause. Due to errors being rare it is unlikely to find blood tests affected by multiple errors at once.

In a lab, blood tests are typically measured together or at least by the same equipment. A consequence of this is that analytical errors typically present themselves as collective outliers; they are data points that differ from the norm in a shared way. Another consequence of this way of measuring is that a time dependency of the errors is created. Incorrectly measured blood tests are usually surrounded in time by other incorrectly measured blood tests (with respect to the same health marker).

Due to the rarity of errors, there is limited data on examples of them and their presentation. Moreover, it is often not possible to look back at historical data and conclude definitively whether a specific measurement was performed erroneously or not. If it was possible to conclude errors in such a manner, they would not constitute a problem. Suspected errors have to be accompanied by an investigation by the performing lab around the time of their appearance to have them validated. If too much time passes, equipment will likely be recalibrated and the ability to follow up on past errors becomes very limited.

As a data-driven field of study, the aforementioned data limitations throw a wrench into the machine learning process. Due to a lack of labels signifying which data points are erroneous, such a signal must be derived from the data by other means.

1.2.2 Research question

When referring to state-of-the-art models for outlier detection we are referring to f-AnoGAN [6] and Skip-GANomaly [7]. Both are built on the same fundamental principle, anomalous data is harder to compress and reconstruct. However, while the two models have the same core principle, the structure and training methodology differ significantly. The methodology behind Skip-GANomaly is concurrent and highly specific to the task. It consists of one part that compresses and reconstructs the data, along with a judge that judges the reconstruction. f-AnoGAN on the other hand focuses on learning the data distribution first, and then complements that with learning to compress the data. f-AnoGAN determines whether a data point is an outlier by trying to compress it, restoring it, and then compressing it again. If this goes well, the data is typical, otherwise it is anomalous. Then there are models that could be considered candidates for best performers. One of these models is the predecessor to Skip-GANomaly, called GANomaly [8]. Other relevant models are autoencoders [9], and EGBAD [10]. Lastly, this study adds simulated errors to the data to evaluate the models. This technique also allows for training regular supervised models [4], so this approach will also be considered as a candidate. The plethora of candidates motivate the research question:

How do the aforementioned methods compare in application to identifying errors in blood work?

1.3 Aim and contributions

The aims of this project were to provide a comparison of state-of-the-art outlier detection models as tools for error detection in blood tests. What separates this from other comparisons is that the models have been adapted for non-spatial data. As part of this investigation, the study also sheds light on the effectiveness of univariate versus multivariate approaches to error detection in blood tests. Similarly, it also reveals the performance differences after using labeled versus unlabeled training.

1.4 Scope and limitations, assumptions

For most models, it seems that using multiple instances in an ensemble improves performance [11], and it is relevant to this study to find the best-performing model. It would therefore seem relevant to include ensembles. But, such an evaluation will not be included. Although it is possible that an ensemble of individually worse models will outperform an ensemble of individually better models, it does not seem likely enough to be worthy of investigation. It would also distract from the main desired contribution, a comparison of individual models.

For similar reasons, only models specialized in outlier detection will be evaluated. There are multi-purpose models that can perform outlier detection but these are not what we are looking for when it comes to state-of-the-art performance in the task. These models are therefore not included.

There will also be no evaluation of different datasets. This study focuses on outlier detection in blood test results. Using multiple datasets would help determine the best overall models, but it would also increase the complexity of the analysis and remove focus from the particular application.

1.5 Thesis outline

So far, an overview of the problem at hand has been presented, along with the basic premise of error and outlier detection. In chapter 2, Background, necessities for understanding the study in sufficient detail is provided. Readers familiar with machine learning and deep learning are recommended to skip sections 2.2 Supervisory signals, 2.4 Artificial neural networks, 2.5 Training and measuring performance, and 2.6 Model types. The chapter 3, Methods, explains the implementation details, the data normalization, data augmentation, error simulation, model types and standardization procedures, and the method of evaluation. Chapter 4, Results, contains the results from the experiments. These results and key findings are discussed in chapter 5, Discussion, along with their expected scientific contribution. This chapter also includes an ethical and sustainability perspective. The chapter 6, Conclusions, contains the clear takeaways from this study and a longer segment on what remains to be done.

Chapter 2

Background

2.1 Blood test data

A common type of data used in medicine is blood test data. Blood tests are used to diagnose or exclude diseases, gauge or monitor health, and measure responses to treatments or interventions [5, 12]. The body is a system full of causal and correlational connections. While there are in medical settings often specific singular health markers of interest, there are times when the relation between two or more health markers are of primary interest. This goes to show that information sometimes lies not in individual health markers, but due to their relatedness, in their union. The values of a set of health markers can alter our expectations for another set. Bringing this knowledge back to error detection in blood tests it can be seen that a multivariate approach is warranted. An error is more easily spotted the more the results defy our expectations, and using other health markers for information could make models hold more informed and specific expectations, aiding detection. This hope is investigated by the experiments.

2.2 Supervisory signals

Some machine learning models need to be trained to improve. Training is the process of making gradual changes to a model to improve performance. Two of the most common approaches to training models are using supervised and self-supervised (alt. unsupervised) learning [13]. To train a model it needs a so-called supervisory signal, an indication of how the model should change to perform better. Supervised methods require labeled data, meaning it needs an external source for its supervisory signal. Self-supervised methods derive their supervisory signal from the data itself, hence its name. A toy example of supervised training is a dog/cat classifier that distinguishes dogs from cats. To function it does not just need images of dogs and cats, but also a label associated with each image that denotes whether it is a cat or a dog. That label has to come from somewhere, typically a human labeler. Given the training examples, the model can learn to recognize dogs and cats that it has never seen. A toy example of self-supervised learning is image colorization, models that turn grayscale images into colorized ones [14]. These models bypass the need for human labeling by employing a trick. If one has a set of colorized images they can be trivially converted to a set of grayscale images. These grayscale images can now serve as input, and the original colorized images can serve as the supervisory signal. These two ways of approaching learning from data differ simply in whether they get their supervisory signals externally or internally.

2.3 Outlier detection

In this section, we will attempt to define outlier detection and motivate its use for error detection. By its simplest definition, an outlier is something that differs from the norm. But this does not capture its full essence. It must also be considered that deviation from the norm comes in degrees. Breaking this down into workable bits shows that to perform outlier detection, there is an inherent need to define the norm and a way to measure deviations from it.

Anyone familiar with classification may be tempted to pose outlier detection as a classification problem (with labels: normal, outlier). This can be attempted but it may not be as fruitful as imagined. While classification and outlier detection methods typically lay close in spirit, they differ in a few key ways. One such difference is the discreteness of classification. Something is either this or it is that. In outlier detection, a data point can and often does comfortably sit somewhere in between. Outlier detection models typically produce an anomaly score, a continuous measure of how anomalous a data point is. Classification is unlike outlier detection methods concerned with boundaries. In classification, where a data point is placed in relation to the classification boundaries defines what the data point ‘is’. This poses a significant issue to the task of outlier detection. The problem with trying to define the boundaries of an outlier distribution is that there are typically many ways to be an outlier. Moreover, data of significant outliers is rare (by definition), and approximating its distribution, if there is one, can be a challenge. From an adversarial perspective, if the defending party learns by defining the outlier distribution, the adversary will craft input that is outside that distribution.

There are many different ways to categorize outliers. For this study, the two most important distinctions are point and collective outliers [15]. A point outlier represents a data point with rare occurrence. Winning big on the lottery would be an point outlier. Collective outliers by contrast are data points that would not be individually considered anomalous, but the co-occurrence of these data points is anomalous. For instance, winning small on the lottery is not rare, but if a person wins on every single lottery ticket they buy then they may eventually become suspected of fraudulent behavior. The rarity is not the event itself, the rarity comes from the context or the other events.

2.3.1 Error detection

Performing the error detection task by using outlier detection methods is not necessarily an obvious strategy, but outlier detection has useful properties for error detection [4]. Firstly, note that in this context, ‘outlier’ is a subjective label, while ’error’ is an objective one. For a data point to be an outlier there has to be a choice of metric and cutoff point. Depending on these choices, a given data point may or may not be an outlier. By contrast, an error label is not caused by how a data point presents, but is caused by what happened for it to present the way it does. In other words, just because a blood test contains some unreasonable value(s) does not mean that it is an error. Only what happened at the lab can determine whether it is an error. However, it is possible to make generalizations about the appearance of errors. By definition, most data points are normal. Now, if one introduces a measurement error to an otherwise normal blood test, is it not likely that it will now appear more anomalous? And conversely, if a data point is already an outlier, it is possible that an error will make it present as more ordinary than before (e.g. if one health marker is elevated or low it could be made closer to its expected value). However, such an event seems improbable since outliers are rarer than ordinary data, and since only a subset of possible errors will make it present normally. With inspiration from how physicians would find more subtle analytical errors, it could be possible to make error detection more reliable by looking at multiple blood tests measured close in time at the same lab. This might have a regularizing effect so that the outlier detection models are not as affected by single blood test outliers, a hypothesis that is tested through the experiments. In other words, if we look at many blood tests it is normal and probable with some more deviating ones. It is improbable that there will be many outliers in the same group of tests. In comparison, if there is an analytical error present, we can still expect it to be present in all blood tests that lie close in time. In other words, analytical errors present themselves as collective outliers.

To conclude, in these sections on outlier and error detection we motivate why these concepts are closely related. In essence, this is due to the fact that errors do not typically present normally, something that outlier detection models are attuned to. It is also hypothesized that looking at multiple blood tests at once that are measured close in time will make it easier to find errors, since errors tend to come in groups (collective outliers).

2.4 Artificial neural networks

The artificial neural network is a mathematical model inspired by the brain. The neural networks consist of layers of simulated neurons (depicted in Figure 2.1). The class of artificial neural networks with many layers is also commonly referred to as deep learning [13]. The first layer of a neural network is called the input layer. Neurons in the input layer are activated in a fashion that is meant to simulate perceiving the input. For instance, a neural network meant to distinguish between night and day based on light levels can, for instance, increasingly activate its input neuron(s) as the amount of light measured increases. After the input layer comes one or more layers. Every neuron that is not in the input layer connects to one or more neurons in the previous layer. Each neuron typically calculates a weighted sum of the output of the neurons it connects to in the previous layer, adds a bias term, and then calculates some typically non-linear activation function. This is the output of the neuron. The weight factors and the bias terms are usually referred to as the weights of the network. At some point, the neural network concludes with an output layer. The output layer is read to conclude the decision the network has made about the input.

One of the major reasons for using neural networks is that given enough data they have proven to be able to automatically create rich internal representations that allow the network to make powerful predictions. Due to their expressiveness, neural networks are often a good model choice in situations where data is plenty. This is in line with previous studies on outlier detection which have found that models based on artificial neural networks outperform other methods on the outlier detection task [15].

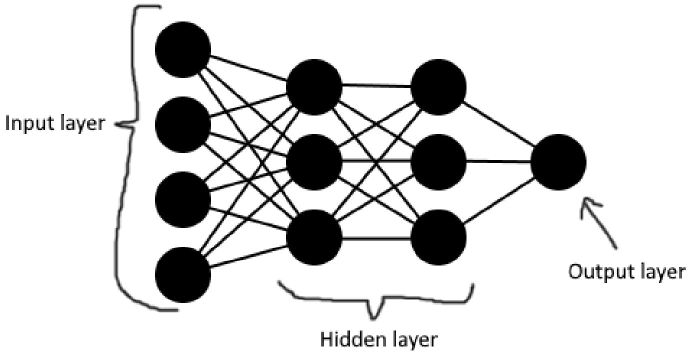

Figure 2.1: Example illustration of a small artificial neural network.

2.4.1 Convolutional neural networks

Convolutional layers give artificial neural networks a greater sense of space or regionality by looking at patches of the data, something that is useful in many cases, perhaps most notably in images [13]. This way of processing input data was inspired by the connectivity patterns in the visual cortex of animals [16]. Most of the models used in this study are originally based on convolutional neural networks. Besides convolutional layers, one of the most typical connectivity pattern for layers in neural networks is “dense” neuronal connections, meaning that given two adjacent layers every neuron in the first layer will connect to every neuron in the second layer. By contrast, convolutional layers process the previous layers in patches using learned filters. On image data, the earlier layers of a convolutional neural network typically learns simple structures such as dots and lines of different orientations. These learned filters are reminiscent of ones that can be found in the vision system of animals. With each additional layer in the network, there is often an associated increase in abstraction and complexity [17]. Depending on the data, of course, one can find more complex composites when visualizing the contents of later layers, such as faces of dogs.

2.5 Training and measuring performance

2.5.1 Training and data splitting

Two central aspects to the training of neural networks is the loss and the data. The loss is a mathematical function that describes how well the neural network is performing [13]. Typically, the loss is said to be better the lower it is. When the model is trained on the data it first makes inferences, then these inferences are scored using the loss function. Since both the neural network and the loss function are mathematical models, it is possible to calculate the gradient of the loss with respect to the network weights. This is done through a process called backpropagation. The weights of the neural network are nudged based on the calculated gradient to lower the loss.

The data is typically split into three parts, training, validation, and test data. The need for separate test data is due to validity concerns. If the model is trained and tested using the same data, it is not possible to tell whether the model has generalized beyond the data it has seen. Validation data is an optional, but useful class of data. During training the model is frequently evaluated on the validation data, which serves as a proxy for how the model will perform on the test data. This way, it is possible to know when to stop training. If the model trains too long, there is a risk that it will overfit to the training data, meaning that it finds patterns that may hold true in the training data, but not in general. There are many ways to measure performance of the network during training. A common way is to use the loss calculated on the validation data (validation loss). However, the validation loss and validation performance do not always go hand in hand. Instead, the area under the precision-recall curve (AUC-PR) can be used. Model sensitivity is an important aspect of error detection. If the model is configured to be sensitive (low threshold), it may catch a lot of errors, but also a lot of false positives. If the model is not sensitive enough (high threshold), it may not suffer many false positives, but it will not catch many errors either. AUC-PR is a measure that captures how well the model does overall by looking at its performance at all possible levels of sensitivity. This method of model selection is suitable for error and outlier detection [18, 19]. Other suitable measures of performance includes the area under the receiving operating curve (AUC-ROC), and F1-score. The former is a similar measure to AUC-PR but considers the trade-off between true and false positives. The latter is calculated by finding a threshold that maximizes the harmonic mean of the precision and recall.

2.5.2 Metrics

In greater detail, when the model scores a data point it outputs an anomaly score. To make a decision about whether to classify the data point as an error it is necessary to pick a threshold. The value of the threshold greatly affects the classification abilities of the model, so performance metrics have to take this into consideration. There are many ways to go about this, as seen above. We start by describing how AUC-ROC solves this. First, we define the receiver operating characteristic (ROC) curve, which captures the relation between the recall and specificity. Mathematically, we define these terms as:

$$\text{Recall} = \frac{\text{TP}}{\text{P}}$$

$$\text{Specificity} = \frac{\text{FP}}{\text{N}}$$

Or in other words, recall is the proportion of true positives (TP) and total positives (P), and specificity is the proportion of false positives (FP) and total negatives (N). The recall and specificity both depend on the chosen threshold, and the ROC curve thus shows how the classification abilities change as the threshold is raised (see the example curve in Figure 2.2). All points in the ROC curve represents how well the model performs at some threshold. The area under this curve therefore represents how well the model performs overall.

Like the AUC-ROC measurement, the AUC-PR and F1 measurements use the recall of the model. But unlike the AUC-ROC, they opt for measuring precision instead of specificity. Precision is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

The precision captures how likely the model is to be right when it classifies a data point as positive. Selecting for precision is desirable in situations where true positives are rare [20]. An example of a precision-recall curve can be seen in Figure 2.3. For the same reasons that the area under the ROC curve captures the overall performance of the model, AUC-PR does as well.

Lastly, although the F1 score is, like AUC-PR, based on precision and recall, it works slightly differently. In short, as mentioned earlier, the F1 score is calculated by finding a threshold that maximizes the harmonic mean of the precision and recall. The harmonic mean is defined as:

$$\text{HM} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

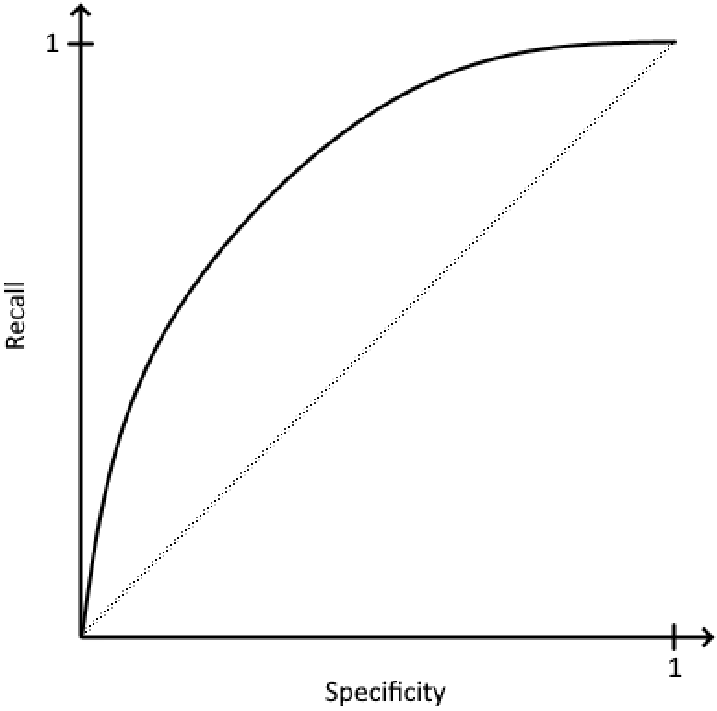

Figure 2.2: Example illustration of a ROC curve. The dotted line corresponds to baseline performance, and the black line corresponds to a non-trivial model. As specificity increases, so does the recall.

Besides the harmonic mean, the maximization is the key differentiation between F1 and AUC-PR. The latter aims to represent the model overall, and the former aims to represent at its best.

2.5.3 Statistics

The Kruskal-Wallis test and Dunn's test [21] are used to verify that the results are statistically significant. Given a set of results from an experiment, the Kruskal-Wallis test determines whether these come from the same distribution. This test is non-parametric which is beneficial since it does not require any assumptions about the distribution of the results. If the test is negative, meaning that two or more distributions produce the results, Dunn's test is performed. This test is also non-parametric but is used to test pairwise differences between sets of results. The Kruskal-Wallis test precedes Dunn's

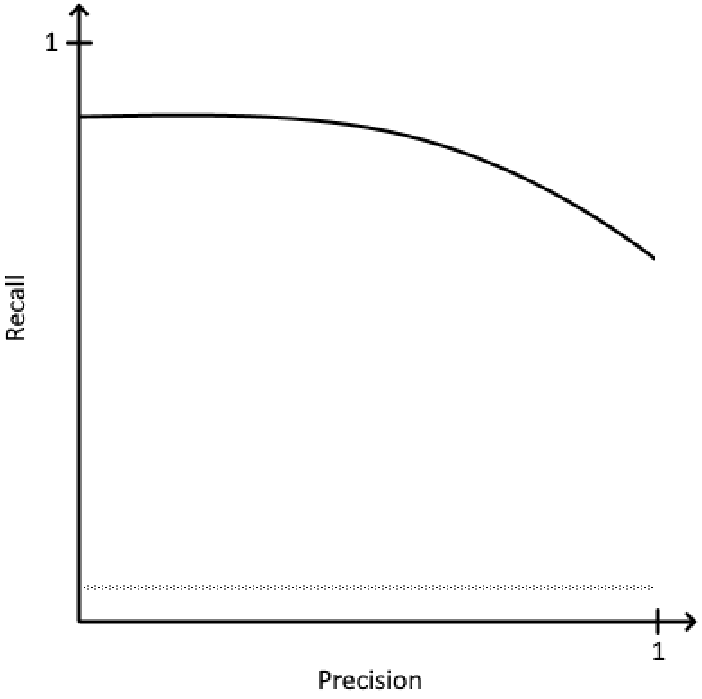

Figure 2.3: Example illustration of a precision-recall curve. The dotted line corresponds to baseline performance, and the black line corresponds to a non-trivial model. As precision increases, the recall drops.

test to prevent type I errors. The significance level is set at $\alpha = 0.05$. To explore correlational relationships between window size and model performance this study uses Pearson's correlation coefficient [21]. This coefficient ranges in value from -1 to 1, where -1 is a linear negative correlation, and 1 is a positive linear correlation. Mathematically, the coefficient is defined as the covariance of two variables divided by the product of their standard deviations.

2.6 Model types

As seen in the section on artificial neural networks, how we choose to structure the connectivity of neurons has implications for the function of the network. In the same way, how we structure and train the network has similarly profound implications for its function. The coming subsections detail different ways artificial neural networks can be constructed to fill new functions. There are simplified illustrations of some of these constructions, a guide to these simplified illustrations is found in Figure 2.4. Each model type mentioned form the basis for one or more of the studied models. These are mentioned later in section 2.7 Related work.

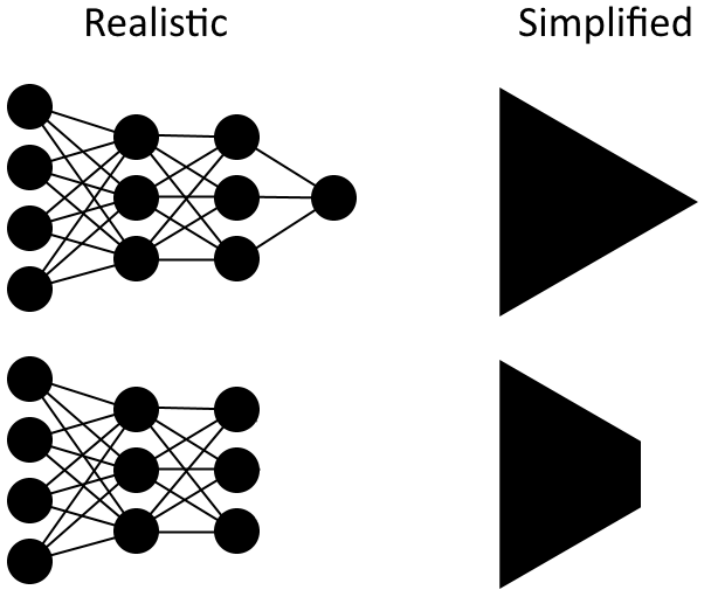

Figure 2.4: The individual neurons are not very important in the constructions and are thus abstracted away.

2.6.1 Classifiers

Among the simplest of models is the classifier, whose job is to classify the input data [13]. This can be binary classification, e.g. determining whether a data point contains an error or not, or n-way classification, determining which of a larger set of classes the data point belongs to (e.g. is the data point a dog, cat, horse, etc.). Structurally, binary classifiers are marked by having one output neuron, whereas n-way classifiers have n output neurons. Classifiers have previously shown success at outlier detection [15], but have the drawback of needing labeled data, which is typically hard to acquire in the circumstances where outlier detection is needed.

2.6.2 Autoencoders

The impact of autoencoders on the field of deep learning and its applications has been vast. The primary contribution of autoencoders was that they enabled self-supervised learning of effective representations of any non-temporal data [13]. There are two main components to the autoencoder, the encoder, and the decoder (depicted in Figure 2.5). The encoder compresses the input data to a smaller representation and the decoder takes that representation and recreates the data. The layer that separates the encoder and decoder is the smallest layer and is called the bottleneck layer or the latent representation. There are many applications of this type of model. For instance, one can train self-supervised on a vast amount of images and then re-train the encoder to classify images using a smaller amount of labeled data. This yields better performance and/or lowers data requirements compared to simply using labeled data. There are also many off-shoot models such as denoising autoencoders, a variant that recovers corrupted data in the input data.

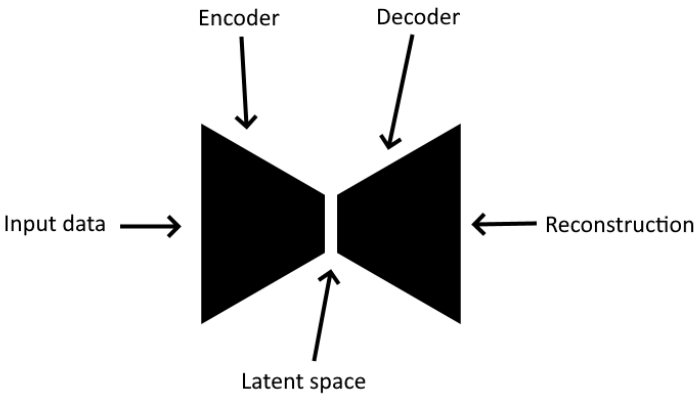

Figure 2.5: A simplified illustration of the structure of an autoencoder.

2.6.3 Generative adversarial networks

One of the most popular classes of generative models is Generative Adversarial Networks (GANs). These models are built on a game-theoretic notion of training where two submodels, the generator, and discriminator, engage in a complementary arms race (adversarial training) [22]. The generator generates synthetic data and the discriminator discriminates between real and synthetic data. The generator aims to fool the discriminator into classifying its data as real, and the discriminator aims to be the perfect inspector. This is a self-supervised approach since the labels are generated during training. The labels required for the discriminator are which data points are real and which are synthetic, this is known. The label required for the generator is how well the discriminator is fooled, this is also known.

On a structural level, GANs are like autoencoders with the components in reverse order (depicted in Figure 2.6). For all intents and purposes, decoders and generators are the same. What unites them is that they both enlarge a small representation to a larger one. What differentiates the two are the task they are trained to perform. The decoder is trained to restore an image from an efficient encoding and the generator to generate a realistic image from a smaller seed of random noise. This seed can be called the latent representation and is roughly equivalent in function to the latent representation of the data in autoencoders. Both are still generative models. The discriminator is similar to the encoder in the sense that it takes complete data as input and reduces its dimensionality. However, while the encoder reduces the data to a smaller representation, the discriminator reduces the data to a single output signifying whether the data is judged to be real or not.

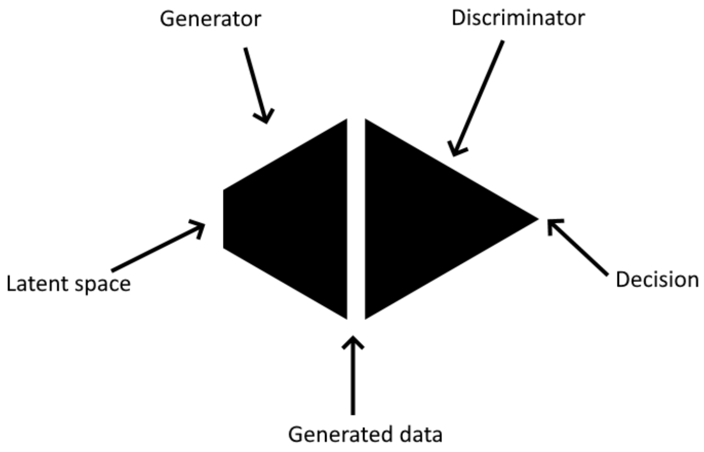

Figure 2.6: A simplified illustration of the structure of a generative adversarial network.

In practice, GANs are prone to some training stability issues such as mode collapse and convergence failure. Mode collapse occurs when the generator finds a niche that it fools the discriminator with. This niche may only cover a small subsection of the data's probability distribution, but the generator has no incentive to change. If adversarial training is an arms race then convergence failure is typically caused by the discriminator becoming too superiorly armed. The gap becomes too large and the improvement of the generator stops. The opposite is also possible, if the discriminator is performing too poorly the generator will not learn anything either. There are many proposed ways of improving the stability of training. Two highlights are feature matching loss and label smoothing [23]. When performing feature matching we change the objective of the generator slightly. Instead of aiming to fool the discriminator, it aims to increase the similarity of the discriminator's internal representation of the fake data to that of the real data. Label smoothing changes the target label of the discriminator. Instead of e.g. denoting real data as 0, and fake data as 1, it adds noise to the labels. The real data labels may instead fall randomly in the range $[0, 0.1]$ and fake data labels $[0.9, 1]$. These techniques can be used in conjunction. For instance, the generator may use feature matching loss while the discriminator uses label smoothing.

2.6.4 Wasserstein generative adversarial networks

A model related to the traditional GAN is the Wasserstein GAN (WGAN) [24]. When reasoning about generative models, one can think of the space of possible outputs as probability distributions, with some outputs being more likely or normal than others. Another way to phrase the goal of the adversarial training is that the probability distribution of the generator should approach the probability distribution of the real data. Through its gamified formulation, the traditional GAN implicitly minimizes the Jensen-Shannon (JS) divergence [22], a measure of the similarity between the two probability distributions. More specifically, the JS divergence is based on how likely the different probability distributions (data distribution, generator) are at producing the observations (fake data). Wasserstein GANs replace this similarity measure with Wasserstein distance (alt. Earth mover's distance). This measures how much the probability distributions have to be altered to be made equal. The main benefit of this is increased training stability. There is also a type of WGAN commonly called WGAN-GP [25]. This version of WGAN changes the training procedure by replacing gradient clipping with a gradient magnitude penalty. The change greatly improves the ability of WGANs to learn the underlying data distribution. When this study refers to WGAN, it is WGAN-GP that is meant.

2.7 Related work

The key works that this study rests on are a variety of (mostly) data-agnostic models. The first key work is regular autoencoders for outlier detection [9]. As mentioned earlier, autoencoders work by reducing the input variables to a smaller dimensionality, followed by restoring the original input to the best of its ability. This act of compression forces the model to learn more efficient representations of the data. These representations are specific to the input data which means that out-of-distribution or more deviating data is typically harder for the autoencoder to generalize to, and therefore to represent and reconstruct. Exploiting this property, autoencoders for outlier detection use the reconstruction error as an anomaly score. A large advance was the AnoGAN model which used GANs instead of autoencoders [26]. The model used latent space differences to score anomalies. However, it was impractically slow since it used iterative backpropagation to infer the location of the data in latent space. Iterative backpropagation gradually tweaks the input noise to the generator to recreate real data as well as possible. The speed issues were improved upon with the BiGAN-based model EGBAD [10, 27] which not only trained a generator and discriminator but an encoder that learned a mapping to the latent space. This was significantly faster and better performing than the AnoGAN method. One interesting off-shoot was the GANomaly model which came shortly after EGBAD. GANomaly is a unique model since it uses autoencoding as a subroutine in an adversarial training paradigm. Regular GANs mix real data with generated data, GANomaly used real data and the same data but autoencoded. As mentioned earlier, it is difficult for the autoencoder to encode out-of-distribution data. In GANomaly, the discriminator is repurposed to spot these issues. After this, a new variant of AnoGAN was released called fast AnoGAN (f-AnoGAN) [6]. This model is WGAN-based and now trains an encoder separately to the model. By contrast, EGBAD trains the encoder jointly. Lastly, the creator of GANomaly presented Skip-GANomaly [7]. The major architectural change of Skip-GANomaly was its skip-connected autoencoder. This design was inspired by U-Net [28], an autoencoder model designed for biomedical image segmentation. The skip-connected autoencoder differs from the regular autoencoder in that it connects layers on both sides of the bottleneck layer (first to last layer, second to second last, etc). It has not been shown if this results in better performance on non-image data, the main motivation for including its predecessor in the comparison as well.

Chapter 3

Methods

3.1 Data

The data used in this example are blood tests provided by the preventative healthcare company Werlabs. This dataset contains blood tests from patients in the time period 2014-2022. There is a large set of different health markers, but patients typically measure just a subset of these, according to wants and needs. The dataset is reduced to two non-health marker parameters and five cardiovascular health markers. These are age and gender, along with HDL-cholesterol, LDL-cholesterol, triglycerides, fasting plasma glucose, and hemoglobin A1c. As proxies for cardiovascular health, they are strongly related [18]. All known errors have been excluded from the data. This is because errors are simulated instead, which provides a greater degree of control. Keeping the errors would interfere with this simulation.

3.1.1 Data normalization

When working with neural networks, data has to be presented in a numerical form. Preferably in similar magnitudes, something that makes it easier for the models to give them similar importance. Gender, being a categorical variable, is either encoded as 0 for females, or 1 for males. Age is zero-centered by subtracting the mean and dividing by the standard deviation. Health markers are treated similarly. Depending on the reporting lab and the equipment they use, health markers are reported in varying units of measurement. This makes it harder, or possibly unreliable to normalize based on dataset mean and standard deviation. However, every data point has an associated reference range. Utilizing this, the data is subtracted by the middle of the reference range, which makes the mean roughly zero. The data is then divided by half the span of the reference range, something that approximates the standard deviation.

3.1.2 Data usage and evaluation data

To enhance the learning of the data, gaussian noise is added to the blood test parameters. This noise is a zero-centered gaussian with a standard deviation of 0.01. Some labs report few significant digits on blood tests, so this makes the possible health marker values less discrete. Before being used in the models, data points are put into fixed-size windows together with other blood tests. Due to the tendency of analytical errors in blood tests to come in groups, this is thought to aid error detection. The blood tests in each window are chosen at random with no replacement. This vastly increases the number of possible data points compared to using a sliding or jumping window which is $O(n)$ where n is the number of data points, whereas random sampling is $O(n^w)$ where w is the window size. This works since there is only a time-dependent component present in blood tests with measurement errors, something that is rare and eliminated as well as possible in the dataset.

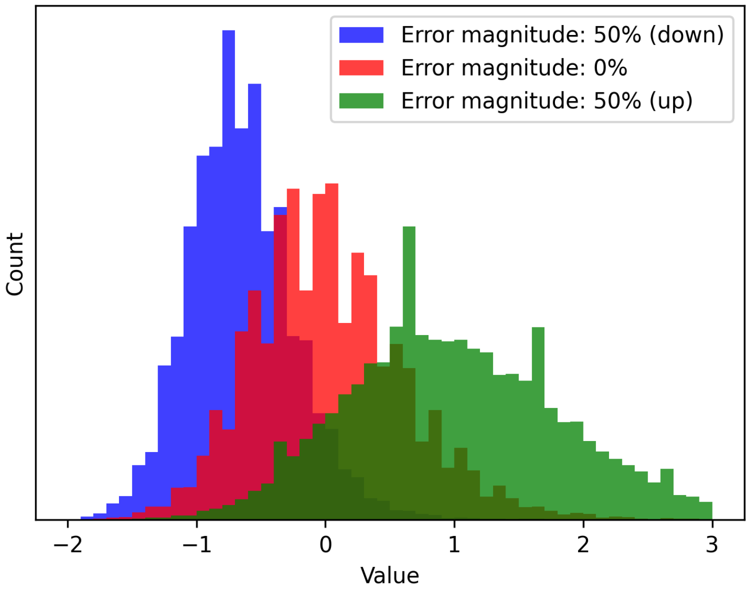

Figure 3.1: Examples of how error simulations affect the distribution of LDL health markers.

For evaluation purposes, errors are added to each window in the evaluation data with a probability of 50%. These errors are added at runtime rather than beforehand. Windows are during this process labeled as either errors or normal data, which is what makes evaluation possible. Errors are added by taking a window and multiplying or dividing one of its health marker columns by an error factor. Examples of this can be seen in Figure 3.1. This error factor is chosen uniformly at random from an error range. For supervised models, this process is also done during training. For testing, this is also done at runtime, but with a fixed seed for the pseudorandom number generator, this is what makes the testing consistent and replicable. The data split was 80% training data, 10% validation data, and 10% test data.

3.2 Implementation

To standardize the implementations the models were created out of the same building blocks. Encoders consisted of an input layer, followed by three hidden layers with a decreasing amount of neurons, and then the bottleneck layer (output layer). The hidden layer sizes were based on the input size. The first hidden layer was 75% larger than the input size, the second 50% larger, and the third was 25% larger. Decoders were reversed copies of the encoders. Generators simply used the decoder structure, and discriminators were encoders with a one-neuron output layer instead of the bottleneck layer. The sizes of the bottleneck layers and generator noise/latent spaces were 40% of the input sizes (rounded to the nearest whole number). This corresponds to 1.2 latent variables per blood test for the naive models, and 2.8 latent variables per blood test for the full models.

ReLU was the choice of activation function. This was applied to all layers except the bottleneck and output layers. All discriminators and supervised classifiers used sigmoid activations for the output layer, except for f-AnoGAN which had no output activation function. The supervised models used binary cross-entropy loss, the losses of other models were implemented as suggested in their respective papers. All GAN-based models except f-AnoGAN used label smoothing for the anomaly side of the output, f-AnoGAN used no label smoothing. All models used the Adam optimizer with a warmup period of 20 epochs. During this warmup the learning rate increased linearly from 0.000001 to 0.001. The models were trained with a patience of ten epochs, meaning that the training terminates after no improvement has been seen for seen after ten epochs. The best model weights seen during training were restored after training.

Notably, any convolutional layers in the models were replaced by dense layers in this study. This is perhaps the most significant deviation in the model reimplementation process. The choice to make this deviation is motivated by the data, which has no topology, the primary reason to use convolution. The implications of this choice is unknown. Perhaps most significantly affected is Skip-GANomaly, whose U-Net-inspired architecture may not necessarily translate to situations with dense layers. Recall that the purpose of U-Net is to improve image segmentation specifically through skip-connections in the autoencoder.

3.2.1 Univariate and focused models

There are two univariate classifiers that look at one health marker at a time. One of the models is based on GANomaly, and the other is based on the supervised approach. These are not true univariate classifiers since they still have access to the non-health marker data. The justification for this is two-fold. Firstly, we are primarily interested in the potential benefit of looking at multiple health markers at once. The model choice did not hinder this. Secondly, age and gender are important explanatory factors for any set of observed health markers. The exclusion of such variables would severely handicap the models and make them less informative as comparisons to the full models. Lastly, there was a 'focused' variant of the supervised model. This variant has access to all health markers but was only trained to recognize errors in one of those health markers. The purpose of this was to bridge the gap between the univariate and multivariate models. Without this bridge, it is hard to control for the fact that the univariate models have the benefit of only having to focus on a single health marker, and the full models have to consider many possible error sources. During the evaluation, errors were only simulated on the health marker in focus. This is a factor that separates the test environment from the real world for the focused supervised model. In the real world, there could be errors in other health markers that set false expectations for the health marker in focus, leading to false positives.

3.3 Training, testing, and evaluation

The outlier detection models were trained on clean data without any errors. This is necessary since their outlier detection ability is dependent on them not learning the error distribution properly. By contrast, the supervised models were trained using an error rate of 50%. All models were evaluated on labeled data, a seemingly paradoxical choice given that the outlier detection models are self-supervised. However, compared to many other labeled datasets, there is no human assistance needed for the labeling. This effectively makes the label cost zero and does not increase the amount of labor required, one of the primary benefits of using self-supervised methods.

At the end of every epoch, the model performance was assessed on the labeled validation data with simulated errors. Models performed their anomaly scoring function, and from the resulting anomaly scores, AUC-PR was measured. After training a model, the version with the best AUC-PR score was restored and used for testing. The exception to this rule is f-AnoGAN, which cannot perform anomaly scoring before the encoder is trained. For this reason, the generator loss was used instead. This is sound because f-AnoGAN is a WGAN-based model, and in such a model the loss is more directly correlated to the convergence of the generator. The error rate used for validation was 50%. This is not a realistic error rate, but is a good choice since this minimizes the impact of chance on model performance.

The final testing is performed in the same manner as on the validation data, but using an error rate of 0.1%. This error rate is unlike the validation error rate chosen for realism. The test performance was evaluated using three main metrics, namely AUC-ROC, AUC-PR, and F1-score. Precision and recall are also measured and are chosen to be the values that together contribute to the maximum F1-score. These are not used for ranking the models but are there to give context as to how a model balances precision and recall to achieve its F1-score.

Chapter 4

Results

4.1 Guiding experiment

The purpose of the main and guiding experiment was to investigate the utility of the models in settings approximating the real world. Using error simulation on real data it is possible to provide a roughly realistic environment in terms of error magnitude and frequency. The parameters of this experiment were a window size of 8 (each model looks at eight blood tests at once), an error magnitude of 20-50%, and an error rate of 0.1%. In the results (shown in Table 4.1), the Kruskal-Wallis test showed that at least two of the groups differed in each metric ($p < 0.001$, $N = 25$). The experiment shows that the supervised approach outperforms other approaches by a large margin on the error detection task ($p < 0.042$, $N = 25$ for all comparisons on each metric). The autoencoder outperformed the GAN-based models on all metrics ($P < 0.041$, $N = 25$ for all comparisons on each metric) Among the GAN-based outlier detection models, f-AnoGAN performed the best on AUC-ROC ($P < 0.002$, $N = 25$ for all comparisons). There were no significant differences between their F1 measurements. There was no significant performance difference between GANomaly and Skip-GANomaly on any of the metrics.

Figure 4.1 shows a clear difference in distribution between normal and anomalous data as classified by the supervised model. There is some overlap between the long tails of the distributions, but most of the mass is well separated.

Table 4.1: Test results of the main models using parameters window size 8, error magnitude 20-50%, and error rate 0.1%. The highest results are highlighted in bold. Results are the average of 25 runs. The baseline performances of a dummy classifier is 0.5 on AUC-ROC, 0.001 on AUC-PR, and ~0.002 on F1.

Model	AUC-ROC	AUC-PR	F1	Precision	Recall
Autoencoder	0.74 ± 0.01	0.01 ± 0.00	0.03 ± 0.01	0.02 ± 0.01	0.10 ± 0.02
EGBAD	0.58 ± 0.05	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	0.06 ± 0.03
f-AnoGAN	0.68 ± 0.01	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.12 ± 0.07
GANomaly	0.63 ± 0.03	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.00	0.06 ± 0.02
Skip-GANomaly	0.60 ± 0.02	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.07 ± 0.02
Supervised	$\mathbf{0.91} \pm 0.01$	$\mathbf{0.14} \pm 0.03$	$\mathbf{0.33} \pm 0.04$	0.29 ± 0.06	0.38 ± 0.02

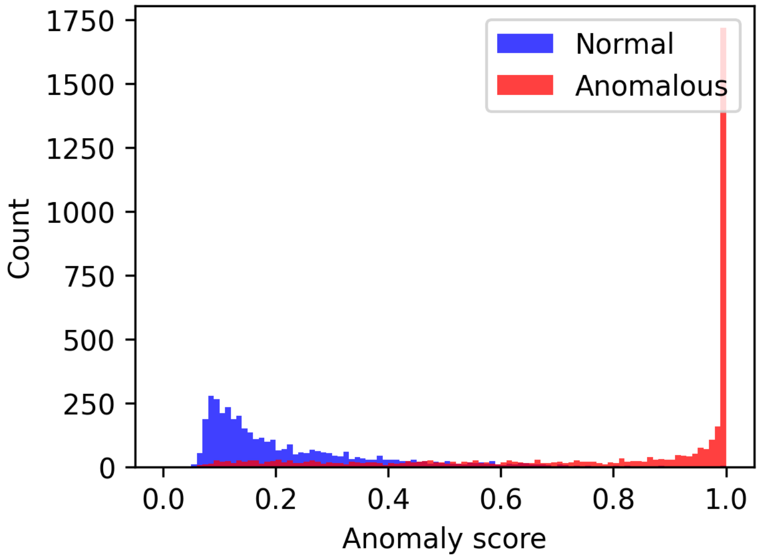

Figure 4.1: Histogram showing the distribution of anomaly scores produced by the Supervised model for the two classes, normal and data with simulated errors. The error rate is set to 50% for visualization purposes.

4.2 Univariate versus multivariate analysis

Complementing the regular experiment we evaluate the univariate and focus model performances. This ablation has the purpose of revealing the contribution of supporting health markers to model performance. The experiments were performed with parameters set to window size 8, error magnitude 20-50%, and error rate 0.1%. The results are shown in Table 4.2, and at least two of the groups differed ($p < 0.001$, $N = 25$). GANomaly-naive significantly outperforms GANomaly in AUC-ROC and AUC-PR ($p < 0.02$, $N = 25$), but it did not quite reach significance in F1. The supervised, focused models significantly outperformed their regular counterpart ($p < 0.02$, $N = 25$ for all metrics). However, when the models Supervised-naive and Supervised-focused were compared to each other, they only significantly differed in AUC-ROC ($p < 0.02$, $N = 25$).

Table 4.2: Test results of the focused models and their regular counterparts using parameters window size 8 and error magnitude 20-50%. The highest results are highlighted in bold. Results are the average of 25 runs.

Model	AUC-ROC	AUC-PR	F1	Precision	Recall
GANomaly	0.63 ± 0.02	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.07 ± 0.03
GANomaly-naive	0.70 ± 0.02	0.01 ± 0.00	0.03 ± 0.01	0.02 ± 0.01	0.11 ± 0.02
Supervised	0.91 ± 0.01	0.14 ± 0.06	0.32 ± 0.06	0.30 ± 0.11	0.39 ± 0.02
Supervised-focused	$\mathbf{0.95} \pm 0.00$	$\mathbf{0.37} \pm 0.06$	$\mathbf{0.50} \pm 0.03$	0.47 ± 0.05	0.56 ± 0.02
Supervised-naive	0.94 ± 0.00	$\mathbf{0.37} \pm 0.05$	0.47 ± 0.03	0.46 ± 0.04	0.52 ± 0.01

Figure 4.2 shows the difference in classification ability between the Supervised-naive and Supervised-focused model per health marker. This analysis, like the overall analysis above, shows no major performance differences between the two models. None of the health markers differed significantly on all metrics.

4.3 Analysis of single blood tests

In this section, the potential regularizing effect of window size is investigated. This is done by looking at a single blood test instead of many blood tests at once. If there is a regularizing effect it should be evident from a decrease in model performance when there are fewer blood tests to look at.

Experiments using the default error magnitude and rate, and window size 1 (single blood tests) show the importance of the window size. All models were struggling to find errors. In the results (shown in Table 4.3), at least two of the groups differed ($p < 0.001$, $N = 25$). Out of the non-focused models, the supervised model scores the highest in all metrics except AUC-

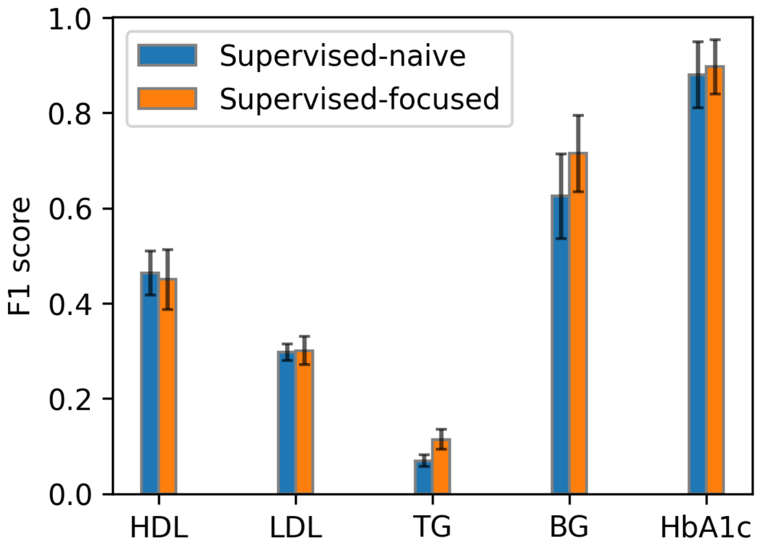

Figure 4.2: The difference in f-score per health marker for the supervised models with focus. Based on results from experiments with window size 8. The health markers in order from left to right, HDL, LDL, triglycerides, blood glucose, hemoglobin A1c. Results are the average of 25 runs.

Table 4.3: Test results of the main models using parameters window size 1, error magnitude 20-50%, and error rate 0.1%. The first set of rows are models without focus, the second set are those with focus. The highest results are highlighted in bold. Results are the average of 25 runs.

Model	AUC-ROC	AUC-PR	F1	Precision	Recall
Autoencoder	0.69 ± 0.01	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	0.08 ± 0.04
EGBAD	0.56 ± 0.03	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.06 ± 0.03
f-AnoGAN	0.60 ± 0.04	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.09 ± 0.04
GANomaly	0.56 ± 0.02	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.05 ± 0.04
Skip-GANomaly	0.57 ± 0.02	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.04 ± 0.03
Supervised	$\mathbf{0.70} \pm 0.01$	$\mathbf{0.01} \pm 0.00$	$\mathbf{0.05} \pm 0.01$	0.03 ± 0.00	0.10 ± 0.02
GANomaly-naive	0.61 ± 0.03	0.00 ± 0.00	0.02 ± 0.01	0.03 ± 0.06	0.08 ± 0.04
Supervised-focused	$\mathbf{0.77} \pm 0.01$	$\mathbf{0.03} \pm 0.00$	$\mathbf{0.08} \pm 0.00$	0.05 ± 0.00	0.17 ± 0.02
Supervised-naive	0.75 ± 0.01	0.01 ± 0.00	0.04 ± 0.01	0.03 ± 0.01	0.14 ± 0.03

ROC ($p < 0.04$ for all F1 and AUC-PR measurements). The supervised model performed significantly better than the GAN-based models in AUC-ROC

($p < 0.001$, $N = 25$), but did not significantly outperform the autoencoder. The GAN-based outlier detection had next to no differentiation in performance as measured by AUC-PR and F1. GANomaly-naive outperformed its parent model in all metrics ($p < 0.04$, $N = 25$ for all metrics). By contrast, Supervised-naive only significantly outperformed its parent model in AUC-ROC ($p < 0.02$, $N = 25$). Unlike the experiments done on window size 8, the multivariate focused model scores notably higher than the univariate classifier on all metrics ($p < 0.02$, $N = 25$ for all metrics).

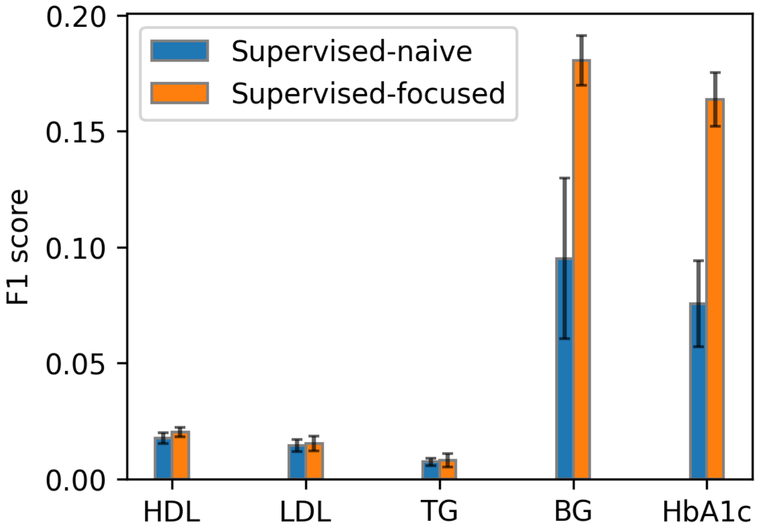

Figure 4.3: The difference in f-score per health marker for the supervised models with focus. Based on results from experiments with window size 1. The health markers in order from left to right, HDL, LDL, triglycerides, blood glucose, hemoglobin A1c. Results are the average of 25 runs.

Figure 4.3 shows the difference in classification ability between the Supervised-naive and Supervised-focused model per health marker. Unlike the previous analysis per health marker, this analysis shows clear performance benefits of using supporting health markers with blood glucose and hemoglobin A1c ($p < 0.001$, $N = 25$ for all metrics). There is also a clear, but small effect for HDL ($p < 0.001$, $N = 25$ for all metrics). There seems to be no effect for the other health markers.

4.3.1 Progressive increases in window size

The previous experiment showed that window size plays an important role in model performance. As part of the investigation into the window size parameter, it is therefore reasonable to complement this result by estimating the performance at a variety of window sizes. Such an investigation would be a good guide for practical choices of window size.

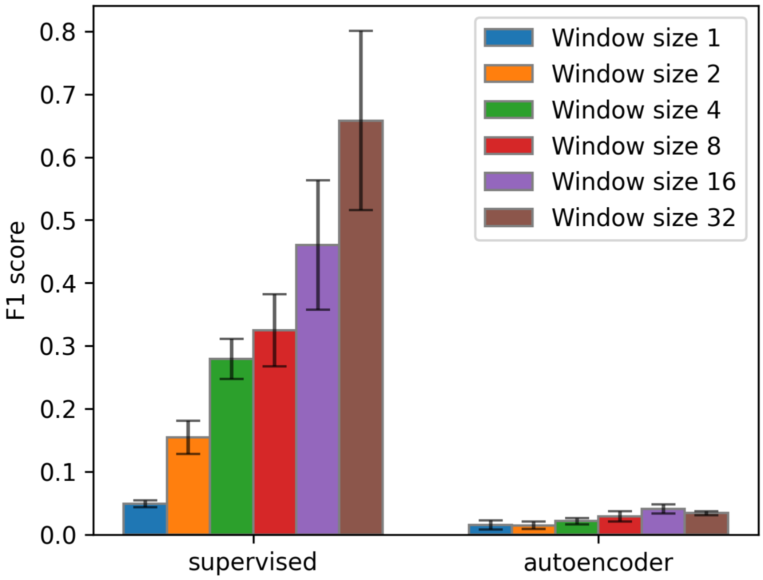

Figure 4.4: The progression of F1-scores as the window size is increased. Error magnitude is 20-50% and error rate 0.1%. Results are the average of 25 runs.

Table 4.4: Correlation coefficients between the window size and model performance calculated for the supervised and autoencoder models. The parameters were window size 1, 2, 4, 8, 16, and 32, as well as error magnitude 20-50%, and error rate 0.1%. Results are the average of 25 runs.

Model	AUC-ROC	AUC-PR	F1
Autoencoder	0.897	0.855	0.763
Supervised	0.814	0.998	0.952

In Figure 4.4 it can be seen that the supervised model experiences large jumps in performance with every doubling of the window size. The increases autoencoder are minor in comparison. The figure clearly showcases the difference in feasibility of using the supervised versus autoencoder model for error detection. Other models were not included in the figure since the autoencoder was the second best performing model and their addition would not be informative. The calculated Pearson's correlation coefficients between window size and model performance are given in Table 4.4 for all metrics. Out of these correlations, the supervised model reaches significance on all metrics ($p < 0.05$, $N = 25$ for all metrics). The autoencoder reaches significance on AUC-ROC and AUC-PR ($p < 0.03$, $N = 25$ for both metrics).

4.4 With less discrete errors

Comparisons between outlier detection models are hard when they are all struggling. One way to remedy this is to make outlier detection easier by increasing the magnitude of the errors.

Table 4.5: Test results of the main models using parameters window size 8, error magnitude 20-50%, and error rate 0.1%. The highest results are highlighted in bold. Results are the average of 25 runs.

Model	AUC-ROC	AUC-PR	F1	Precision	Recall
Autoencoder	$\mathbf{0.91} \pm 0.01$	$\mathbf{0.05} \pm 0.01$	$\mathbf{0.18} \pm 0.03$	0.12 ± 0.03	0.36 ± 0.04
EGBAD	0.79 ± 0.02	0.01 ± 0.01	0.07 ± 0.03	0.05 ± 0.02	0.17 ± 0.06
f-AnoGAN	0.85 ± 0.01	0.01 ± 0.00	0.07 ± 0.01	0.05 ± 0.00	0.17 ± 0.01
GANomaly	0.80 ± 0.03	0.01 ± 0.01	0.08 ± 0.03	0.05 ± 0.02	0.18 ± 0.04
Skip-GANomaly	0.80 ± 0.03	0.01 ± 0.00	0.07 ± 0.01	0.04 ± 0.01	0.17 ± 0.03

The results seen in Table 4.5 are consistent with previous findings. The Kruskal-Wallis test found that at least two of the groups differed ($p < 0.001$, $N = 25$). The regular autoencoder is again the top-performing outlier detection model ($p < 0.009$ for all comparisons on all metrics). None of the other models are very competitive with the autoencoder. Between these, models only differ significantly in AUC-ROC. EGBAD, Skip-GANomaly, and GANomaly could not be signifcantly distinguished, but f-AnoGAN stands out as the second best outlier detection model in this metric.

In Figure 4.5 we see a histogram of the distribution of anomaly scores produced by f-AnoGAN. They are weakly separated for the lower error magnitudes and more clearly separated for larger magnitudes. A similar

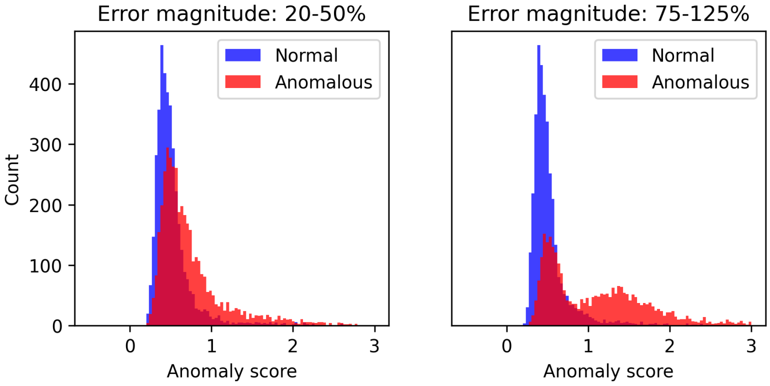

Figure 4.5: Histogram of anomaly score distributions produced by f-AnoGAN on error magnitudes 20-50% and 75-125%, the error magnitude 20-50%. The error rate is set to 50% for visualization purposes.

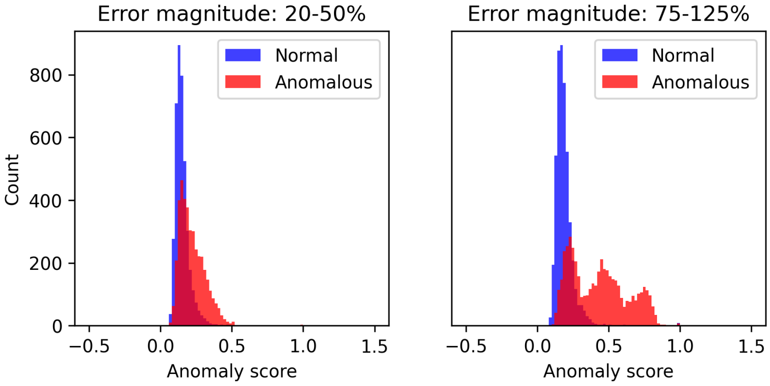

Figure 4.6: Histogram of anomaly score distributions produced by the autoencoder on error magnitudes 20-50% and 75-125%. The error rate is set to 50% for visualization purposes.

pattern can be seen in the distributions produced by the autoencoder as seen in Figure 4.6. Note that a significant portion of the anomalous data are put in a distribution similar to the normal data even for the larger error magnitudes. When scoring, the models will deem these simulated errors as plausible. Interestingly, f-AnoGAN produces a bimodal distribution for the larger error magnitude, and the autoencoder a trimodal distribution.

Chapter 5

Discussion

5.1 Key findings

The study aimed to answer the question of which neural network-based models were most suitable for detecting errors in blood tests. The candidate models were the autoencoder, EGBAD, f-AnoGAN, GANomaly, Skip-GANomaly and a regular supervised model. All self-supervised models were trained on clean data to learn its probability distribution, and the supervised model was trained using error simulation (based on real data). Across all parameters and metrics, the supervised model and its derivatives outperformed other approaches. There was typically a large performance gap between the supervised approach and the next best performing approach, the autoencoder. Similarly, a clear discrepancy could be observed between the autoencoder and GAN-based models, but the magnitude of the separation was smaller. There was also a significant correlation between model performance and window size, especially for the supervised model. Recall that the window size represents how many blood tests each model considers at once. The positive correlation between model performance and window size means that it is important to let the model use multiple blood tests as references for error detection.

5.2 Other findings and explanations

Notably, there was a major performance advantage of the supervised model compared to the outlier detection models. Explaining this discrepancy, the main factor appears to be specificity. A key difference between outlier detection models and the supervised model is that the latter can be explicitly taught to have a singular focus on collective outliers since its training data is labeled and includes errors. By contrast, outlier detection models can only look for outliers in general because their training data does not include any errors. In light of this, the small effect of error simulation on anomaly scores as shown in Figure 4.6 versus the large ones seen in Figure 4.1 is not as surprising. Ideally, for error detection, the model would account for the (un)likelihood of having multiple anomalous values along the same health markers. However, in the training data for the outlier detection models, health markers from different blood tests never depend on each other since analytical errors are never introduced. Moreover, due to data normalization, health markers are also not related to each other by magnitude and reference ranges. This effectively blocks the outlier detection models from learning which health markers are of the same type in every blood test and explains their ineffectiveness at error detection. Due to the inability to perceive collective outliers the outlier detection models become more sensitive to other types of outliers. Combine this with the fact that analytic errors only are a subset of the complete outlier distribution (e.g. various disease states), and it is clear why the outlier detection models did not succeed. Conversely, a major benefit for the supervised model is that analytic errors are low in complexity. They only affect a well-defined portion of all input variables (health markers of the same type), and the errors all manifest the same way (as a constant multiplicative increase or decrease of all affected health markers). This increases the feasibility of a supervised approach, but does not benefit the other models that are not taught to look for this specific signature.

Throughout the experiments, there was a discrepancy between the scores on AUC-ROC and those produced by AUC-PR and F1. Specifically, the outlier detection models perform better on AUC-ROC than the latter metrics. This is possible because increasing specificity positively affects AUC-ROC but not necessarily other metrics. The performance difference is consistent with the function of the outlier detection models. Increases in specificity correspond to a model becoming better at telling whether a data point is normal, which is the sole training objective of the outlier detection models. To improve the other metrics, the model still needs to differentiate errors from regular outliers, which it is not trained to do. Thus, AUC-ROC is more easily improved than the other metrics.

The multivariate, focused models consistently performed slightly better than their univariate counterparts. On larger window sizes (as seen in Figure 4.2) the performance increase was small, indicating that the information gain was also small. However, error detection in blood glucose and hemoglobin A1c seemed to be greatly aided by a multivariate approach in experiments with window size 1 (as seen in Figure 4.3). Based on the error detection ability increasing with window size, but the impact of supporting health markers decreasing, it seems that the need to use health marker correlations in a multivariate approach decreases with the number of reference blood tests.

The experiments clearly show that window size is a powerful regularizer for error detection. It is almost impossible to detect errors when the model considers just a single blood test. As the number of data points in the window increases, error detection gets easier. The regularizing effect of window size is more pronounced in the supervised models, but also weakly present in the outlier detection models (as seen in Figure 4.4). The window size may not have the same regularizing effect for outlier detection models because they are not taught to exploit the fact that errors come in groups. This is consistent with the larger performance increases seen in the supervised model per increase in window size, compared to the increases seen in the autoencoder.

5.3 Ethical considerations

A primary concern when providing data-driven services is fairness. Unfairness is often driven by biases in the data or methods. For the full supervised models, unhealthy patients with uninformative health markers may not be as likely to have errors in blood tests caught. This is due to the information gained from using other health markers as support. For the outlier detection models, the opposite is true. The experiment results suggest that more regular blood tests will get lower anomaly scores. Already anomalous blood tests may become even more anomalous when coupled with errors, making them more likely to be caught. The former issue is a larger concern since outlier detection models do not seem suited for error detection in blood tests. Regularization in the form of larger windows somewhat addresses this, as that makes the emphasis on individual blood tests smaller. Building on this idea, models that are not bound to fixed window sizes and streams through the data (like recurrent neural networks [13]) can potentially be a way to detect errors earlier and more efficiently. Streaming models in this context are like models with large window sizes, a property that has proven beneficial in the experiments.

There are also some ethical considerations to take into account regarding the patients' choice of health markers to measure. Firstly, some health markers are measured more rarely than others. If the volume is not enough, there may not be enough measurements to catch errors in time. Patients who choose to have the rarer health markers measured are therefore at greater risk of errors. Lastly, when the better models are the models that use more health markers, patients who test fewer health markers are at a disadvantage. One possible way to tackle this would be to always measure all health markers.

5.3.1 Sustainability and societal impact

The primary societal contribution of this study is that it is a step towards more reliable blood test results. Early detection of measurement errors is essential to avoid producing more of them. As blood tests sometimes motivate medical or lifestyle interventions, they must be correct. Otherwise, these interventions may have unintended consequences. Therefore, by preventing errors from reaching unknowing physicians, this study contributes to the progress of the UN's 3rd Sustainable Development Goals, the promotion of health and well-being [29].

5.3.2 Scientific contribution

One methodological weakness is that the supervised model is not evaluated on real data, instead, it is both evaluated and trained on simulated data. This leaves open questions about how closely the simulated data match real data, and how well the model performance translates to real-world settings. Moreover, in contrast to the self-supervised models, the supervised model trained with the advantage of access to examples of (simulated) errors. Its dominance is therefore not necessarily surprising, but it is informing all the same. On the one hand, the outcome, although expected, was not certain. Secondly, the magnitude of the differences in performance could not have been predicted in advance. Compared to before experiments began, bridging the realism gap between real and simulated errors now seems more fruitful for further improving real-world error detection than continuing to explore self-supervised approaches.

Another unexpected result is the poor performance of the GAN-based models compared to the autoencoder. This stands in direct contrast to what is supposed to be (or be candidates for) state-of-the-art. Attribution is difficult but the cause of this poor performance could plausibly stem from replacing convolution by dense layers, properties of the data, or the specific task (error detection).

Chapter 6

Conclusions

Primarily, the experiments show that outlier detection models are not suitable detecting errors relative to other alternatives. Supervised methods were better performing in all chosen metrics. Disregarding their impracticality, regular autoencoders were best at the task of the outlier detection models. The GAN-based models could not meaningfully outperform each other. Consequently, there was not any significant performance difference between GANomaly and its successor Skip-GANomaly, either. It therefore seems that the skip-connected autoencoder in the newer model is not an advantage on non-spatial data. Moreover, the results show that a multivariate approach has at least some advantage over a univariate approach. It is also confirmed that window size, the amount of blood tests a model looks at, is an important regularizer for error detection in blood tests. Increases in window size yield large increases in error detection ability. Given the effectiveness of the supervised model there is reason to further study and potentially employ deep learning-based error detection to lower the risk of errors.

6.1 Future work

While the results of the experiments do not support the hypothesis that outlier detection models are suitable for analytical error detection, there are other plausible applications for outlier detection models using the same data. Specifically, future work could investigate the use of outlier detection models for finding unspecified disease states (rather than diagnosing specific diseases), hemolysis errors, and as physician support tools. Using outlier detection models as support tools could look like using anomaly scores as blood test complexity measurements for workload management, or as signals to look out for disease states. With interpretability methods like SHAP [30] or DeepLIFT [31], it could be possible to find the contributions of individual health markers to anomaly scores.

Supervised models were the dominating method of error detection in the experiments. This warrants further investigation into techniques to improve error detection performance further. One such technique would be to turn the classification problem into a regression problem by having the model estimate the error magnitude, either in conjunction with or replacing the binary classification label. Since the errors are simulated, the error magnitude is available for use as a label. Such a choice could make for a more powerful supervisory signal since it contains more information. The supervised model may also benefit from, at least in terms of interpretability, making the task an n-way classification (or regression) task so that it identifies which health marker has been affected.

A gap in the experiments is that the models are only trained and tested on windows completely filled with errors. It is not yet known how the models would perform on partially filled windows. This is a sub-theme of the general point mentioned in section Scientific contribution 5.3.2, the need to explore the gap between real and simulated data.

The results of this study clearly demonstrates the performance benefits of looking at multiple blood tests at once. By similar reasoning, looking at the anomaly scores of multiple windows close in time could aid error detection. This is similar in spirit to increasing the window size since more blood tests are included in such an analysis. Alternatively, using a recurrent neural network (as suggested in section 5.3) could also mimic increasing the window size. The model could output an anomaly score per data point. Such a model could be trained by simulating errors, but adding the error labels to individual data points, rather than windows.

Further improvements in outlier detection could likely be achieved by training an ensemble of models. This could either be done by separately training several instances of the same model. GANs can also be trained by having a pool of generators and a pool of discriminators that are continuously paired at random during training. One could also combine different models into one ensemble.

References

[1] D. Broadhurst, R. Goodacre, S. N. Reinke, J. Kuligowski, I. D. Wilson, M. R. Lewis, and W. B. Dunn, "Guidelines and considerations for the use of system suitability and quality control samples in mass spectrometry assays applied in untargeted clinical metabolomic studies," *Metabolomics*, vol. 14, no. 6, p. 72, Jun. 2018. doi: 10.1007/s11306-018-1367-3

[2] K. Chrominski and M. Tkacz, "Comparison of outlier detection methods in biomedicaI data," *Journal of Medical Informatics & Technologies*, vol. 16/2010, Jan. 2010.

[3] J. de Souza Gaspar, E. Catumbela, B. Marques, and A. Freitas, "A Systematic Review of Outliers Detection Techniques in Medical Data - Preliminary Study," in *Proceedings of the International Conference on Health Informatics*. Rome, Italy: SciTePress - Science and and Technology Publications, 2011. doi: 10.5220/0003168705750582. ISBN 978-989-8425-34-8 pp. 575–582. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0003168705750582

[4] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Deep Learning for Medical Anomaly Detection – A Survey," *ACM Computing Surveys*, vol. 54, no. 7, pp. 1–37, Sep. 2022. doi: 10.1145/3464423

[5] G. Gunčar, M. Kukar, M. Notar, M. Brvar, P. Černelč, M. Notar, and M. Notar, "An application of machine learning to haematological diagnosis," *Scientific Reports*, vol. 8, no. 1, p. 411, Dec. 2018. doi: 10.1038/s41598-017-18564-8

[6] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-AnoGAN: Fast unsupervised anomaly detection with

generative adversarial networks," *Medical Image Analysis*, vol. 54, pp. 30–44, May 2019. doi: 10.1016/j.media.2019.01.010

[7] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-GANomaly: Skip Connected and Adversarially Trained Encoder-Decoder Anomaly Detection," *arXiv:1901.08954 [cs]*, Jan. 2019, arXiv: 1901.08954.

[8] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training," *arXiv:1805.06725 [cs]*, Nov. 2018, arXiv: 1805.06725.

[9] M. Sakurada and T. Yairi, "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis - MLSDA'14*. Gold Coast, Australia QLD, Australia: ACM Press, 2014. doi: 10.1145/2689746.2689747. ISBN 978-1-4503-3159-3 pp. 4–11. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2689746.2689747

[10] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-Based Anomaly Detection," *arXiv:1802.06222 [cs, stat]*, May 2019, arXiv: 1802.06222.

[11] X. Han, X. Chen, and L.-P. Liu, "GAN Ensemble for Anomaly Detection," *arXiv:2012.07988 [cs]*, Dec. 2020, arXiv: 2012.07988.

[12] Y. Celik, K. Sabanci, A. Durdu, and M. F. Aslan, "Breast Cancer Diagnosis by Different Machine Learning Methods Using Blood Analysis Data," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 6, no. 4, pp. 289–293, Dec. 2018. doi: 10.18201/ijisae.2018648455

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[14] S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan, and A. W. Muzaffar, "Image Colorization: A Survey and Dataset," *arXiv:2008.10774 [cs, eess]*, Jan. 2022, arXiv: 2008.10774.

[15] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," *arXiv:1901.03407 [cs, stat]*, Jan. 2019, arXiv: 1901.03407.

[16] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *The Journal of Physiology*, vol. 160, pp. 106–154, Jan. 1962. doi: 10.1113/jphysiol.1962.sp006837

[17] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” *arXiv:1311.2901 [cs]*, Nov. 2013, arXiv: 1311.2901.

[18] H.-R. Park, S.-R. Shin, A. L. Han, and Y. J. Jeong, “The Correlation between the Triglyceride to High Density Lipoprotein Cholesterol Ratio and Computed Tomography-Measured Visceral Fat and Cardiovascular Disease Risk Factors in Local Adult Male Subjects,” *Korean Journal of Family Medicine*, vol. 36, no. 6, pp. 335–340, Nov. 2015. doi: 10.4082/kjfm.2015.36.6.335

[19] F. Di Mattia, P. Galeone, M. De Simoni, and E. Ghelfi, “A Survey on GANs for Anomaly Detection,” *arXiv:1906.11632 [cs, stat]*, Sep. 2021, arXiv: 1906.11632.

[20] H. R. Sofaer, J. A. Hoeting, and C. S. Jarnevich, “The area under the precision-recall curve as a performance metric for rare binary events,” *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 565–577, Apr. 2019. doi: 10.1111/2041-210X.13140

[21] D. Freedman, R. Pisani, and R. Purves, “Statistics (international student edition),” *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Networks,” *arXiv:1406.2661 [cs, stat]*, Jun. 2014, arXiv: 1406.2661.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved Techniques for Training GANs,” *arXiv:1606.03498 [cs]*, Jun. 2016, arXiv: 1606.03498.

[24] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” *arXiv:1701.07875 [cs, stat]*, Dec. 2017, arXiv: 1701.07875.

[25] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved Training of Wasserstein GANs,” *arXiv:1704.00028 [cs, stat]*, Dec. 2017, arXiv: 1704.00028.

[26] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery," *arXiv:1703.05921 [cs]*, Mar. 2017, arXiv: 1703.05921.

[27] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial Feature Learning," *arXiv:1605.09782 [cs, stat]*, Apr. 2017, arXiv: 1605.09782.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv:1505.04597 [cs]*, May 2015, arXiv: 1505.04597.

[29] United Nations, "Transforming our world: the 2030 Agenda for Sustainable Development," United Nations, New York, Agenda A/RES/70/1, Oct. 2015. [Online]. Available: https://sdgs.un.org/2030agenda

[30] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *arXiv:1705.07874 [cs, stat]*, Nov. 2017, arXiv: 1705.07874.

[31] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning Important Features Through Propagating Activation Differences," *arXiv:1704.02685 [cs]*, Oct. 2019, arXiv: 1704.02685.

Appendix A

Source of claims

Claims of how physicians typically operate and how they discover error in blood tests are based on interviews with physicians at Werlabs. How error in blood tests typically present, their rarity, the consequences of errors, and desired properties from error detection models are also based on interview with physicians and other personnel at Werlabs. Therefore, no other source is provided for these claims.

For DIVA

```
{
"Author1": {
            "Last name": "Vinell",
            "First name": "Paul",
            "Local User Id": "u100001",
            "E-mail": "vinell@kth.se",
            "ORCiD": "0000-0002-00001-1234",
            "organisation": {"L1": "School of Electrical Engineering and Computer Science ",
                              }
           },
"Degree": {"Educational program": "Master’s Programme, Computer Science, 120 credits"},
"Title": {
            "Main title": "Error detection in blood work",
            "Subtitle": "A comparison of self-supervised deep learning-based models",
            "Language": "eng" },
"Alternative title": {
            "Main title": "Felupptäckning i blodprov",
            "Subtitle": "En jämförelse av självbevakade djupinlärningsmodeller",
            "Language": "swe"
},
"Supervisor1": {
                  "Last name": "Herman",
                  "First name": "Pawel",
                  "Local User Id": "u100003",
                  "E-mail": "paherman@kth.se",
                  "organisation": {"L1": "School of Electrical Engineering and Computer Science ",
                                   "L2": "Computer Science" }
              },
"Examiner1": {
                  "Last name": "Kumar",
                  "First name": "Arvind",
                  "Local User Id": "u100004",
                  "E-mail": "arvkumar@kth.se",
                  "organisation": {"L1": "School of Electrical Engineering and Computer Science ",
                                   "L2": "Computer Science" }
              },
"Cooperation": { "Partner_name": "Werlabs AB"},
"Other information": {
"Year": "2022", "Number of pages": "1,45"}
}
```