CE888 – Assignment 2 – Project 1: Auxiliary feature learning for small dataset regularization (April 2019)

Sylvia Tattersfield Marty – st18580

***Abstract*— Data has a big impact on the outcome of the desired tasks that it creates, one of the biggest issues for machine learning and deep learning is having insufficient data or training examples. While different solutions have been suggested and explored this research paper will focus on a proposed method for using auxiliary feature learning for small dataset regularization. This will be done through the implementation of both an autoencoder and a neural network. The autoencoder will work to learn and create feature selection, these features will then be used to train the neural network which will be a discriminative classifier. The project will be tested in three different datasets, which will be manipulated to include less data and increase the amount of data as part of an experiment to properly evaluate its performance, as the amount of data increases. Lastly the performance will be compared to an SVM classifier.**

***Index Terms*— Autoencoders, Deep Learning, Multi-task Learning, Neural Network, Supervised Learning.**

I. INTRODUCTION

IN the following report a proposed method for using auxiliary feature learning for small dataset regularization is suggested.

The anticipated method will consist of two main parts, an autoencoder used for feature selection and a neural network used for classification. The autoencoder will focus on receiving the input data and making a feature extraction for the representation of the data which will be fed to the neural network as inputs, which become act as a discriminative classifier.

One of the main premises of this project is the impact the amount of data has therefore an important element of the project was the selection manipulation of the datasets,
explained on further sections. It must be mentioned that the project will focus on the difference in performance when utilizing different sizes of datasets, therefore the experiments that will be done will follow a data incrimination plan. Another of the main aspects of the project is the manipulation of the attributes within the input data, through the autoencoder and the impact that this can have on the overall performance of the trained classifier.

The proposed approach is attempting to tackle one of the greatest issues when regarding data science and processing, insufficient amount of data or training examples. One of the main issues of having small datasets is that of not having a precise prediction model, especially since the information gained in the learning process is fragile and unreliable. Therefore, the prediction uncertainty is largely affected by the lack of information obtained from the small dataset and leads to poor learning performances. This is due to the fact that small datasets are not able to provide enough information, and this leads to gaps between samples, this information gaps make the learning difficult to predict. [1]

Several approaches have been done previously, and have been successful, however the current approach will be based on the implementation of an autoencoder, since they have the advantage of using a specialized training process, and the capability of employing feature selection. [2] Autoencoders have the same inputs and outputs, therefore they use supervised learning which allows for the problem of small data to be tackled. The purpose of having an autoencoder which obtains feature selection from small datasets, it is expected to obtain a relatively solid training with both small training sizes and even when the testing sizes are not big enough. To prove the effectiveness of an autoencoder, the features obtained from the autoencoder will be used in a Neural Network and in an SVM classifier. This comparison will help evaluate the performance of using an autoencoder for feature selection to then use for training for a discriminative classifier.

II. BACKGROUND RESEARCH

Machine learning (ML) usually consists in the training of a single or multiple model to perform a specific task, this performance is then evaluated for the model to be fine-tuned until the performance no longer improves. [3] However, there are multiple approaches which complement ML and can increase efficiency and the overall execution depending on the desired task.

A. Multi-Task Learning

One of these approaches is Multi-Task Learning (MTL), which consist of utilizing the information obtained from the training signals of the related or auxiliary tasks and using this information to allow the model to make a better generalization of the original task. [3] This approach is actually fairly common, since every time more than one loss function is being optimized in the model, MTL is being executed. [3]

Multi-task learning is based on trying to exploit the results of certain features in the model by allowing them to be used in other auxiliary tasks. Firstly common features are found in the foremost layers of the network, while individual tasks are solved in later branches of the network. This can be done through what can be known as an encoder-decoder structure. [4] Auxiliary tasks are less important or might even be irrelevant to the overall main application of the model, yet despite being unrelated they are used to find a tougher and more robust representation of the input data which is then used for the main task and as a consequence improve the performance of the network. One of the main characteristics of these related tasks is that they should be easy enough to be learned and that they must require little to no effort to obtain the labels or annotations. The use of these tasks is to force the network to generalize a bigger amount of tasks and by having the auxiliary tasks the network restricts the parameter space during optimization and are therefore used as a regularization measure. [4]

In the encoder-decoder structure the auxiliary tasks explained previously, enact as a specialized decoder to the representation supplied by the encoder. The encoder favors the learning of the features in common, which are in turn exploited by the rest of the tasks, which enhance the performance of both the auxiliary tasks and the overall network. [4]

B. Deep Learning

Multi-task learning can also be used in deep learning methods, one of this methods is the Deep Neural Network, which consists of a neural network with a hierarchy of layers, which is used to extract representations form raw input data. Each hidden layer has an output that can be considered as a feature extraction, each output is used to construct advanced representation of the original data, and this is known as feature learning. Once a respectable representation of the features is obtained the classification can be done satisfactorily. [2] Deep Neural Networks are only one of the methods of deep learning, these approaches are important since according to [2] studies have shown, that for many applications, using a deep learning approach can outperform a standard machine learning method. One of the main issues of the deep learning approaches is overfitting, which will be discussed to a bigger extent later in the section.

C. Autoencoders

Another approach of Deep learning includes autoencoders, which play an important part in the extraction of the representation of input training patterns. [5] The representation that the autoencoder creates is an abstract representation which includes informative features to demonstrate a large set of data. [5] An autoencoder compresses and decompresses data, however they are data specific, which means that the data that they will be able to compress must be similar to the one they were trained on, and the decompressed data will always loose resolution, when compared to the original input data. [6] Autoencoders encompass the previously discussed terms, they follow a deep learning architecture and follow multi-task learning in the distance or loss function. They can also be used for classification and feature selection, especially after sparse regularization of the hidden outputs which allows for a high learning performance. [5]

D. Previous approaches

Recently several methods have bene proposed to deal with the problem of having a small data sets or a limited number of training examples. [2] One of the approaches is Deep Neural Networks, since they have a large number of parameters and therefore a great ability to classify complicated tasks, but it can lead to overfitting, especially when training samples are not enough. [2] One of the solutions for both overfitting and dealing with small data sets is data augmentation, which is basically to increase the size of the data set. The basis of data augmentation is to create new samples from the original samples by applying transformations and then using these new samples to enlarge the data set. [2] Some of the possible transformations include rotation, translation and flipping, to name a few, this allows the network to learn different variations of each of the features, which leads to a better rate of training and prediction effectivity. [2]

Another approach is the use of stacked autoencoders, this approach consists of a neural network built with autoencoders. This method has a specialized training process, it has the feature selection training and a fine-tuning section. Firstly, the feature selection training uses unsupervised learning to pre-train the layers of the network, the target output of this phase of each autoencoder is later used as an input. After the pre-training, the parameters are fine-tuned, this time using supervised learning, afterwards a classification layer is added, which uses the output of the last hidden layer as inputs. The network is then trained using backpropagation expecting to minimize the classification error. [2]

To complement the previous approaches, it must be mentioned the impact of semi-supervised learning, since it provides a framework to influence or manipulate unlabeled data. This means that unlike purely supervised learning, semi-supervised learning can improve their performance using both labelled and unlabeled examples. [7] According to [7] studies have proven that in certain cases, semi-supervised learning can almost match the performance of the pure supervised learning

approaches, even when labels have been discarded from the dataset.

III. Methodology

1. *Model created and proposed analysis*

The proposed project, as stated previously, will focus on the creation of a method that allows for the use of auxiliary tasks to regularize a network, this will be done through the implementation of an autoencoder for feature selection and a neural network for classification. The main premise of this approach is that there is important information contained in the features. The programming language in which the algorithm will be developed will be python, and it will make use of different libraries such as numpy, pandas, sckit-learn, seaborn and other libraries that will be used for the plotting of results and for the overall implementation of the algorithm. The main goal of this project is to utilize a small portion of each of the proposed data sets in order to verify that the features learned through the autoencoder can be used to train a standard discriminative neural network and progressively verify this by analyzing the whole data sets. Finally, the results will be compared to those obtained with a Support Vector Machines classifier.

The algorithm created was duplicated among three separate files, one for each of the datasets, since each dataset required a specific preprocessing approach. The models that the three files have in common is the autoencoder, the neural network and the SVM classifier. The model as created specifically for the autoencoder to realize a feature selection from the preprocessed datasets, which was then in turn used as an input for both of the classifiers. This was done to find two different approaches, while the original proposed method, specifies the use of an autoencoder with a neural network to solve the small dataset issue, the comparison will be done against an SVM classifier. This will be done by using an accuracy score obtained when comparing the predicted values of each classifier, against the actual values from the datasets, since the selected approach is supervised learning.

2. *Datasets used and preprocessing of each.*

Three different data sets were selected from the UC Irvine Machine Learning Repository, the data sets where selected from the main source to facilitate the acquiring of the data. The data sets that were selected are Breast Cancer Coimbra Data Set, Heart Disease Data Set and Autistic Spectrum Disorder Screening Data for Adolescent Data Set.

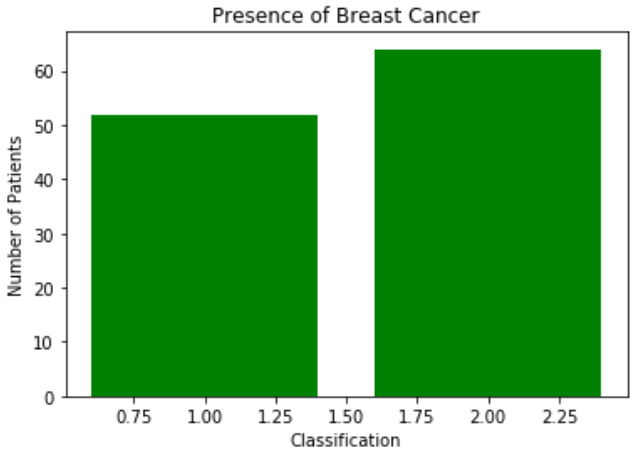

Fig. 1. Bar graph of the distribution of patients with breast cancer and the control group for the Breast Cancer Coimbra Data Set.

A. *Breast Cancer Coimbra Data Set*

The breast cancer Coimbra data sets is a clinical data which was obtained at the Faculty of Medicine of the University of Coimbra as well as University Hospital Centre of Coimbra, in the city of Coimbra, Portugal. The clinical research was observed in 64 patients with breast cancer, and a control group of 52 healthy persons. The data consist of 10 attributes or predictors, shown in table 1, all the attributes are quantitative; also a final attribute, a binary variable that indicates either the presence or absence of breast cancer. All the attributes are anthropometric and where all gathered through routine blood tests. The distribution of the data can be seen in figure 1.

The preprocessing of this dataset consisted in obtaining the type of object of each of the values in the columns. Then the data was visualized and inspected to see if the data contained missing values, when it did not, the data stayed with the dimension of 116 rows and 10 columns. The data was then split into “X” and “Y” components, to use for the autoencoder and the classifier.

B. *Heart Disease Data Set*

The heart disease data set is a clinical data which consists of data from 5 different sources: the Hungarian Institute of Cardiology, University Hospital Zurich, University Hospital Basel, V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. However the data selected for the development of this project will be focused specifically on the data from the V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. This database contains 76 attributes but only 14 of the 76 attributes are used for the diagnosis, therefore only these 14 attributes will be considered in the method, these can be seen in table 2.

TABLE I
QUANTITATIVE ATTRIBUTES:

Attribute	Quantity
Age	years
BMI	(kg/m2)
Glucose	(mg/dL)
Insulin	(µU/mL)
HOMA	
Leptin	(ng/mL)
Adiponectin	(µg/mL)
Resistin	(ng/mL)
MCP-1	(pg/dL)
Labels:	
1	Healthy controls
2	Patients

Attributes with corresponding units for each of features in the test for the breast cancer diagnosis on for the Breast Cancer Coimbra Data Set.

The classification of the results is based on whether the patient has a heart disease or not, if there is no presence of heart disease the value is 0, yet there are 4 different categories for the presence of heart disease. The cataloguing of the patients who do have heart diseases varies from 1 to 4, depending on the results of the heart disease. In figure 2 we can see the graphical representation of the dataset distribution among the 5 possible classifications depending on the presence of heart disease. These 5 possible classes will be the output used in the proposed method, the approach to dealing with a multi-class output will be explained with the preprocessing of the data.

The preprocessing of this dataset consisted in obtaining the type of object of each of the values in the columns. Then the data was visualized and inspected to see if the data contained missing values. The data contained 6 missing values, so data was processed to remove the rows with missing values, this left the dataset with the new dimension of 297 rows and 14 columns, compared to the original dimension of 303 rows and 14 columns. The data was then split into "X" and "Y" components, to use for the autoencoder and the classifier.

An important characteristic of this specific dataset was the classification classes, since this dataset was multi-class, instead of binary. In order to solve this the function "LabelBinarizer" was used. This creates a matrix in which the 5 classes are set to be filled with 0s and 1s, changing the output from a single column to 5 columns containing as a label the original value and having a 1, in the intersection of the row and column were their numerical value is located. Afterwards this new output was used for the classifier, since the output is now multi-class, the SVM classifier was not done in this specific dataset.

TABLE 2
QUANTITATIVE ATTRIBUTES:

Attribute	Quantity
Age	years
Sex	(1 = male; 0 = female)
Chest Pain Type	Value 1: typical angina Value 2: atypical angina Value 3: non-anginal pain Value 4: asymptomatic
Resting Blood Pressure (trestbps)	(mm Hg)
Cholesterol	(mg/dl)
Fasting Blood Sugar (fbs > 120mg/dl)	(1 = true; 0 = false)
Resting Electrocardiographic Results (restecg)	-- Value 0: normal -- Value 1: having ST-T wave abnormality -- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
Maximum Heart Rate Achieved (thalach)	
Exercise Induced Angina (exang)	(1 = yes; 0 = no)
Oldpeak	
Slope	-- Value 1: upsloping -- Value 2: flat -- Value 3: downsloping
Number Of Major Vessels Colored By Flourosopy (ca)	(0-3)
Thal	3 = normal; 6 = fixed defect; 7 = reversible defect
Diagnosis Of Heart Disease (num)	-- Value 0: < 50% diameter narrowing -- Value 1: > 50% diameter narrowing

14 of the 76 possible attributes for each of the features in the test for the heart disease diagnosis with their corresponding units for the Heart Disease Data Set.

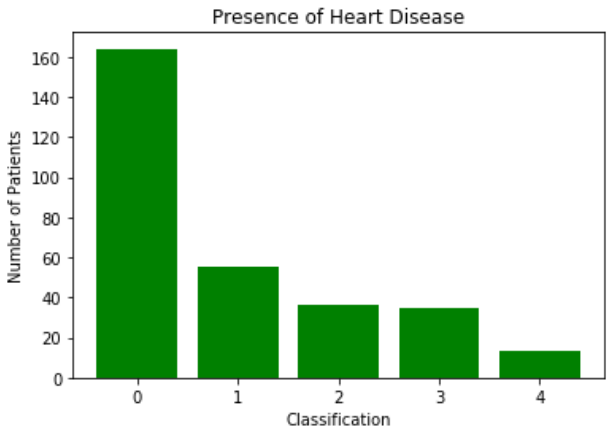

Fig. 2. Bar graph of the distribution of patients with heart for the Heart Disease Data Set.

C. Autistic Spectrum Disorder Screening Data for Adolescent Data Set

The autistic spectrum disorder screening data for adolescent data set is a clinical data set which was recorded in Auckland, New Zealand, they proposed a new data set which consists of 20 features that are used or determining influential autistic traits in adolescents, 10 of these features are behavioral, while the other ten are individual characteristics which have been proven to help in the detecting of autism, these attributes can be seen in table 3. These attributes where preprocessed, so they could be introduced into the model. The data sets consist of 104 instances and the final result consists of a predicted diagnosis of Autism Spectrum disorder or ASD, in figure 3 we can see the graphical representation of the dataset distribution among the results of the predicted diagnosis of the clinical condition.

The preprocessing of this dataset consisted in obtaining the type of object of each of the values in the columns. Then the data was visualized and inspected to see if the data contained missing values. Once the data was visualized the columns with qualitative attributes, meaning they were string objects not numerical, were highlighted and preprocessed. The ".map" function was used to change the string values to numerical values.

The data contained 12 missing values, so the data was processed to remove the rows with missing values, this left the dataset with the new dimension of 98 rows and 21 columns, compared to the original dimension of 104 rows and 21 columns. The data was then split into "X" and "Y" components, to use for the autoencoder and the classifier. However, when the string values were converted into numerical values the numerical value has a value which does not represent the data, so the "get_dummies" function was used to make sure the values in the multi-class columns did not affect the training and feature selection. It must also be mentioned that two columns were dropped due to their content being very similar to other columns, these were the columns of "Age Description" and "Country of Residence", however the columns "Age" and "Ethnicity" remained.

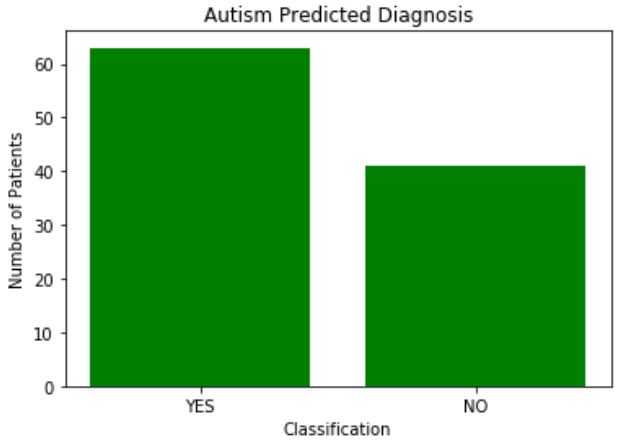

TABLE 3
QUANTITATIVE & QUALITATIVE ATTRIBUTES:

Attribute	Quantity
Age	years
Gender	(m = male; f = female)
Ethnicity	List of possible ethnicities (String)
Born with Jaundice	Boolean (yes or no)
Family Members with PDD	Boolean (yes or no)
Who is completing the test?	List of possible family relations (S
Country of Residence	List of possible ethnicities (String)
Used the Screening App Before	Boolean (yes or no)
Screening Score (Results)	Final numeric score obtained basec
Class/ASD predicted diagnosis	Boolean (yes or no)
Question 1 Answer	Binary (0,1)
Question 2 Answer	Binary (0,1)
Question 3 Answer	Binary (0,1)
Question 4 Answer	Binary (0,1)
Question 5 Answer	Binary (0,1)
Question 6 Answer	Binary (0,1)
Question 7 Answer	Binary (0,1)
Question 8 Answer	Binary (0,1)
Question 9 Answer	Binary (0,1)
Question 10 Answer	Binary (0,1)

20 possible attributes for each of the features in the test for the predicted diagnosis of ASD with their corresponding units for the Autistic Spectrum Disorder Screening Data for Adolescent Data Set

Fig. 3. Bar graph of the distribution of the predicted diagnosis of patients with ASD for the Autistic Spectrum Disorder Screening Data for Adolescent Data Set.

IV. EXPERIMENTS

Once the proposed model was created, the approached followed was the creation of an autoencoder followed by a neural network. The objective of the autoencoder was to learn feature selection from the input data, features which in turn were fed to the neural network as inputs, this neural network worked as a discriminative classifier.

The main goal of this project was to analyze the potential of using feature extraction in small datasets, and the experiments focused on the impact the amount of training data had on the accuracy score of the classifiers. The experimentation followed the increase of the amount of data used form training in the autoencoder, since the autoencoder did not make use of the "Y" component of the data, this was used purely for the classification algorithms.

Several experiments were made for each of the datasets, the approach followed for the experiments was the following:
The encoder sizes were varied depending on the datasets, this was to see the impact of the feature selection on the classification.

The number of neurons in the neural network used were also varied, however to maintain certain homogeneity among the experiments within the different datasets, the neurons used were between 28 and 30. Also the number of hidden layers was established as 2, this allowed for better results in the first drafts of the experiments and was ultimately established as a result. The number of neurons was also established through trial and error, based on the documentation of the Scikit-learn description of the neural network function.

Where the experiment focused was the amount of training data used. The first iteration used 30% of the whole dataset for training, followed by 50%, then 75%, then 90%, and finally 98%.

While it might seem strange to use almost all of the data for training, since it can easily lead to overfitting, as it did on certain points, the idea behind was to see the importance the training had on the feature selection, and how small testing datasets could also be an area of opportunity for the autoencoders.

Breast Cancer Coimbra Data Set						
Neural Network						
Encoder Size	**Neurons**	**30% training**	**50% training**	**75% training**	**90% training**	**98% training**
4	28	50%	46.55%	41.38%	41.67%	50%
	30	50%	53.45%	58.62%	58.33%	53.45%
5	28	50%	53.45%	58.62%	58.33%	50%
	30	48.78%	46.55%	41.38%	41.67%	50%
SVM Classifier						
Encoder Size	**Kernel**	**30% training**	**50% training**	**75% training**	**90% training**	**98% training**
4	Linear	54.29%	63.79%	44.83%	41.67%	66.67%
5		60%	55.17%	75.86%	33.33%	33.33%

Fig. 4. Table showing the results for the Breast Cancer Coimbra Dataset of both classifiers and for each of the experiments.

In figure 4, the results for the experiments done with the Breast Cancer Coimbra Dataset can be seen. As stated previously the experiment followed the used of two classifiers, an SVM and a Neural Network, the parameters modified were the encoder size and the number of neurons. In the table we can see that the best overall results were obtained with both 75% and 90%, in the 98% we can see that the accuracy percentage decrease, this could be due to overfitting, and because the amount of testing data was very small.

It is important to point out that the project was focusing on small datasets and in having a small training data, it is expected for the accuracy to increase as the amount of training data increases, however the accuracy score for both classifiers in the 30% is over 50%, which is adequate for the amount of training data provided. This score is due to the use of the autoencoder. It is also important to mention that the best results were obtained with the SVM classifier, at 75% training data, which is a standard amount for ML and other data analytics.

Heart Disease Data Set						
Neural Network						
Encoder Size	**Neurons**	**30% training**	**50% training**	**75% training**	**90% training**	**98% training**
6	28	52.40%	49.66%	52%	53.33%	66.67%
	30	52.00%	50.00%	52%	53.00%	66.67%
7	28	52.40%	49.66%	52.33%	53.33%	66.00%
	30	3%	3%	4%	4%	4%

Fig. 5. Table showing the results for the Heart Disease dataset of both classifiers and for each of the experiments.

In figure 5, the results for the experiments done with the Heart Disease Dataset can be seen. This dataset was analyzed differently than the two other datasets, unlike the other experiments, this dataset was only analyzed using the Neural Network, because of the multi-class classification output. The parameters modified where the encoder size and the number of neurons. In the table we can see that the best overall results was obtained at 98%, which is not very reliable, since a small amount of testing data can lead to a false high accuracy.

When compared to the previous dataset, the Breast Cancer dataset, the Heart Disease dataset had more parameters, this means that the feature selection contained more attributes. It is important to remark that while the feature selection had more attributes, the accuracy score was lower overall in the heart disease dataset. Actually, the fact that the accuracy is close to 50% in most of the experiments can lead to the conclusion that the autoencoder proposed, doesn't work that well with multi-class labels, when compared to the 75% accuracy obtained in the previous dataset, which is binary.

Autistic Spectrum Disorder Screening Data for Adolescent Data Set						
Neural Network						
Encoder Size	**Neurons**	**30% training**	**50% training**	**75% training**	**90% training**	**98% training**
25	28	66.67%	77.55%	80%	90%	97%
	30	63.77%	63.27%	72%	80%	98%
30	28	63.77%	63.27%	72%	80%	98%
	30	63.77%	63.27%	72%	80%	98%
SVM Classifier						
Encoder Size	**Kernel**	**30% training**	**50% training**	**75% training**	**90% training**	**98% training**
25	Linear	94.20%	89.80%	97%	80%	98%
30		97.10%	91.84%	96%	96%	97%

Fig. 6. Table showing the results for the Autistic Spectrum Disorder Screening Data for Adolescent Data Set of both classifiers and for each of the experiments.

In figure 6 the results of the experiments done with the Autistic Spectrum Disorder Screening Data for Adolescent Data Set can be seen. In this experiment both the Neural Network and the SVM classifier where used. The parameters that were updated were the encoder size from 25 to 30 and the number of neurons. It can be seen in the table that the best overall results are focused on the 75% and 90% of training data, ignoring as in the first dataset the results of 98%, since this could be due to a false high

accuracy. In this specific dataset we can see how the accuracy score of using 30% for training data is higher that the overall accuracy score of the Heart Disease dataset.

V. Discussion

In the previous section the results of the experiments done for each dataset can be seen. While the results are briefly discussed, in this section certain key aspects will be presented. The most important inference that can be derived from the results is the impact of the number of attributes that a dataset contains. This can be reflected in the results tables, figures 4 – 6, in the encoder size. This impact goes beyond the simple variation done for the experiments but based on them it can be inferred that a larger number of attributes will lead to a better feature selection created by the autoencoder. This can be seen clearly in fig 6, where there is an almost constant growth between data intervals, since the accuracy increases as each iteration of training data increases. The dataset of Autistic Spectrum Disorder Screening Data for Adolescent is the one with the highest performance out of the three datasets, this can be due to the autoencoder creating a more reliable feature selection, since it is based on a broader quantity of attributes.

Another important aspect is the outperformance of the SVM classifier over the Neural Network. One of the possible reasons is that the data simply has a best fit for the SVM, another possible reason is that the Neural Network was a shallow-Neural Network and that could lead to the learning of the model not being as robust.

Nevertheless, obtaining an 80% accuracy with a 75% training data is a good accuracy score for the Autistic Spectrum Disorder Screening Data for Adolescent dataset. The other datasets contain less attributes, which as stated previously could be one of the reasons that the accuracy score increase in the Autism dataset.

In the experiments done with the Heart Disease dataset, the accuracy scores were fairly consistent throughout all the different variations, this can also be observed in the experiments done with the Breast Cancer dataset, this more stable learning rate, leading to a more stable increase in accuracy could be due to the small number of features learned, and the small encoder size.

VI. Conclusion

It can be concluded that the proposed project of using an autoencoder for feature selection and then using the learned features as inputs for a Neural Network will work suitably. However, the datasets that are being used must be taken into consideration carefully, because even if the proposed model works on small datasets, the bigger the training data, the more accurate learning. Also, as discussed previously, the number of features also play a big part on the autoencoder phase, and therefore they will also affect the classifier. One of the main issues with the model is that it was outperformed in some cases by the SVM classifier, for further work, this issue could be explored profoundly. Also, further work could focus on using more multi-class data, in order to be able to compare the behavior of different classifiers and with different data classification. Similarly, once more datasets are selected which are multi-class, a second classifier would have to be researched for comparing the results from the proposed model to another classifier that can be used for multi-classes.

VII. References

[1] M. A. Lateh, A. K. Muda, Z. Izzah Mohd Yussof, N. A. Muda and M. S. Azmi, "Handling a Small Dataset Problem in Prediction Model by employ Artificial Data Generation Approach: A Review," *Journal of Physics: Conference Series,*, vol. 892, no. 1, 2017.

[2] J. Xue, P. P. Chan and X. Hu, "EXPERIMENTAL STUDY ON STACKED AUTOENCODER ON INSUFFICIENT," in *Proceedings of the 2017 International Conference on Wavelet Analysis and Pattern Recognition*, Ningbo, China, 2017.

[3] S. Ruder, "An Overview of Multi-Task Learning," *Insight Centre for Data Analytics*, 2017.

[4] L. Liebel and M. Körner, "Auxiliary Tasks in Multi-task Learning," *Computer Vision Research Group, Chair of Remote Sensing Technology*, 2018.

[5] J. Liu, C. Li and W. Yang, "Supervised Learning via Unsupervised Sparse Autoencoder," *IEEE Access*, vol. 6, pp. 73802 - 73814, 2018.

[6] F. Chollet, "The Keras Blog," Pelican, 14 May 2016. [Online]. Available: https://blog.keras.io/building-autoencoders-in-keras.html. [Accessed 19 February 2018].

[7] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk and I. J. Goodfellow, "Realistic Evaluation of Deep Semi-Supervised," *Google Brain*, 2018.