VISION-BASED ESTIMATION OF VOLUME STATUS IN ULTRASOUND

By

© Jordan Smith

A Thesis submitted to the

School of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Engineering

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

May 2020

St. John's Newfoundland and Labrador

Abstract

This thesis provides a proof-of-concept approach to the analysis of ultrasound imagery using machine learning and computer vision for the purposes of tracking relative changes in apparent circulating blood volume.

Data for the models was collected from a simulation which involved having healthy subjects recline at angles between 0 and 90 degrees to induce changes in the size of the internal jugular vein (IJV) resulting from gravity. Ultrasound video clips were then captured of the IJV. The clips were segmented, followed by feature generation, feature selection and training of predictive models to determine the angle of inclination. This research provides insight into the feasibility of using automated analysis techniques to enhance portable ultrasound as a monitoring tool.

In a dataset of 34 subjects the angle was predicted within 11 degrees. An accuracy of 89% was achieved for high/low classification.

Acknowledgements

I would like to thank my supervisors Dr. Andrew Smith, Dr. Peter McGuire and Dr. Mohamed Shehata. Without their continued support I could not have perservered to complete this.

The work presented in this thesis was supported in part through funding from Memorial University of Newfoundland, MITACS, ignite and in-kind support from C-Core.

Thanks also to Spencer Bellows, Ramsey Powell, Des Whelan, Javad Anaraki, Ebrahim Karami and the others who have helped painstakingly process this dataset.

Finally, I’m eternally grateful to Melissa Mackey and my parents for their patience, encouragement and care throughout the process.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Overview

Point of Care Ultrasound (POCUS) is being used in many different disciplines of medicine due to its availability, low cost and lack of ionizing radiation. It can diagnose abdominal bleeding in trauma, rule out an ectoptic pregnancy, assess for gallstones or be used for needle guidance during invasive procedures [1]–[3]. Ultrasound also enhances diagnostic ability if a patient has an unclear history. Abnormal intravascular volume status (the amount of fluid in a subject's circulatory system) may also indicate sepsis, trauma or congestive heart failure [4]–[8]. As morbidity and mortality increases if too much or too little fluid is given, volume status monitoring is critical in guiding fluid resuscitation of critically ill patients [9].

The usefulness of ultrasound is highly dependent on the operator's skill in image generation. Algorithmic processing can be used to augment the normal utility of the device and allow more effective treatment. Currently a clinician may visually estimate the jugular venous pressure (JVP) of a subject to assess volume status or estimate the anteroposterior (AP) diameter of the internal jugular vein (IJV) in ultrasound [10], [11]. However, modern image processing techniques allow for the extraction of a great deal of information from images and videos. By applying image processing techniques to portable ultrasound it is possible to obtain a more consistent or reliable estimation [12]–[16].

1.2 Objectives and scope

The objective of this research is to develop an algorithm to detect relative changes in blood volume using healthy subjects positioned at different angles of inclination. The angles of inclination change the size of the internal jugular vein (IJV) as a result of gravity. The size of the IJV has been shown to correlate with cardiac preload[17], [18].

Videos can be preprocessed with a variety of methods including watershed [19], [20], active contours [13], [21] or convolutional neural networks [22], [23]. The result of preprocessing is either a contour drawn around the internal jugular vein (IJV) from which several measurements can be made, or a direct embedding of IJV characteristics from the frame into a lower dimensional tensor. These representations can be used to train models which predict the angle of inclination. To test the model a new unseen video clip can be processed and the model should predict the correct metadata for the clip (such as bed angle for a simulation or estimates of a subject's volume status)

1.3 Thesis structure

Chapter 2 presents background and relevant literature to describe the application and establish context. Next, Chapter 3 describes collection and treatment of data. This is followed by a discussion of features which were created from the dataset in Chapter 4, feature selection methods used to pare down these feature are described in Chapter 5, Chapter 6 compares several models which were trained from these feature subsets, Chapters 7 and 8 summarize and discuss the results and a reflection inviting further research on the topic is given in Chapter 9.

1.4 Contributions

As a result of this research we have successfully i) augmented point of care ultrasound [24] ii) segmented the vessels of interest [19], [20] iii) found a better understanding of what characteristics of ultrasound are useful in predicting volume status [25] iv) extracted physiological information from ultrasound clips [26] and v) made reliable predictions using ultrasound recordings.

1.5 Publications

The following papers were published during this research:

J. P. Smith, M. Shehata, R. G. Powell, P. F. McGuire, and A. J. Smith, "Physiological Features of the Internal Jugular Vein from B-Mode Ultrasound Imagery," in *International Symposium on Visual Computing (ISVC)*, 2016, pp. 312–319.

J. P. Smith, M. Shehata, P. F. McGuire, and A. J. Smith, "Features of Internal Jugular Vein Contours for Classification," in *International Symposium on Visual Computing (ISVC)*, 2016, vol. 10073 LNCS, pp. 421–430.

J. Smith, M. Shehata, P. McGuire, and A. Smith, "Texture Features for Classification of Vascular Ultrasound," *International Association for Pattern Recognition (IAPR) 3rd International Workshop on Pattern Recognition for Healthcare Analytics*. 2016.

(Best poster award) J. Smith, M. Shehata, P. McGuire, and A. Smith,

“Augmentation of Point of Care Ultrasound,” in *IEEE Newfoundland Electrical and Computer Engineering Conference (NECEC)*, 2016.

S. Bellows, M. Shehata, J. Smith, P. Mcguire, and A. Smith, “Validation of a Computerized Technique for Automatically Tracking and Measuring the Inferior Vena Cava in Ultrasound Imagery,” *Biosyst. Eng.*, vol. 138, pp. 4–10, Oct. 2015.

S. Bellows, J. Smith, P. Mcguire, and A. Smith, “Validation of a computerized technique for automatically tracking and measuring the inferior vena cava in ultrasound imagery.,” *Stud. Health Technol. Inform.*, vol. 207, pp. 183–92, 2014.

In addition to the publications above the dataset we collected was involved in several other publications and the research further extended, providing additional support for the utility of the work. Additional support for the predictive capability of IJV area can be found in Javad R. Anaraki’s work and the initial segmentation and tracking was improved by Ebrahim Karami [12]–[16], [27], [28]

Chapter 2 Background and literature review

2.1 Anatomy

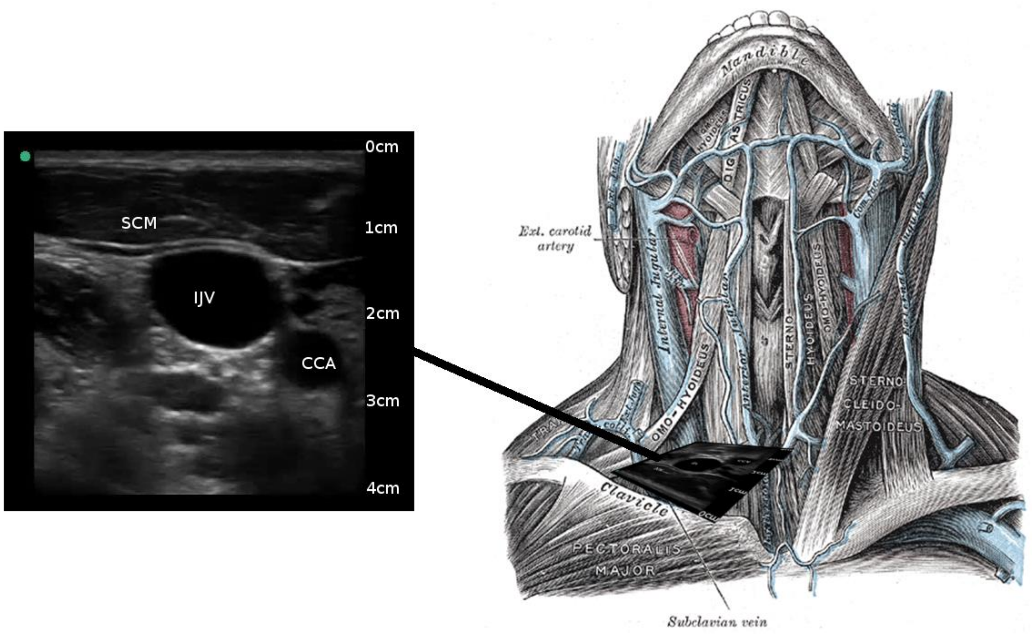


Figure 1 : Context for ultrasound imagery, plate from Gray's Anatomy at right [29]

The internal jugular veins (IJV) are a pair of thin walled vessels that run the length of the neck that return deoxygenated blood from the brain to the heart. They are beneath several muscles including the sternocleidomastoid (SCM). Each runs alongside a high pressure, thick walled artery called the common carotid (CCA). The jugular veins drain the head and neck, getting larger at the bottom before merging into the superior vena cava and entering the heart. [29]–[31]

At left of Figure 1, ultrasound of the internal jugular vein. At right, diagram of the neck showing where the ultrasound intersects the jugular (image becomes flipped).

2.2 Motion of the vessels of the neck

When a subject is standing upright the jugular forms a vertical column of blood. Blood pools at the base due to gravity and pulls the top closed behind it. This is because the carotid sheath is filled with a material that is less dense than blood and the IJV is very flexible. As blood enters the heart, the column drops and as the top part of the heart pumps, blood is forced back up the column. The top of the column of fluid bounces with respect to Jugular Venous Pressure (JVP) and has an interesting two part cycle (compared to blood exiting the heart, which moves the carotid with only one pulse per cycle). With sufficient training a bright indirect light source allows both pulses to be seen on the surface of the neck. [32]–[34]

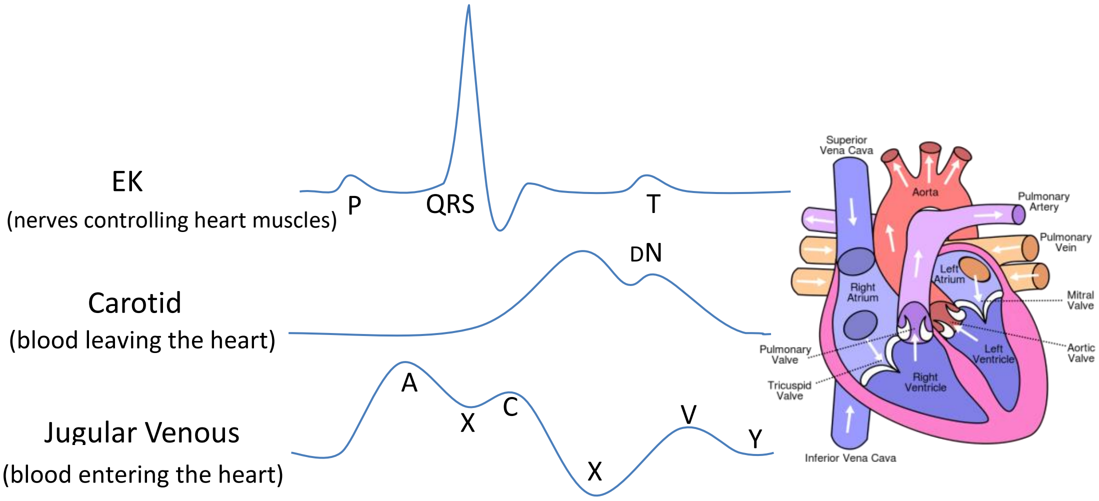


Figure 2 : Cardiovascular waveforms. JVP waveform derived from[35]**, heart diagram from** [36]

The portion of the signal labelled QRS triggers a big pump and the carotid receives blood - a little notch (DN) is seen as the aortic valve closes. After a delay the blood returning down the jugular collides with the closing atrium of the heart, which pushes blood back upward (A). Pressure then drops (X) as the atrium of the heart relaxes and pulls new blood in. While this is happening the ventricles of the heart are contracting, buldging the tricuspid (C). Once the atrium is full incoming blood will start to back up (V) as blood continues to enter while the valve is closed – it drains into the right ventricle once the tricuspid opens (Y).

The peaks of the JVP are out of phase with those of the carotid and they also vary with breathing, as seen in Figure 3.

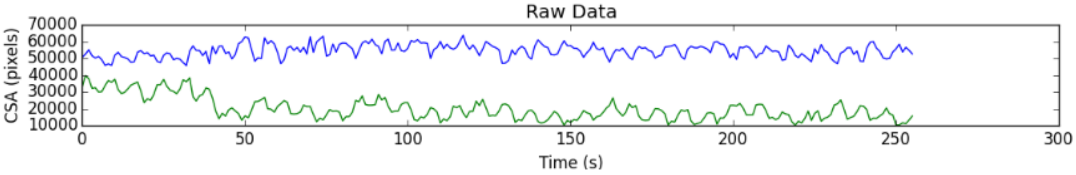


Figure 3 : Cross sectional area of each vessel from a transverse ultrasound of the neck (data collected during this research)

At top of Figure 3, the cross-sectional area of the carotid vs time. The lower plot of the jugular shows dips where the subject inhaling has pulled blood from the jugular. Note that the jugular is out of phase with the carotid and while noisy, the landmarks of the previous plot are visible in some sections. JVP waveforms are generally difficult to obtain noninvasively but are easy to obtain using ultrasound.

2.3 Jugular venous pressure (JVP)

The height of the column of fluid in the jugular is a clinical measure of ‘volume status’, the amount of fluid in the body[37]. If the pulsing top of the column is found too high up the neck the subject’s JVP is considered elevated. Elevated JVP is a sign that the subject is volume overloaded. In cases of congestive heart failure, the heart is failing as a pump and fluid is backing up before reaching the heart resulting in an elevated JVP.

Modelling the internal jugular vein as an idealized column of fluid helps to explain how the change in a subject’s position affects the appearance of the jugular.

The following hydrostatics equation models a column of fluid [38]:

$$P = \rho g h \tag{1}$$

Where:

- P is fluid pressure (in pascals)
- ρ is density (1.0 gram per cm^3 for water, 1.010-1.060 g/cm^3 for blood)
- g is gravity (980.7 cm/s^2 on earth)
- h is height (in centimeters)

It is likely safe to assume gravity and density are constant. By measuring height one can infer pressure; it is important to note that the height is determined entirely by pressure, not the shape of the container. Higher columns of fluid have a higher pressure at the base. Likewise, bursts of pressure at the base will raise a column.

If the container is angled, fluid will move farther along the length of the tube (L) until it has traveled the required height (h) according to:

$$P = \rho g sin(\theta) \cdot L \quad (2)$$

$$L = \frac{1}{\rho g \sin(\theta)} \cdot P \quad (3)$$

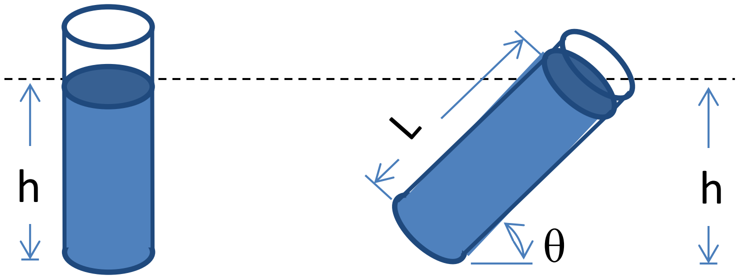


Figure 4 : Fluid model of JVP

The way that jugular venous pressure is measured reflects this. In most healthy subjects the height of the column of blood is only a few centimeters above the heart, placing it behind the collar bone. However, if the subject is gradually reclined the JVP appears in a visible location along the lower part of the neck [17], [39], [40].

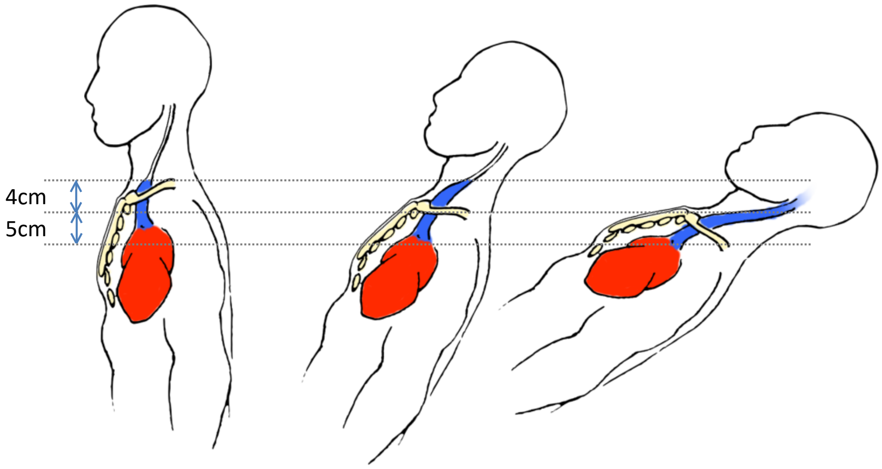


Figure 5 : Effect of pose on JVP

The top of the sternum in the center of the chest (the sternal angle) lies just at the top of the heart (the bottom of the column of fluid). JVP is measured in 'centimeters above sternal angle' because it is a direct representation of the h in equation (1) above, proportional to pressure. After 5cm is added to account for depth to right atrium, the measurement estimates CVP [34]. Volume status is inferred as follows [41]:

Table 1 Effect of Volume on JVP

Volume deplete (Hypovolemic)	Normal (Normovolemic)	Volume overload (Hypervolemic)
<5cm	6-10cm	>10cm

It is measured with a ruler and straight edge as shown in Figure 6 [37]

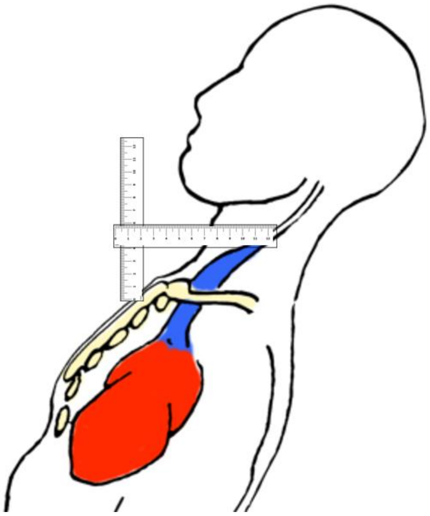


Figure 6 : Manual measurement of JVP with rulers

If an ultrasound machine is used to image the jugular vein so that its whole length is visible, the point of taper can be found with more precision than when observing skin for a pulse. The internal jugular vein (IJV) is shown in cross section in Figure 7. The size and shape of the IJV cross section at a given position is proportional to the amount of fluid in the body (volume status). As a patient's volume status changes the shape of the vessel will change. Elevated volume status related to congestive heart failure might register as a change in this vessel. [18], [42]–[44].

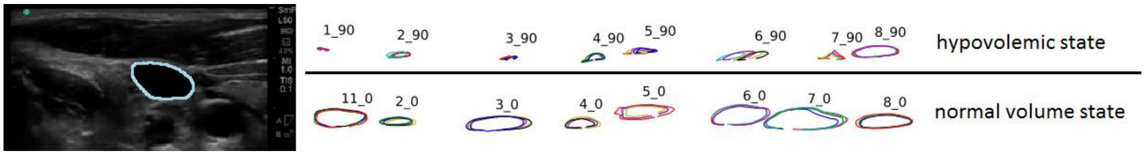


Figure 7 : (left) Ultrasound of the neck showing the IJV (right) A sample of IJV contours for different volume statuses.

2.4 Simulation

If an ultrasound is placed at a fixed point along the neck, the apparent diameter of the IJV should change with subject angle. As angles approach vertical, the IJV should appear collapsed as the vessel’s taper passes below the point of ultrasound. Also as pressure increases, the elastic IJV stretches. Pressure is proportional to the amount of fluid above the measurement point. Reclining the subject will increase pressure and the cross sectional area of the vessel will increase as seen in Figure 7.

The simulation used in this research involved changing a subject's position from flat on their back to an upright sitting position using a regular folding stretcher. As shown in Figure 8, the angle of recline starts at sitting (90°) and approaches horizontal (0°). The maximum increase in IJV cross sectional area is seen at the 0° position.

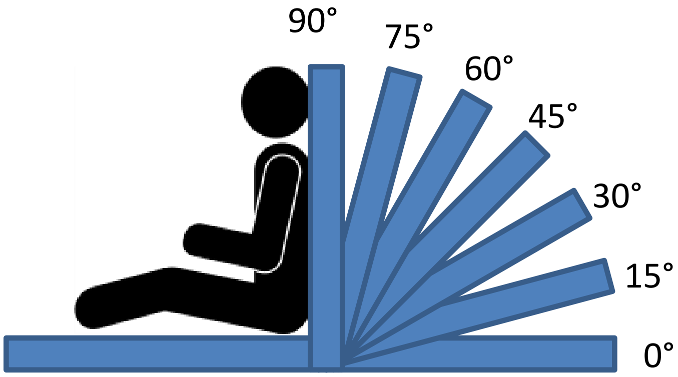


Figure 8 : Different bed angles at which subject was reclined.

2.5 Characteristics of ultrasound

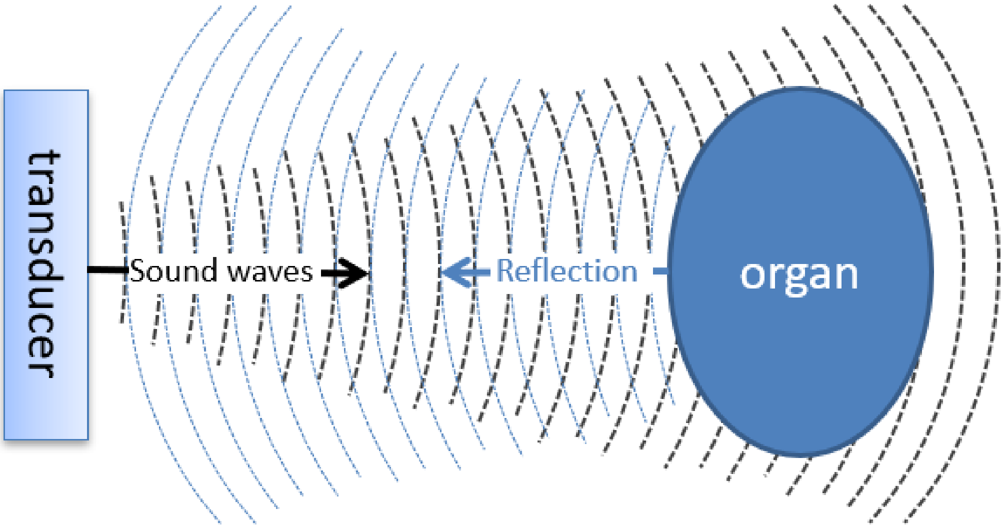


Figure 9 : Ultrasound image production

As illustrated in Figure 9, ultrasound imagery is produced by transmitting high frequency sound waves through a material. As the waves reflect, sound which bounces off object that are farther away will take longer to return. The distance to objects can be measured by timing the reflections. By directing the sound to different locations a sort of map is produced showing the shape of reflective things.

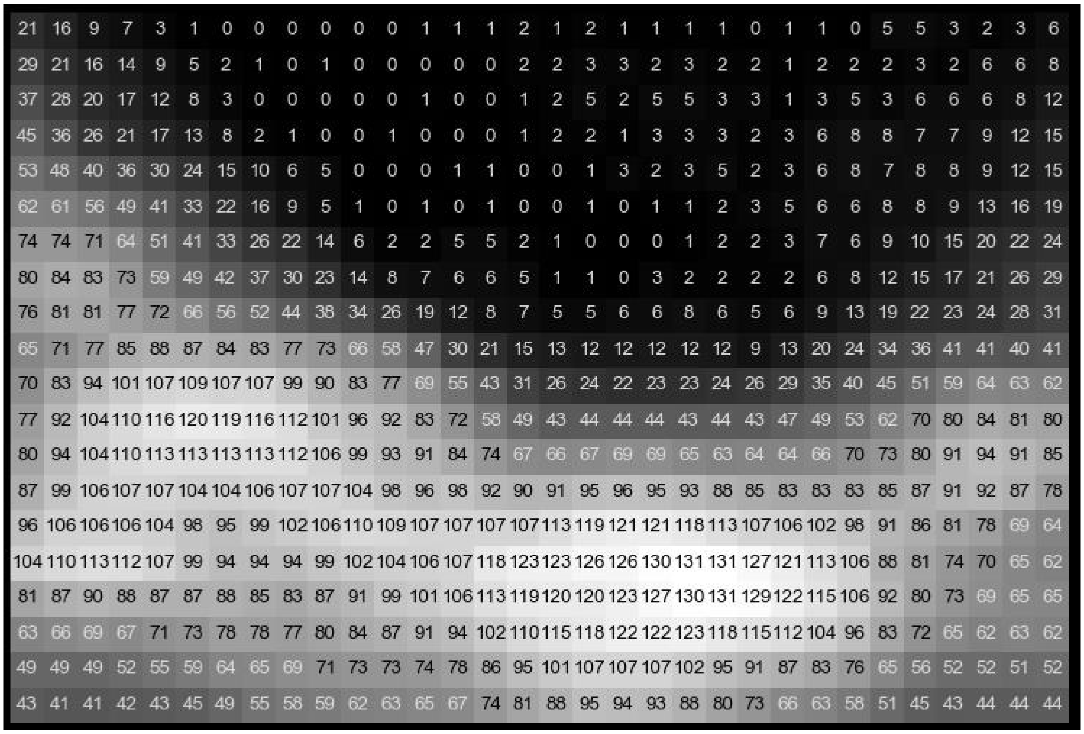
Figure 10 : Textures seen in ultrasound

However, it’s not just hard white edges that can be found. The sound will continually reflect back as it passes through some materials. Things which are better at absorbing sound (like blood) will reflect back less sound. Regions of higher densities and viscosities (like those found in different organs) can appear as blobs of different shades of grey. In addition, certain materials will have interesting density patterns which may scatter sound waves in unique ways giving the grey shapes their own texture.

There are many regions which still appear indistinguishable from each other in location, brightness and shape – for example many blood vessels are just round black shapes that vary in position between patients. If the scan repeatedly collects images to create a video, the motion allows better classification of anatomy - vessels such as veins may move differently than arteries. The motion itself is useful for diagnosis as illness may change the behavior of organs (as seen in the collapsibility index used for blood volume assessment).[45], [46]

2.6 Image and video processing

A grayscale picture can be represented as a grid of numbers, where a dark point in the image is a low number and a bright point in the image is a higher number (see Figure 11). An image will usually have hundreds of thousands of these picture elements (or pixels). Videos are just a sequence of images displayed at a rate of around 20 to 60 per second. By preprocessing a video, the amount of information that needs to be analyzed by a classifier can be significantly reduced.



21	16	9	7	3	1	0	0	0	0	0	0	1	1	1	2	1	2	1	1	1	1	0	1	1	0	5	5	3	2	3	6
29	21	16	14	9	5	2	1	0	1	0	0	0	0	0	2	2	3	3	2	3	2	2	1	2	2	2	3	2	6	6	8
37	28	20	17	12	8	3	0	0	0	0	0	1	0	0	1	2	5	2	5	5	3	3	1	3	5	3	6	6	6	8	12
45	36	26	21	17	13	8	2	1	0	0	1	0	0	0	1	2	2	1	3	3	3	2	3	6	8	8	7	7	9	12	15
53	48	40	36	30	24	15	10	6	5	0	0	0	1	1	0	0	1	3	2	3	5	2	3	6	8	7	8	8	9	12	15
62	61	56	49	41	33	22	16	9	5	1	0	1	0	1	0	0	1	0	1	1	2	3	5	6	6	8	8	9	13	16	19
74	74	71	64	51	41	33	26	22	14	6	2	2	5	5	2	1	0	0	0	1	2	2	3	7	6	9	10	15	20	22	24
80	84	83	73	59	49	42	37	30	23	14	8	7	6	6	5	1	1	0	3	2	2	2	2	6	8	12	15	17	21	26	29
76	81	81	77	72	66	56	52	44	38	34	26	19	12	8	7	5	5	6	6	8	6	5	6	9	13	19	22	23	24	28	31
65	71	77	85	88	87	84	83	77	73	66	58	47	30	21	15	13	12	12	12	12	12	9	13	20	24	34	36	41	41	40	41
70	83	94	101	107	109	107	107	99	90	83	77	69	55	43	31	26	24	22	23	23	24	26	29	35	40	45	51	59	64	63	62
77	92	104	110	116	120	119	116	112	101	96	92	83	72	58	49	43	44	44	44	43	44	43	47	49	53	62	70	80	84	81	80
80	94	104	110	113	113	113	113	112	106	99	93	91	84	74	67	66	67	69	69	65	63	64	64	66	70	73	80	91	94	91	85
87	99	106	107	107	104	104	106	107	107	104	98	96	98	92	90	91	95	96	95	93	88	85	83	83	83	85	87	91	92	87	78
96	106	106	106	104	98	95	99	102	106	110	109	107	107	107	107	113	119	121	121	118	113	107	106	102	98	91	86	81	78	69	64
104	110	113	112	107	99	94	94	94	99	102	104	106	107	118	123	123	126	126	130	131	131	127	121	113	106	88	81	74	70	65	62
81	87	90	88	87	87	88	85	83	87	91	99	101	106	113	119	120	120	123	127	130	131	129	122	115	106	92	80	73	69	65	65
63	66	69	67	71	73	78	78	77	80	84	87	91	94	102	110	115	118	122	122	123	118	115	112	104	96	83	72	65	62	63	62
49	49	49	52	55	59	64	65	69	71	73	73	74	78	86	95	101	107	107	107	102	95	91	87	83	76	65	56	52	52	51	52
43	41	41	42	43	45	49	55	58	59	62	63	65	67	74	81	88	95	94	93	88	80	73	66	63	58	51	45	43	44	44	44

Figure 11 : Pixel values at edge of vessel in ultrasound image taken from videos clip collected during research.

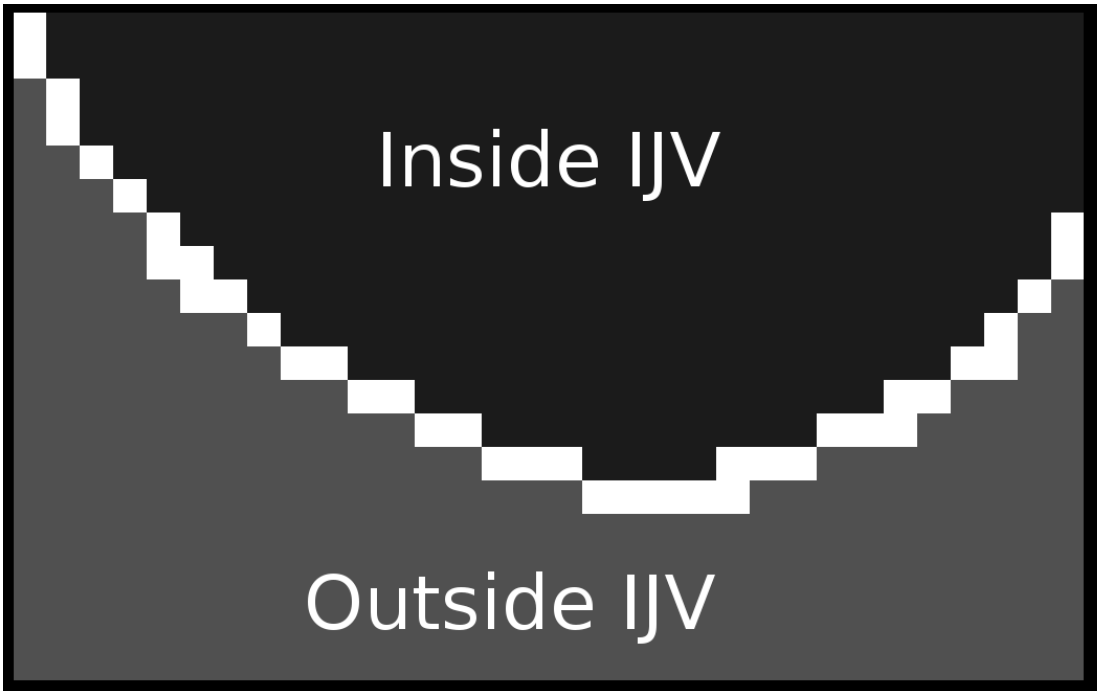


Figure 12 : Simplification of Figure 11 showing boundary of vessel seen in ultrasound image.

In processing an ultrasound video clip, one might extract the contours that define relevant anatomy, labeling what parts of the image are inside or outside of a vessel. A sample of such a labelled contour is shown in Figure 12. In a clip showing vessels of the neck, the internal jugular vein is usually a region of dark pixels surrounded by bright pixels. This the effect of hypo-echoic material compared to hyper-echoic material. Sound passes well through fluid with little to no reflection, but highly reflective materials like bone or dense tissue reflect back a lot of sound with little sound transmitting deeper into the body. The boundary between the low numbers of the inside of the vessel and the higher numbers outside can be found using a variety of techniques.

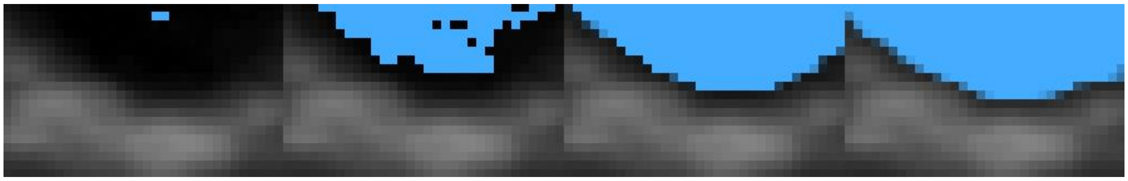


Figure 13 : Watershed segmentation of vessel as used in [20]

One technique involves creating a chain of adjacent pixel locations inside the vessel that is gradually expanded to include all neighbors of a similar grey level. This is called the 'watershed' algorithm and was successfully tested on a vein deep inside the body called the inferior vena cava and also the internal jugular vein in the neck[19], [20]. An illustration of the watershed process is shown in Figure 13. A drawback of the watershed technique is that dense materials create shadows in ultrasound. These artifact regions are completely black and may be confused with a vessel, leading to exaggeration of the area measurements created by the segmentation.

A more complex algorithm called ‘active contours’ or ‘snakes’ involves smoothing and refitting the chain[47]. The equations model it as an elastic band with specific tension and curvature requirements that keep it looking like the vessels it is outlining. This approach was first developed by Kass, Witkin and Terzopoulos 1988 [47] and applied to the jugular by Qian et al [21] and also Karami et al [15]. As shown in Figure 14, active contours seek to minimize an 'energy' equation balancing the internal forces of tension and curvature versus the change in brightness in the image (the image gradient).

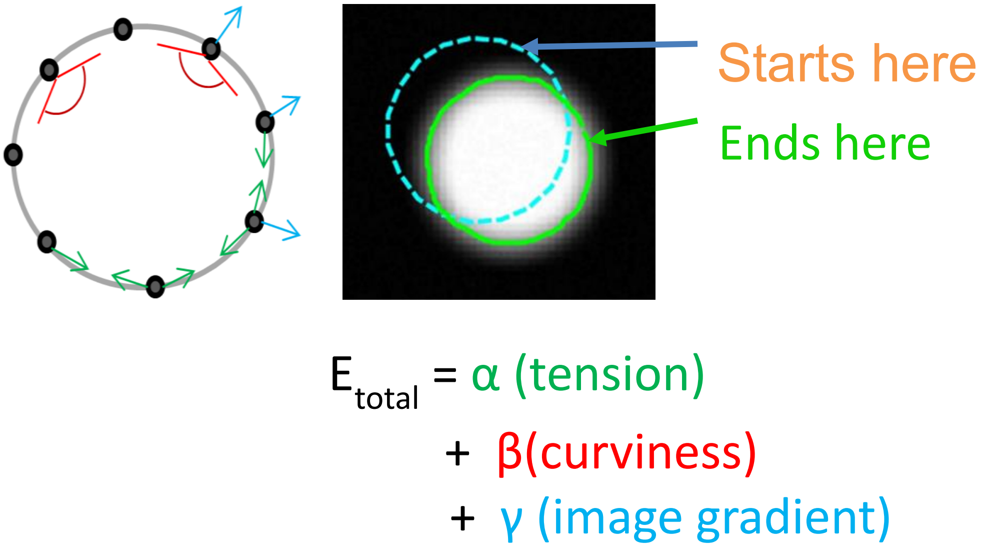


Figure 14 : Active contour description as given in [47]**.**

While active contours are bounded and a better model of the target vessel (the veins themselves are thin elastic walls), they can be quite slow to compute due to the iterative nature of the contour fitting and the complexity of the actual energy calculation.

Starting in the mid 2010’s, fully neural network based segmentation became possible [48]. These approaches allow processing images on specialized hardware and can be quite fast to compute. Architectures such as UNet [49] are particularly appropriate for biomedical image segmentation where images tend to be grayscale and of particular resolutions or content. These have successfully been applied to medical ultrasound of vessels and a sample of UNet segmentation applied to our dataset is shown in Figure 16 [22], [23], [50], [51].

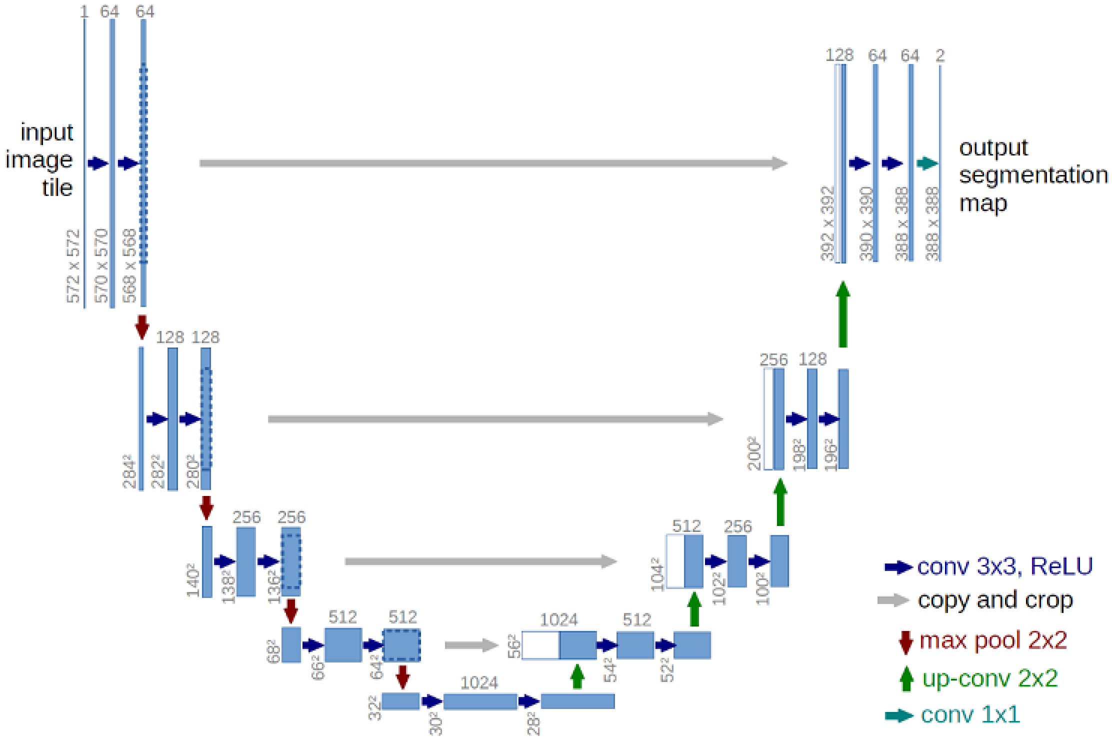


Figure 15 : UNet architecture. Figure from [49]**.**

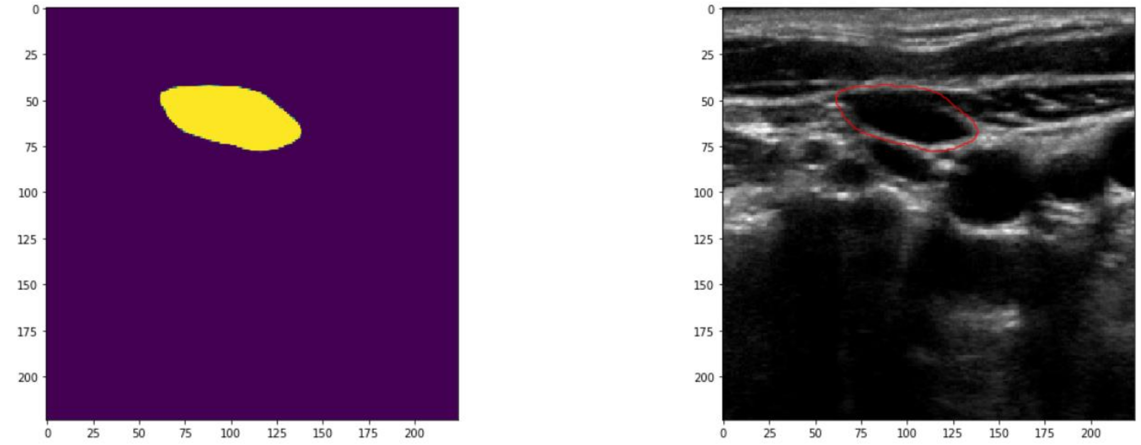


Figure 16 : UNet segmentation of our ultrasound dataset

They work by progressively pooling the content of a frame to a simpler tensor representation through a series of residual neural network (ResNet) [52] type convolutional filters before expanding that representation back into frame segmentation through additional convolutional filters which are somewhat like ResNet analogs to morphological dilation (see Figure 15).

The advantage of an approach like this is that the pretrained components of the segmentation network can be used as input 'heads' for other neural networks which then may process sequences of frames to make predictions about the video. Recurrent neural networks (RNN) allow for processing of videos of arbitrary length [53]. In this case clips can be trimmed to periods of time where the capture quality is especially good or shortened to reduce the computation required to make as prediction.

2.7 Regression and classification

Once an ultrasound clip is processed to produce a compact, lower dimensional representation of the input, the next step is to use the new representation to classify the clip. The neural network approach is from a category of data analysis sometimes called ‘machine learning’ in which a computer is ‘trained’ on a set of input data from which it produces its own unique representation with minimal prior knowledge. The representation is then used for predictive, descriptive or automation tasks but can be as simple as a single number representing a metric like ‘volume status’.

There are two broad categories based on data and objectives: ‘Supervised learning’ is an application in which the model is trained with a specific goal. An example of supervised learning could include situations when the training data is ‘labeled’ so that the model produced can predict a label for unseen data. ‘Unsupervised learning’ is an application in which structure is learned from unlabeled data without prior knowledge; this is common in tasks where one wants to learn new information from raw data. Examples include mining social networks to determine public perception of a given product or analysis of genetic sequences to find interacting genes [54]–[56].

The algorithms used in machine learning generally fall into two categories:

1) Classification algorithms put things into groups; the groups can be used to analyze structure of existing data or to guess labels for new data.
2) Regression algorithms are fit to continuous data to produce a numerical model; these models may then be used to make numerical predictions based on new data.

Chapter 3 Methodology and study design

3.1 Participants

A cohort of 34 healthy subjects between the ages of 20 to 50 were recruited from Memorial University. Informed consent was obtained. The research protocol was reviewed and approved by the Canadian Health Research Ethics Authority. Subjects were told to breath freely as a transverse ultrasound of the internal jugular vein was recorded at the apex of the triangle made between the sternocleidomastoid and the middle scalene muscles using a SonoSite® M-Turbo Ultrasound. Clips were recorded at a depth of 4.0cm with a HFL50 linear array probe. Once the probe was correctly positioned 15 second video clips were recorded, but care was taken to ensure no identifying information was logged and all data was kept confidential.

3.2 Data collection

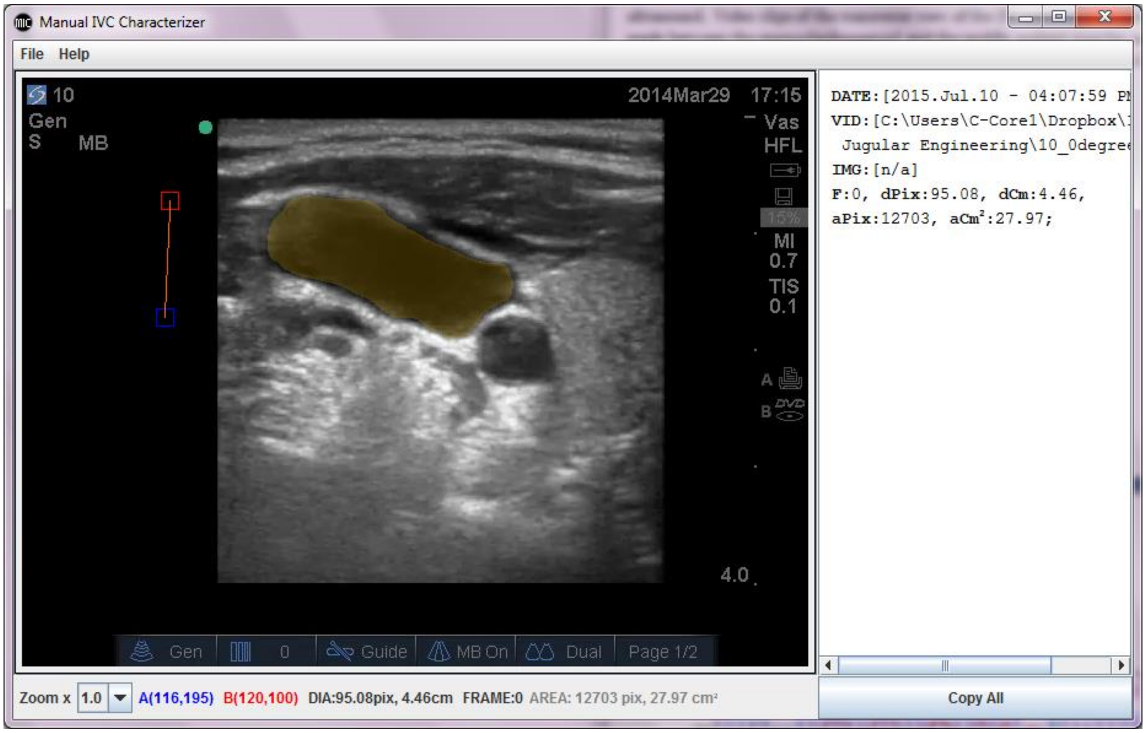
Initial video clips were recorded as the subject lay flat on a hospital bed facing the ceiling. The bed was then adjusted as additional videos were recorded at several angles until the subject was sitting with a 90 degree bend at waist with straightened legs. The change in pose gives the appearance of volume depletion as described in section 2.4.

34 subjects were recorded for a single session across the range of angles as described in Figure 8 : Different bed angles at which subject was reclined.. This cross-sectional data is intended to represent a range of different anatomy in the population.

3.3 Data annotation

Under the guidance of a clinician, the first frame of all video clips was manually segmented by an expert labeler to delineate the boundary of the internal jugular vein. For

14 subjects every frame was manually segmented using a Wacom drawing tablet to provide reference for human identification of content. All videos were also manually processed by an automated segmentation algorithm based on the clinician's segmentation of the first frame. The annotation was produced using custom software written for the task. The software is seen in Figure 17. Each frame had a 640 by 480 resolution.



```
DATE:[2015.Jul.10 - 04:07:59 PM
VID:[C:\Users\C-Core1\Dropbox\I
 Jugular Engineering\10_0degree
IMG:[n/a]
F:0, dPix:95.08, dCm:4.46,
aPix:12703, aCm²:27.97;
```

Zoom x 1.0 A(116,195) B(120,100) DIA:95.08pix, 4.46cm FRAME:0 AREA: 12703 pix, 27.97 cm²

Figure 17 : Video annotation tool produced at Memorial University of Newfoundland

3.4 Dataflow

Figure 18 outlines the flow of data and processing. The figure reads from left to right. Sample clips were taken from several dozen subjects as described above. Every frame of those clips was segmented and processed for edge, texture, moment and shape factor features with reference annotations as described above. For each sequence, temporal features were created to account for the motion of the contour. These operations produce the main training data set. The total feature set contains a lot of features, so subsets are selected based on the properties of the training data. Every predictive algorithm is then retrained with every subset of the training data. The trained models are then passed previously unseen data to generate a table of results.

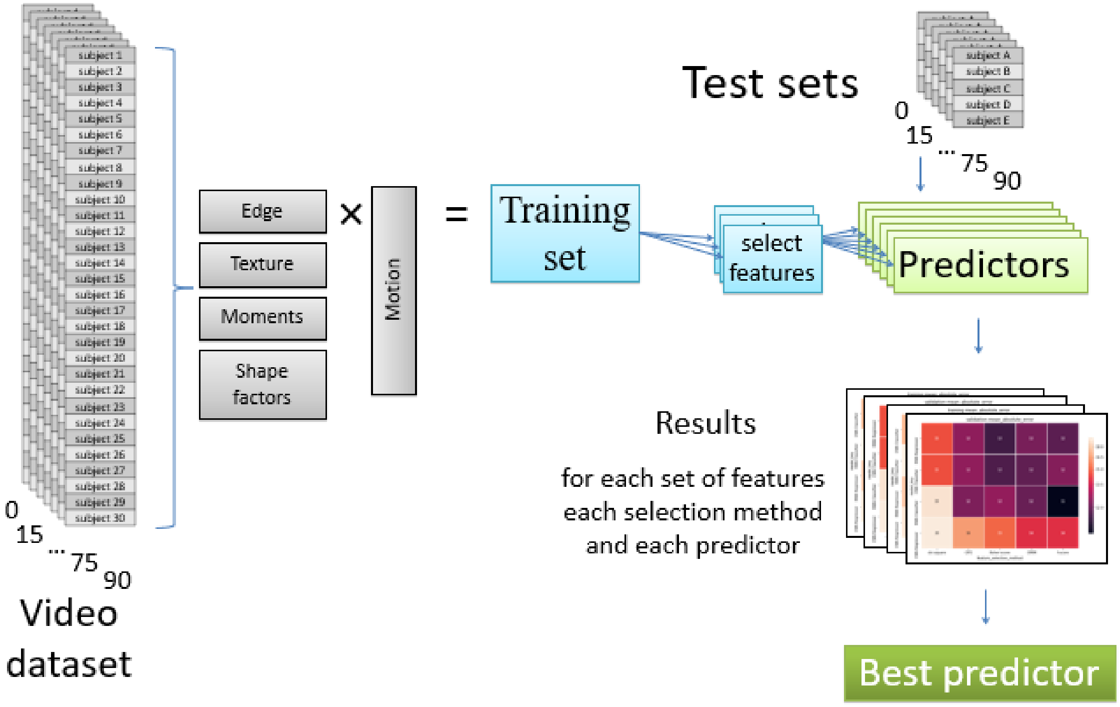


Figure 18 : Feature aggregation methods and data processing flow

3.5 Cross validation

The overall processing pipeline is described in Figure 18 for a single training set and a single testing set. However, the processing was actually repeated 3 times as part of a stratified k-fold cross validation process. Stratified k-fold cross validation splits data into training and testing datasets as follows: The data was shuffled and split into 3 groups (called ‘folds’). The analysis was performed 3 times (once per fold). For each analysis, one group is chosen for testing and the remaining data is used for training. At the end, scores from all tests are averaged. This helps reduce the possibility of outlier samples changing outcomes due to their inclusion in the testing or training set.

Figure 20 describes the total dataset. There are 35 subjects. Each subject had 15 second samples collected at different angles (0° to 90°).

		Subject ID														
		1	2	3	4	5	6	7	8	9	10	11	12	13	…	34
Bed Angle	0°	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	…	15s
	15°	-	-	-	-	-	-	-	-	-	-	-	15s	15s	…	15s
	30°	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	…	15s
	45°	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	…	15s
	60°	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	…	15s
	75°	-	-	-	-	-	-	-	-	-	-	-	15s	15s	..	15s
	90°	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	15s	…	15s

Figure 19 : Visualization of complete dataset

The subjects from 1 to 10 were randomly put into 3 groups (or ‘folds’). The subjects from 13 to 34 were also shuffled and split into 3 groups:

GROUP 1: [1, 3, 6, 9, 13, 14, 21, 25, 27, 30, 33]

GROUP 2: [2, 4, 8, 11, 12, 15, 17, 23, 24, 29, 31, 34]

GROUP 3: [5, 7, 10, 16, 18, 19, 20, 22, 26, 28, 32]

The folds are as follows, as shown in Figure 20:

a. Fold 1 uses group 1 for testing and the rest of the samples for training.

b. Fold 2 uses group 2 for testing and the rest of the samples for training.

c. Fold 3 uses group 3 for testing and the rest of the samples for training.

At the end of the 3 'folds' the results from each test set are combined.

🟥	Testing	🟦	Training

	Subject →	1	3	6	9	2	4	8	11	5	7	10	13	14	…	32
Bed Angle	0°	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	…	🟦 15s
	15°	🟥 -	🟥 -	🟥 -	🟥 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟥 15s	🟥 15s	…	🟦 15s
	30°	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	…	🟦 15s
	45°	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	…	🟦 15s
	60°	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	…	🟦 15s
	75°	🟥 -	🟥 -	🟥 -	🟥 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟥 15s	🟥 15s	..	🟦 15s
	90°	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	…	🟦 15s

(a) Fold 1

	Subject →	1	3	6	9	2	4	8	11	5	7	10	13	14	…	32
Bed Angle	0°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	…	🟦 15s
	15°	🟦 -	🟦 -	🟦 -	🟦 -	🟥 -	🟥 -	🟥 -	🟥 -	🟦 -	🟦 -	🟦 -	🟦 15s	🟦 15s	…	🟦 15s
	30°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	…	🟦 15s
	45°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	…	🟦 15s
	60°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	…	🟦 15s
	75°	🟦 -	🟦 -	🟦 -	🟦 -	🟥 -	🟥 -	🟥 -	🟥 -	🟦 -	🟦 -	🟦 -	🟦 15s	🟦 15s	..	🟦 15s
	90°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	…	🟦 15s

(b) Fold 2

	Subject →	1	3	6	9	2	4	8	11	5	7	10	13	14	…	32
Bed Angle	0°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	…	🟥 15s
	15°	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟥 -	🟥 -	🟥 -	🟦 15s	🟦 15s	…	🟥 15s
	30°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	…	🟥 15s
	45°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	…	🟥 15s
	60°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	…	🟥 15s
	75°	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟦 -	🟥 -	🟥 -	🟥 -	🟦 15s	🟦 15s	..	🟥 15s
	90°	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟦 15s	🟥 15s	🟥 15s	🟥 15s	🟦 15s	🟦 15s	…	🟥 15s

(c) Fold 3

Figure 20 : Diagram of K-Fold cross validation process

3.6 **Treatment of Outliers**

As seen in Figure 30 there are occasional ‘blips’ in measurement lasting about 1 frame. They stem from errors in manual or automatic segmentation but are recovered from quickly. If prediction is done ‘per frame’ these outliers would need to be removed. If classification is done per sequence it may be reasonable to ignore them.

The entirety of some video clips contain values far outside expectations. Figure 21 shows the variation of the cross-sectional area (CSA) in pixels for all subjects at 60 degrees. The left portion shows the pattern of CSV variation in first 150 frames. The right shows the distribution. At left there is a single sequence which falls far outside a normal distribution. The sequence is marked in black at left and a fitted normal distribution is shown in black at right.

A bad clip would be flagged by the system if the algorithm was making predictions in a real medical application: it would prompt the operator to re-record data. These were left in the dataset and may impact measurements. Given the small sample size and the possibility of a robust algorithm overcoming this they may help give confidence in the results as what is presented might be more of a lower bound on performance.

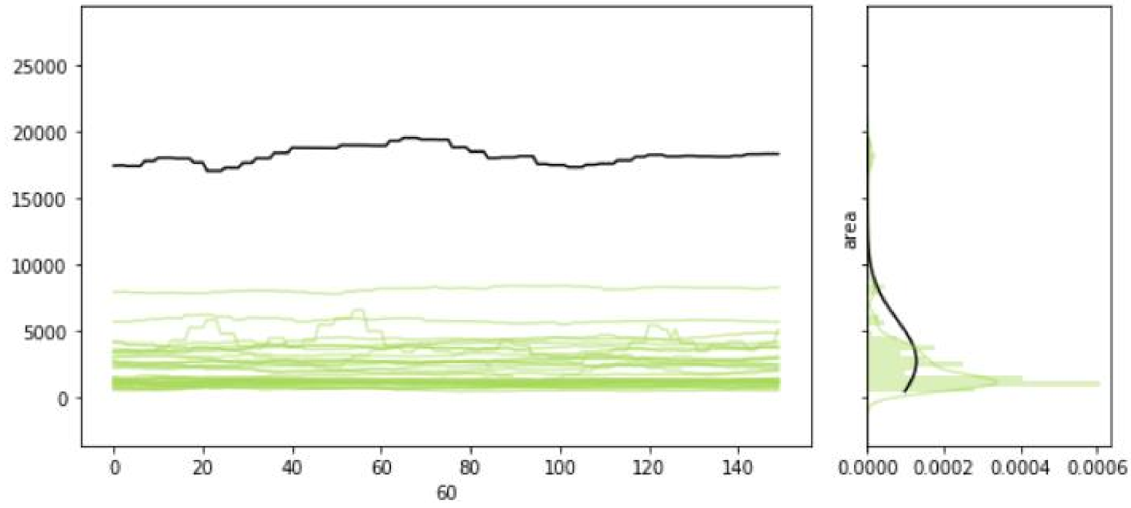


Figure 21 : Sequence outlier

3.7 **Truncating clips**

Shorter clips can be processed faster. 150 frames or 5 seconds of video should contain most of the information needed to make a prediction. This is the length used for both sequences and aggregate training/testing data.

Figure 22 shows a sample of periodograms from several clips:

- There are peaks between 1.0 and 1.67 Hz - this the range for heartrate [57]
- 1.0 to 1.67 Hz corresponds to .59 to 1 second intervals
- The normal rate of 15-30 breaths per minute [58] or a breath every 2-4 seconds can be seen in the .25 to .5 Hz range.
- Further peaks at 0 are related to steady state offsets.
- Given the repetitive nature of clips and the target of cardiovascular information 5 second clips are a reasonable length.

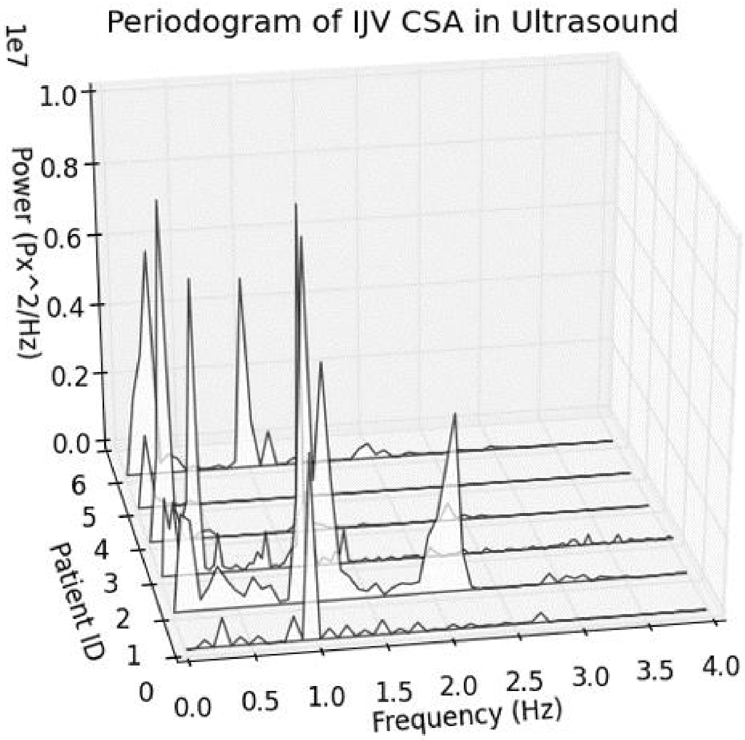


Figure 22 : Frequency components in various clips.

3.8 Display of results

Figure 23 shows a sample of the results produced during this research. The title is the dataset (in this case the validation/testing dataset) and the metric being presented (in this case the accuracy score). Each cell in the chart gives the mean accuracy score computed across all validation sets from the cross validation. The y-axis shows different types of models, such as a Recurrent Neural Network (RNN) classifier vs a Convolutional Neural Network (CNN) classifier. The x-axis shows different feature selection methods. The circled cell shows that for a particular context, when the RNN classifier was trained on features selected using f-score, the average accuracy score across 3 validation datasets was 70%.

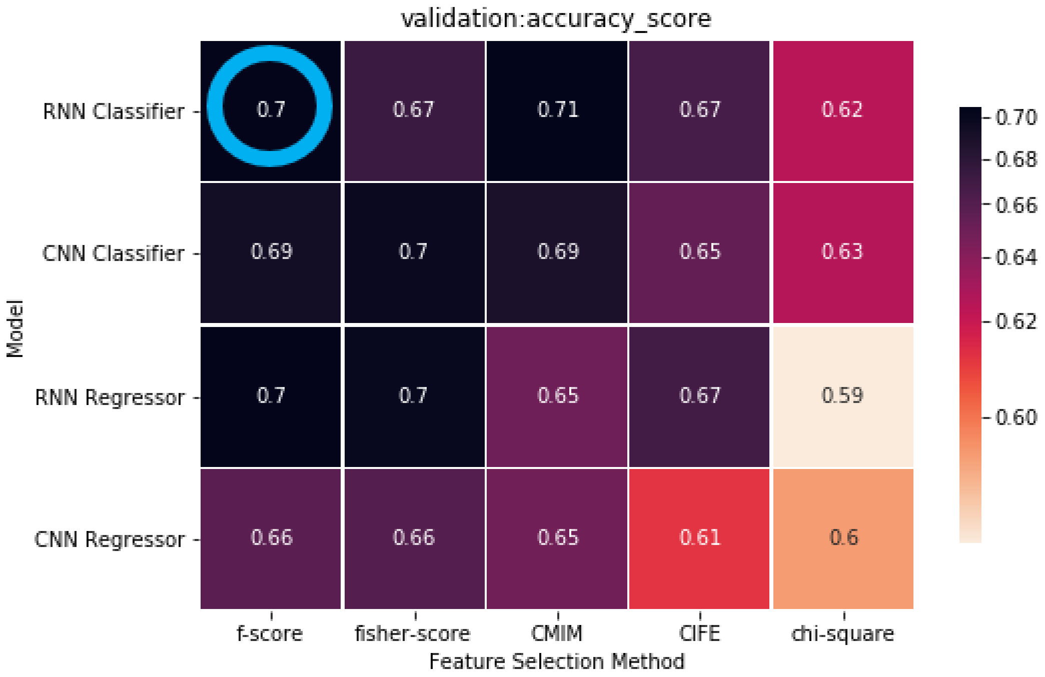


Figure 23 : Sample results table

3.9 Baselines

For reference purposes, a baseline classifier was included in the results. The baseline randomly returns one of the target classes. The implementation comes from scikit-learn's DummyClassifier [59], it is denoted 'RandomGuess Classifier' in the results. Also, a reference for the absence of feature selection shown as 'None'.

Chapter 4 Features

4.1 Structure

In performing an ultrasound, a physician will begin by assessing what is to be studied and placing the ultrasound transducer based on external anatomy references (such a viewing between ribs). Then, looking at the ultrasound screen, they will assess internal 'landmarks' to make fine adjustments to probe placement based on the anatomy they see.

Anatomy and the layout of shapes on the screen are very important in this initial assessment. If organs are in strange orientations or of abnormal size these characteristics may be important in diagnosis.

Common ways to describe shapes such as Hu moments [60], [61] and shape factors [62]–[65] can be applied to segmented ultrasound images. The resulting numerical descriptions of orientation, location, size, and shape of the target anatomy may then be used to train a machine learning model which can separate the different groups of video clips.

As an example, the simplest shape descriptor is 'size of the shape'. Figure 24 shows the distribution of the size (cross-sectional area) of the internal jugular vein is different for the subjects sitting upright at 90 degrees (top of Figure 24) vs those who are lying horizontally (bottom of the Figure 24). However, the distributions have multiple peaks; using other descriptors or combining descriptors may give a cleaner separation of the groups.

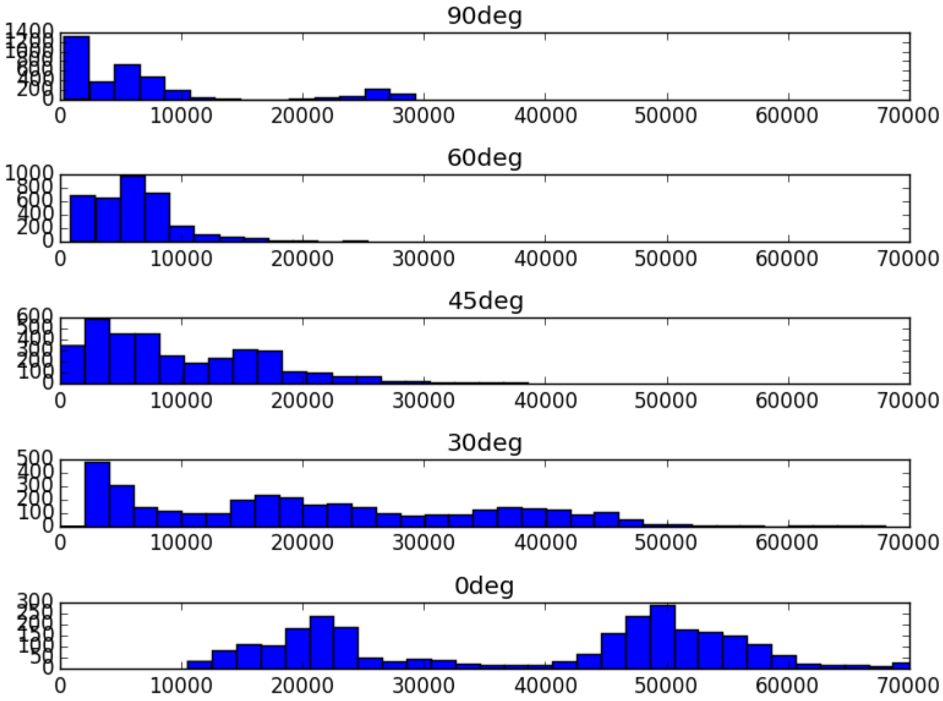


Figure 24 : Distribution of Cross Sectional Area (in pixels) with respect to subject's torso angle in degrees

4.2 Statistical moments

In 1962 Ming-Kuei Hu[60] published a set of 7 abstract descriptors for 2D shapes. By moving around the boundary of a shape and computing an integral you can find the area of the shape. By expanding this idea, Hu created a 'moment generating function' from which several equations were derived that describe not just the area, but distribution and variance of the 'mass' of the shape. They were effective because they still gave the same numbers if you applied transformations to the image like scaling or translation.

A moment is just an n-dimensional calculation computed around an axis. Any object in space has a unique distribution of mass that determines properties like center of gravity and around which axis is most difficult to spin the object (moment of inertia). These properties would be the same even if you looked at a miniature version of the original object, or if you moved the object or if you rotated the object. By applying the same thinking to a distribution of pixels you get transformation invariant descriptors that are useful for object detection.

Jan Flusser and Tomas Suk [66] expanded on Hu's work to add the rotation invariance, add to the list of moments and provide a general formula for creating these descriptors in any dimension or quantity.

For a 2 dimensional continuous function (contour):

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \rho(x,y) dx\, dy \tag{4}$$

Where p, q = 0,1,2,3 … N

For a 2 dimensional discrete function (contour):

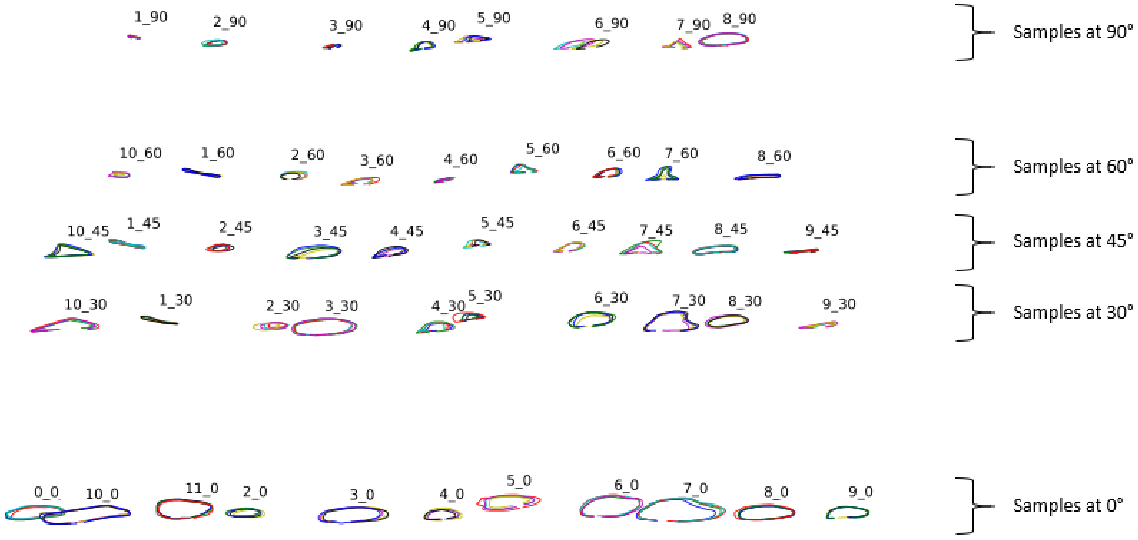
$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \tag{5}$$

This will give several 'raw' moments which have unique meaning. For example, area is given by M_{00}. A bit of math produces an additional set of moments called standard or 'central' moments that are not affected by scale. Further processing produces a third set which are the 'Flusser' moments that are not affected by rotation and several other transforms.

This complete set of moments is very descriptive but rather abstract. However, a more intuitive set of descriptors may be derived from these moments.

4.3 Shape factors

As seen in Figure 25, samples at 0° show a full vessel that is ‘round’, but at higher angles collapses into more angular shapes. When describing the shape of a vessel it is intuitive to describe it as ‘round’, ‘circular’, ‘squared’ or ‘triangular’. Using a ‘shape factor’ formula on a contour produces a number describing how similar it is to a given shape. This has the advantage of being understandable by a human compared to Hu and Flusser moments with names like ‘nu02’ describing abstract properties of the curve. As seen in Figure 26 these same shape factors can still be used to find distinct groups of contours. Using these descriptors can allow for models which both categorize samples correctly, but which are also interpretable.



* Identifiers are written as ‘SubjectId_Angle’

Figure 25 : IJV contour with respect to angle

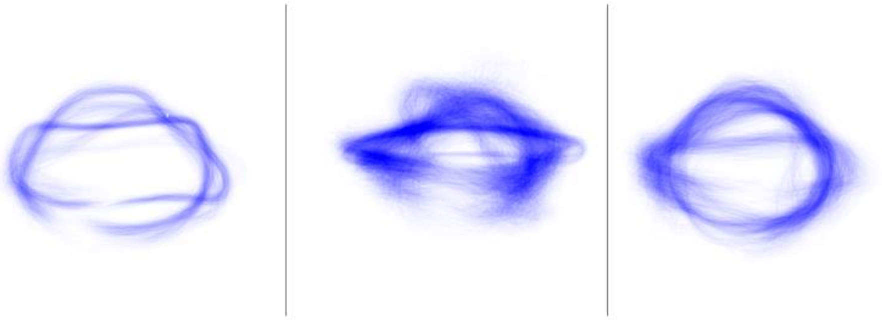


Figure 26 : Distinct shape groups found by clustering shape factors

A reasonable starting point for describing a given shape might include how circular or triangular or elongated an object is. A simple way to do this would be to compare measurements of an arbitrary object to the measurements for simple shapes of similar size. For example: a circle is defined by its radius (r) to give:

$$\mathit{perimeter}, \mathrm{P} = 2\pi\mathrm{r} \tag{6}$$

$$\mathrm{area}, \mathrm{A} = \pi\mathrm{r}^2 \tag{7}$$

Solving for r, a circle has the relationship

$$\frac{\mathrm{P}^2}{\mathrm{A}} = 4\pi \tag{8}$$

If the area and perimeter of another object is known, deviation from the number 4π would measure circularity or compactness. M_{00} from equation (5555) gives the area (A) for a segmented object. The perimeter (P) is often captured while processing the elements of the contour. Therefore, in a range from 0 to 1:

$$compactness = \frac{4\pi M_{00}}{P^2} \tag{9}$$

The same process can be applied to other shapes. By comparing a shape to an ideal shape it is possible to create measures of ellipticity, rectangularity, triangularity and others. One method of doing this which gives values from 0 to 1 is to find the minimum size of ideal shape that will still fit around the object and dividing as follows:

$$bounding\ shape\ index = \frac{M_{00}}{area\ of\ minimum\ enclosing\ shape} \tag{10}$$

The algorithm used in this research for finding bounding circles (for circularity) was given by Emo 1991 [67]. The algorithm for triangles given by O'Rourke 1986 [68]. The algorithm for rectangles given by Toussaint 1983 [69].

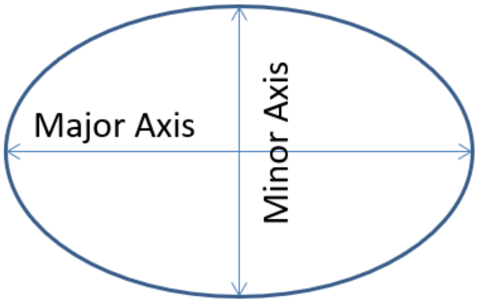


Figure 27 : Major vs Minor Axis of a circle

Recall the Maximum Moment of Inertia – the axis about which is most difficult to spin the object. In an ellipse this is called the major axis and is the longest straight line through the shape, as shown in Figure 27. At a right angle to this axis is the minor axis and is the shortest straight line through an ellipse. This is only true for an ellipse however all objects have minimum and maximum moments of inertia which pass through their center of mass. Moments of inertia may not define the actual length or width of the object – consider a triangle. In this case bounding ellipses or rectangles may also be used.

4.4 Texture

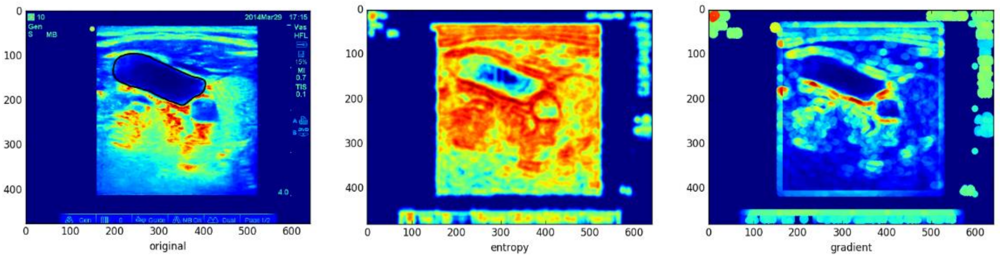


Figure 28 : Local texture feature differences from the IJV dataset collected as part of this research

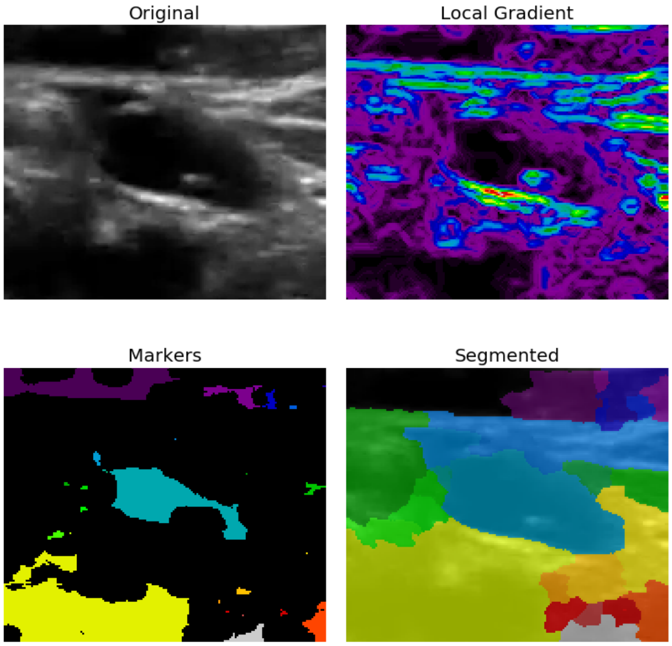


Figure 29 : Texture-based segmentation of the IJV dataset

Tissue may have a variety of different texture characteristics, even when shown in a similar shade of grey (see Figure 10). As the target vessel collapses the interior boundary of the vessel becomes indistinct and may be easily lost. However, the vessel walls are still present in the region of interest. Using texture changes to describe the region may still be effective for classification when segmentation is difficult. As seen in Figure 28 processing images to show entropy and gradient-based texture differences can allow for a more detailed description of vessel boundaries and surrounding tissue. In Figure 29 a gradient-based approach combined with segmentation techniques can provide distinct coloring for neighboring anatomy where boundaries may be unclear in a simple greyscale representation. Each of the texture descriptors from equations 11 to 21 below were applied to every frame of this dataset. These texture descriptors may help a machine learning based approach remain accurate when segmentation quality is not ideal.

Grey Level Co-ocurrance Matrices (GLCM) have been used for texture identification in many ultrasound applications [70]–[73] are a structure devised by Haralick et. al. which displays the spatial relationship between pixels of a given value using a two-dimensional histogram. This matrix can be computed for any given region and summarized with a single number in a variety of ways[74]. Referring to the formulations used by Scikit-image [75], [76] these are:

$$\mathrm{contrast} = \sum_{\mathrm{i,j}=0}^{\mathrm{levels}-1} \mathrm{P_{i,j}}(\mathrm{i}-\mathrm{j})^2 \tag{11}$$

$$\mathrm{dissimilarity} = \sum_{\mathrm{i,j}=0}^{\mathrm{levels}-1} \mathrm{P_{i,j}}|\mathrm{i}-\mathrm{j}| \tag{12}$$

$$\text{homogeneity} = \sum_{i,j=0}^{\text{levels}-1} \frac{P_{i,j}}{1 + (i-j)^2} \tag{13}$$

$$\text{angular second moment} = \sum_{i,j=0}^{\text{levels}-1} P_{i,j}^2 \tag{14}$$

$$\text{energy} = \sqrt{\text{angular second moment}} \tag{15}$$

$$\text{correlation}, C = \sum_{i,j=0}^{\text{levels}-1} P_{i,j}\left[\frac{(i-\mu_i)\left(j-\mu_j\right)}{\sqrt{(\sigma_i^2)\left(\sigma_j^2\right)}}\right] \tag{16}$$

where $P_{i,j}$ is the value of the glcm matrix at index (i,j) and levels refers to the range of the intensity histogram. Additional measures include [77]:

$$\text{entropy} = \sum_{i,j=0}^{N-1} -P_{ij}\ln\left(P_{ij}\right) \tag{17}$$

$$\boldsymbol{mean, \mu} = \sum_{\mathbf{i,j=0}}^{\mathbf{N-1}} \mathbf{iP_{ij}} \tag{18}$$

$$\text{variance}, \sigma^2 = \sum_{i,j=0}^{N-1} P_{ij}(i-\mu)^2 \tag{19}$$

$$\text{shade} = \text{sign}(A)|A|^{1/3} \text{ where } A = \sum_{i,j=0}^{N-1} \frac{(i+j-2\mu)^3 P_{ij}}{\sigma^3\left(\sqrt{2(1+C)}\right)^3} \tag{20}$$

$$\text{prominence} = \text{sign}(B)|B|^{1/4} \text{ where } B = \sum_{i,j=0}^{N-1} \frac{(i+j-2\mu)^4 P_{ij}}{4\sigma^4(1+C)^2} \tag{21}$$

As depth increases sound waves are absorbed and get weaker - images become less sharp with depth. Also, with finite resolution, small targets may become distorted. As the target vessels may change in size and location their edges may change in a measurable way. Contour acutance (the sharpness across the boundary) can be measured from the image gradient, entropy across the boundary and mean intensity. This could be done by computing each metric with respect to the normal of the contour at each point along the boundary as described by Rangayyan et. al.[64]. Given the discrete nature of digital images, another approach would be to filter the region of interest to compute local entropy and gradient with a small (5 pixel) structuring element, then sum the result at each point. Equations for mean and entropy are given above, local gradient is given by the difference between the local maximum and local minimum.

4.5 Temporal features

Jugular vein 'collapsibility' index is correlated with central venous pressure [45], [46]. This measure is related to the maximum and minimum size of the jugular vein as it pulses in time. As seen in Figure 30 the descriptors we have collected for each frame change in time and there is a different pattern for change for each angle.

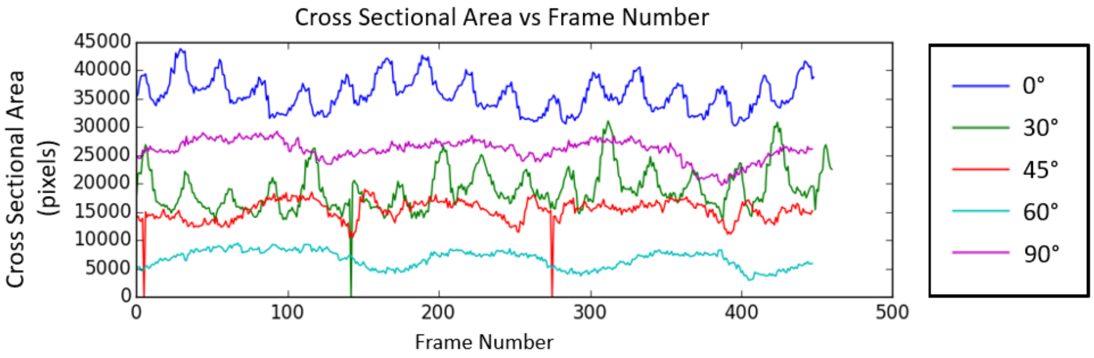


Figure 30 : CSA change with time and angle. Per frame outliers can be seen at 0, 150, 275

Rather than make a prediction on single frame measurements it's likely beneficial to combine multiple measurements. The question becomes 'which method of aggregation provides the most value?' The standard formula [78] from equations (22) to (37) were used to summarize the per-frame features of the dataset. Given $\boldsymbol{X} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{N}\}$

$$sum(X) = \sum X \tag{22}$$

$$max(X) \tag{23}$$

$$min(X) \tag{24}$$

$$mode(X) = most\ common\ value(X) \tag{25}$$

$$median(X) = \frac{X_{\lfloor (N+1)\div 2 \rfloor} + X_{\lceil (N+1)\div 2 \rceil}}{2} \tag{26}$$

$$mean(X) = \mu = {}^{\sum X}/_{N} \tag{27}$$

$$range(X) = max(X) - min(X) \tag{28}$$

$$percentrange(X) = range(X)/mean(X), \quad \text{where } mean(X) > 0 \tag{29}$$

$$variance = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{30}$$

$$standard\ \ deviation, \sigma = \sqrt{variance} \tag{31}$$

$$skewness(third\ \ moment) = \frac{E[(X-\mu)^3]}{(E[(X-\mu)^2])^{3/2}} \tag{32}$$

$$kurtosis(fourthmoment) = \frac{E[(X-\mu)^4]}{(E[(X-\mu)^2])^{4/2}} \tag{33}$$

$$coefficient\ of\ variation = \sigma/\mu \tag{34}$$

$$coefficient\ of\ dispersion = \sigma^2/\mu \tag{35}$$

$$signal\ to\ noise = \mu/\sigma \tag{36}$$

$$root\ \ mean\ \ square = \sqrt{\frac{1}{\mathrm{n}}(\mathrm{x}_1^2 + \mathrm{x}_2^2 + \cdots + \mathrm{x}_\mathrm{n}^2)} \qquad (37)$$

4.6 Results

Many plausible features were generated for each frame, however if features are similar, they may not contribute anything once the first of the related features is included. For limited numbers of observations degenerate matrices can cause issues with some kinds of regression.

A correlation matrix is a way to visually inspect a large dataset for related features; one is shown in Figure 31. This figure shows all features on both axes in the same order. Each cell shows the absolute value of Pearson's correlation coefficient for the pair of features. Related features will have a value close to 1 and are bright, unrelated features have a value close to 0 and are dark. The features are ordered by the sum of correlations across the row. Since the features would repeat, the cells above the diagonal are omitted.

In Figure 31 we can see that a contour's cross sectional area is related to the contour's perimeter, as we might expect. Their intersection is very bright and close to 1. Their rows make a band of similarly alternating colors. We could probably use only one of these features and have good results.

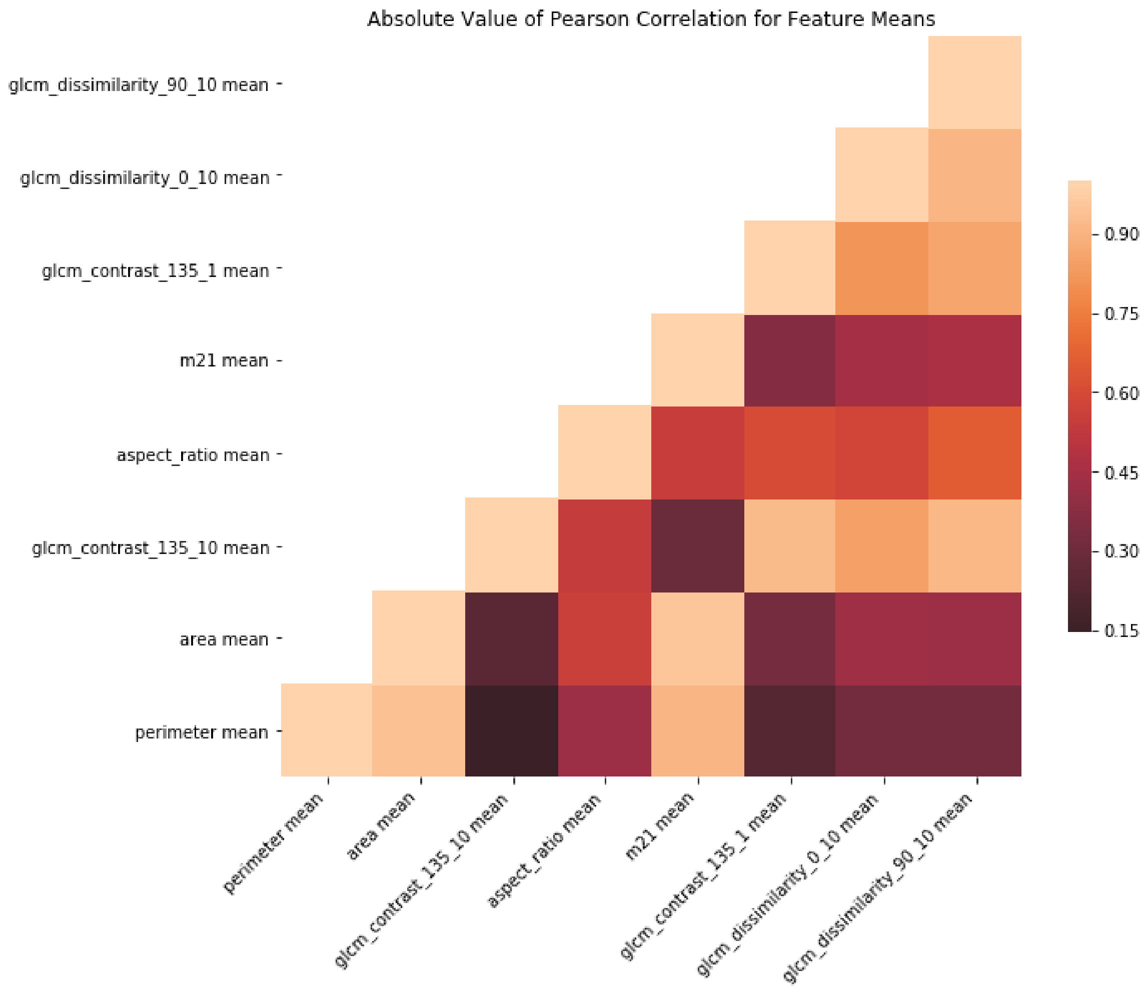


Figure 31 : Correlation matrix for a random selection of features

For larger datasets assessment is qualitative. A bright matrix is bad, a dark matrix is good and there should not be bands of repeating columns or rows. A correlation matrix for the entire feature set is given in Figure 32. Labels removed as there are hundreds of features being shown but there are many redundant features. After aggregation methods are applied, the number of features expands dramatically. Correlation matrix for this data is shown in Figure 33. Clearly some feature selection is needed.

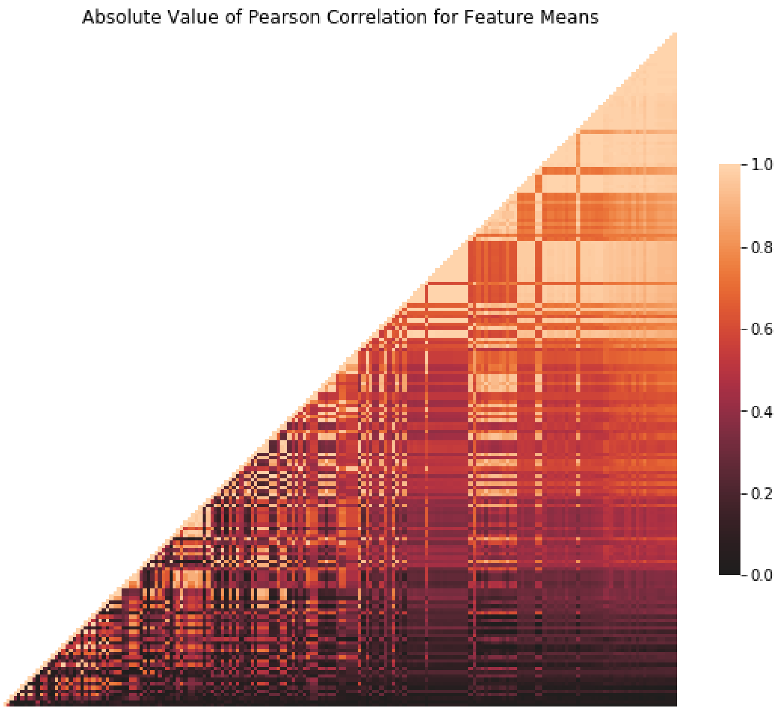


Figure 32 : Correlation matrix for mean of each feature descriptor

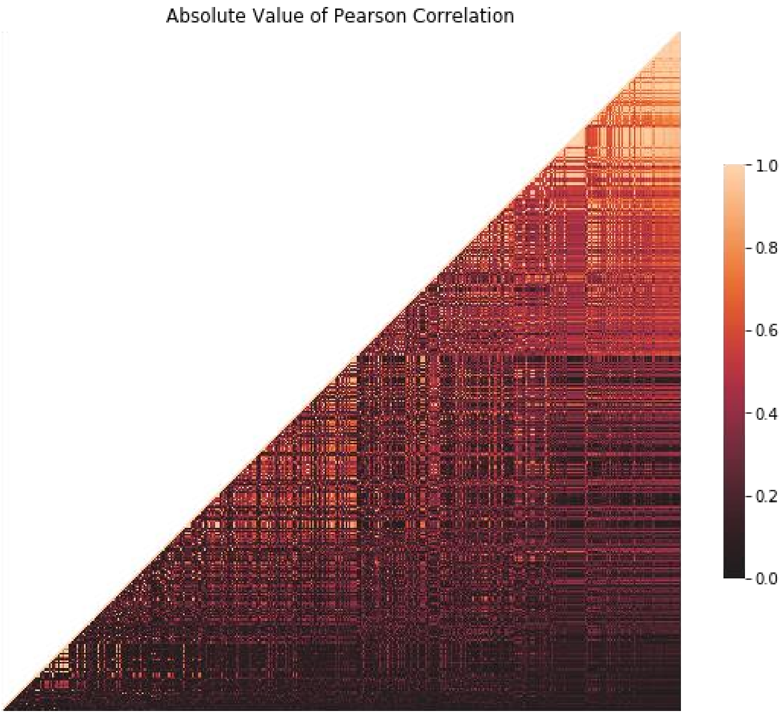


Figure 33 : Correlation matrix for every feature's temporal aggregates

For each feature, several aggregating functions were applied. For example, given the feature ‘cross sectional area’ the mean was taken across all frames to generate a new feature ‘mean cross sectional area’, the maximum was taken to give ‘max cross sectional area’ and so on all under the umbrella of ‘cross sectional area’. Figure 34 shows the weighting for base features according to f-score with respect to each of their aggregates. As shown in later figures f-score is a reasonably effective feature selection method for this problem and the top performing set of predictive algorithms. This ranking does not take the estimator into effect.

Figure 35 shows the use of the top 30 features used in an XGBoost classifier which is predicting subject angle using the entire feature set. XGBoost like other boosted trees must choose features on which to split the dataset according to the improvement each feature gives to the model. However, direct measurement of feature importance using gain, coverage and frequency as defined by the gradient boosting tree can be inconsistent or irrelevant to predictions [79], [80]. Feature attribution using model agnostic explanation algorithms with multiple attribution models aligns much better with human intuition [80]–[82].

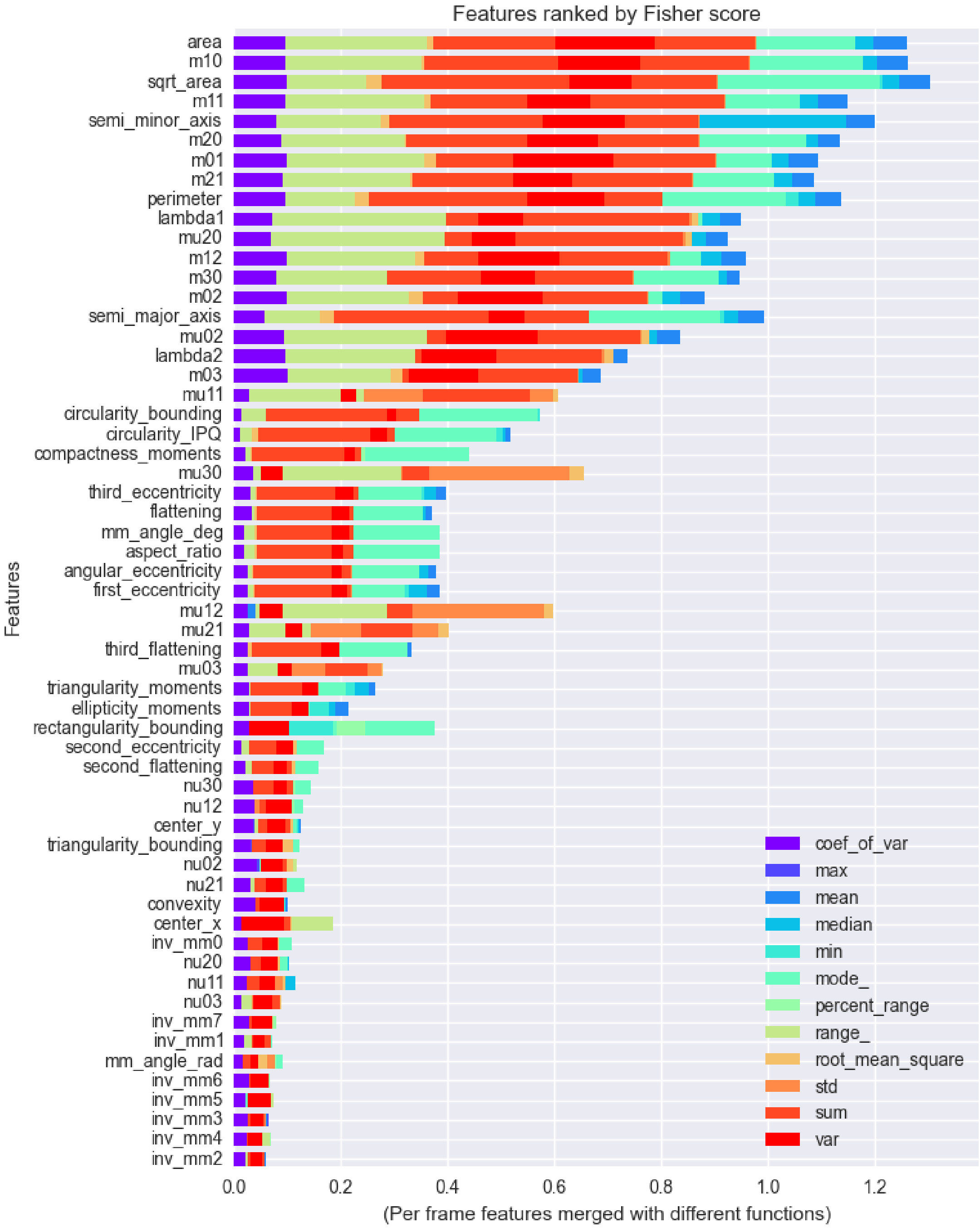


Figure 34 : Feature Rank by Fisher Score

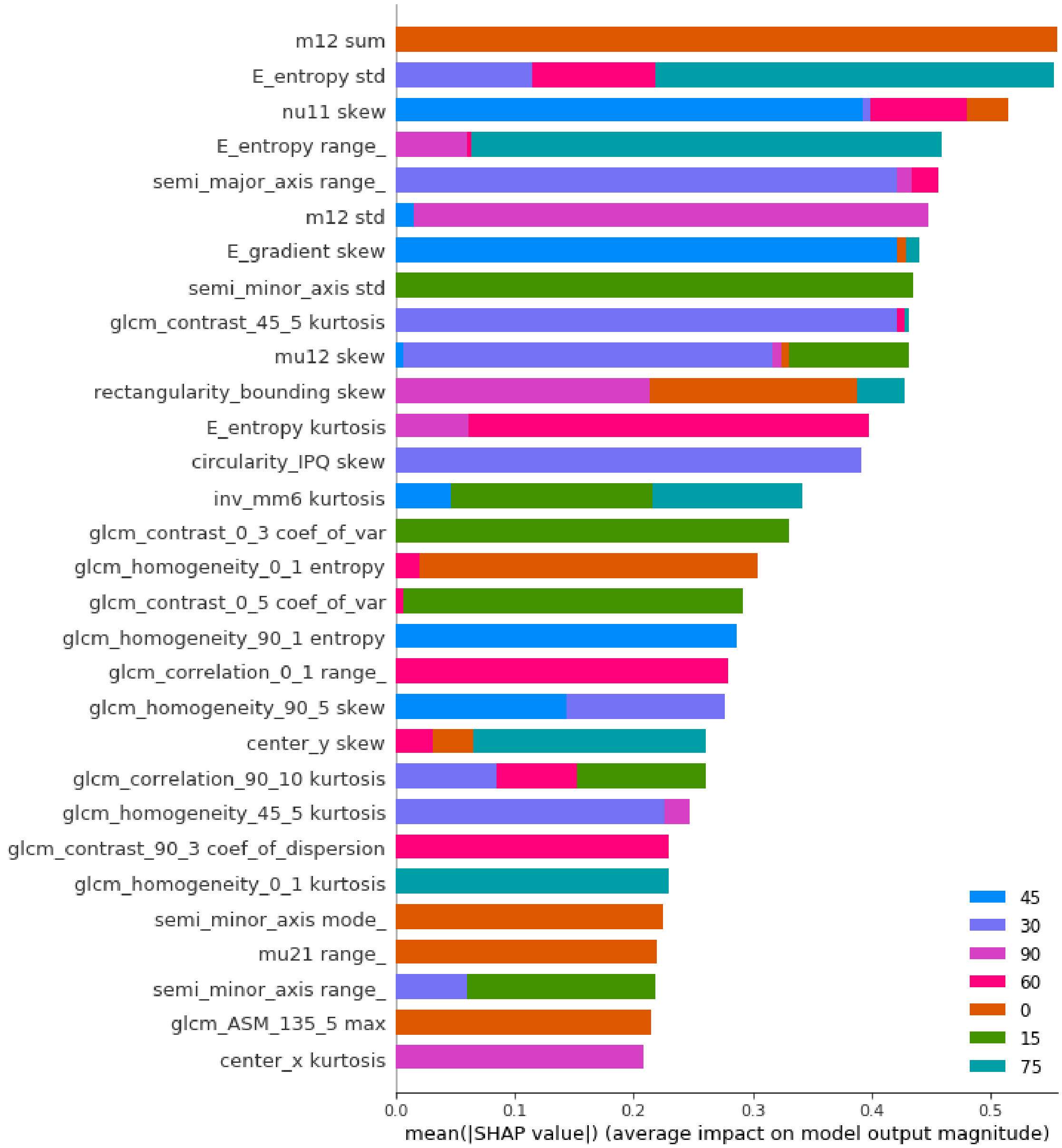


Figure 35 : SHAP feature rank for XGBoost

4.7 Discussion

As shown in the referenced publications [25], [26], [83] any of the subsets of features (Statistical moments, shape factors, texture, motion) are sufficient to produce good binary classification accuracies. The original hypothesis of the utility of IJV CSA appears to hold merit as it is chosen in several of the best feature selection methods and used as a prominent feature in self-pruning algorithms. Adding size and shape to the mix can further improve accuracy – as seen in Figure 64 they may help with producing understandable prediction explanations. Texture features appear very promising as the top accuracy in a single fold was achieved using only GLCM_ASM_135_2 and Grey Level Co-occurrence Matrices rounded out several of the estimators.

However, as seen in Figure 35 it is possible different features become more useful in different contexts and no single category of features outshines all others. Several features appear to be the sole deciding factor in that particular GBM's ability to delineate certain angles. Intuitively it appears that low angles primarily using semi minor axis, mu21, mu12 might be due to the visibility of a full IJV while E_entropy measurements used by the high angles where the IJV could be completely collapsed are an attempt to rely on texture and a lack of black pixels when the IJV is barely visible.

Features selection via manual description with aggregation is quite computationally and developmentally intensive, while the neural network approaches consumed the sequence and even generated reasonable representation of their own with very little effort. They appear to be a superior approach, but as seen in Figure 25 : IJV contour with respect to angle vs Figure 62: Activation of CNN layers, perhaps the simple shape features are more intuitive and interpretable.

Chapter 5 Feature selection methods

There is a phenomenon called the curse of dimensionality which comes from the fact that for any finite number of observations about a population, there are many perspectives or dimensions which can describe the dataset [84]. There are several impacts. For one, if the number of features used to describe the observations increases, more observations are required to fully describe the population. For another, each new feature represents an input to an algorithm. All algorithms have a complexity defined by the size of their input, hence increasing the amount of input increases the amount of computation required to use the algorithm. There are other impacts, but the result is that the requirements exponentially increase with dimension and it is best to reduce it by weighting the inputs or removing less important features.

There is also a theorem about optimization algorithms called ‘No Free Lunch”[85], which suggests that no algorithm can out-perform an exhaustive search of data treatments and estimator architectures. A model must be fit to a dataset or vice versa. There are lots of possible ways to transform data to extract knowledge and it is possible to modify features to be more useful to a given algorithm. Intuitively – a categorizer might do well with category features and a predictor of numbers might do well with continuous numerical features.

The article “Feature selection: A Data Perspective” provides a fairly comprehensive review of feature selection methods [86]. It provides a categorization for the methods which I use below and a useful python library called ‘scikit-feature’[87], [88].

5.1 Similarity-based feature selection methods

As mentioned in many texts on the topic, a simple way to select features is to rank them by their relationships with other features [89], [90]. This method is tightly coupled with popular statistical methods of regression which use relative variance with respect to a target variable to fit models. The Fisher scoring algorithm uses the expectation or the variance of a feature with respect to others [91]. It is a brute force operation applied to all features and may be computationally expensive in high dimensions, but is very effective. The Trace Ratio is similar, but iteratively optimizes a subset-level score for features in a more efficient way [92]. There are several other methods which use similarity such as ReliefF [93]. The following methods were each separately applied to the dataset to reduce the number of features as implemented in [88] and described in Figure 18 : Feature aggregation methods and data processing flow:

- Trace Ratio [92]
- Fisher Score [91]

5.2 Information-theoretical-based feature selection methods

Instead of optimizing for relative variance, it is possible to use other measures of information. Claude Shannon defined measures of 'entropy' using sums of the logarithm of probability [94]. Differences in the entropy measurement for a feature can be used in much the same way as variance and are used in several decision tree algorithms to split datasets. Techniques from this category which were applied to the dataset are given below, as implemented in [88]:

- MIM - Mutual Information Maximization [95]

- MRMR - Max-Relevance Min-Redundancy [96]
- MIFS - Mutual Information Feature Selection [97]
- CIFE - Conditional Infomax Feature Extraction [98]
- JMI - Joint Mutual Information[99]
- CMIM - Conditional Mutual Information Maximization [100]
- ICAP - Interaction Capping [101]
- DISR - Double Input Symmetrical Relevance [102]
- FCBF - Fast Correlation-Based Filter [103]

5.3 **Statistics-based feature selection methods**

The central limit theorem suggests that sufficiently large volumes of observations will tend toward a normal distribution of sample means, the impact of which lead to a great deal of statistical techniques [104] By assessing features using traditional statistical tests instead of learning algorithms features with low predictive power, low contribution to the explanation of variance, low variance, low correlation with the target or independence from the target can be removed. The following scores were used as implemented in [59], [88]:

- Chi-square score [105]
- F - score [106]
- CFS - Correlation-based Feature Selection [107]

5.4 **Feature selection with streaming data**

Streaming data implies that not all data from a sample is available in advance of calculation, meaning that several of the previous methods cannot be used. Using an adaptive threshold for the inclusion of features allows for streaming data to have feature selection applied. The following technique was applied to the dataset as implemented in [88]:

- Alpha-investing [108]

5.5 Classifier specific wrapper feature selection methods

Feature selection methods select features for their own criterion, but the preferences of an algorithm might differ. As some algorithms such as decision trees and Support Vector Machines (SVMs) allow measurement of their use of a feature, such as hierarchal importance in making classification decisions, proximity to a margin or degree of separation achieved by a feature. By iteratively removing features from the total set which contribute the least (backward feature selection or recursive feature elimination) or by iteratively adding features to the total set (forward feature selection) the features can be better aligned with the predictive algorithm. The difficulty in this approach is that it requires retraining the model for each feature under consideration which can be extremely computationally intensive. The following were tried, as implemented in [88]:

- Forward Feature selection using a decision tree [109]
- Forward Feature selection using an SVM [109]
- Backward feature selection using an SVM [109]

5.6 Results

After an effective feature selection is applied the correlation of features might be expected to decrease. As an example, the effect of reducing correlation using Fisher

Score is shown in Figure 36 and it seems reasonably effective. The effect of using Chi-Square for feature selection is shown in Figure 37 for comparison.

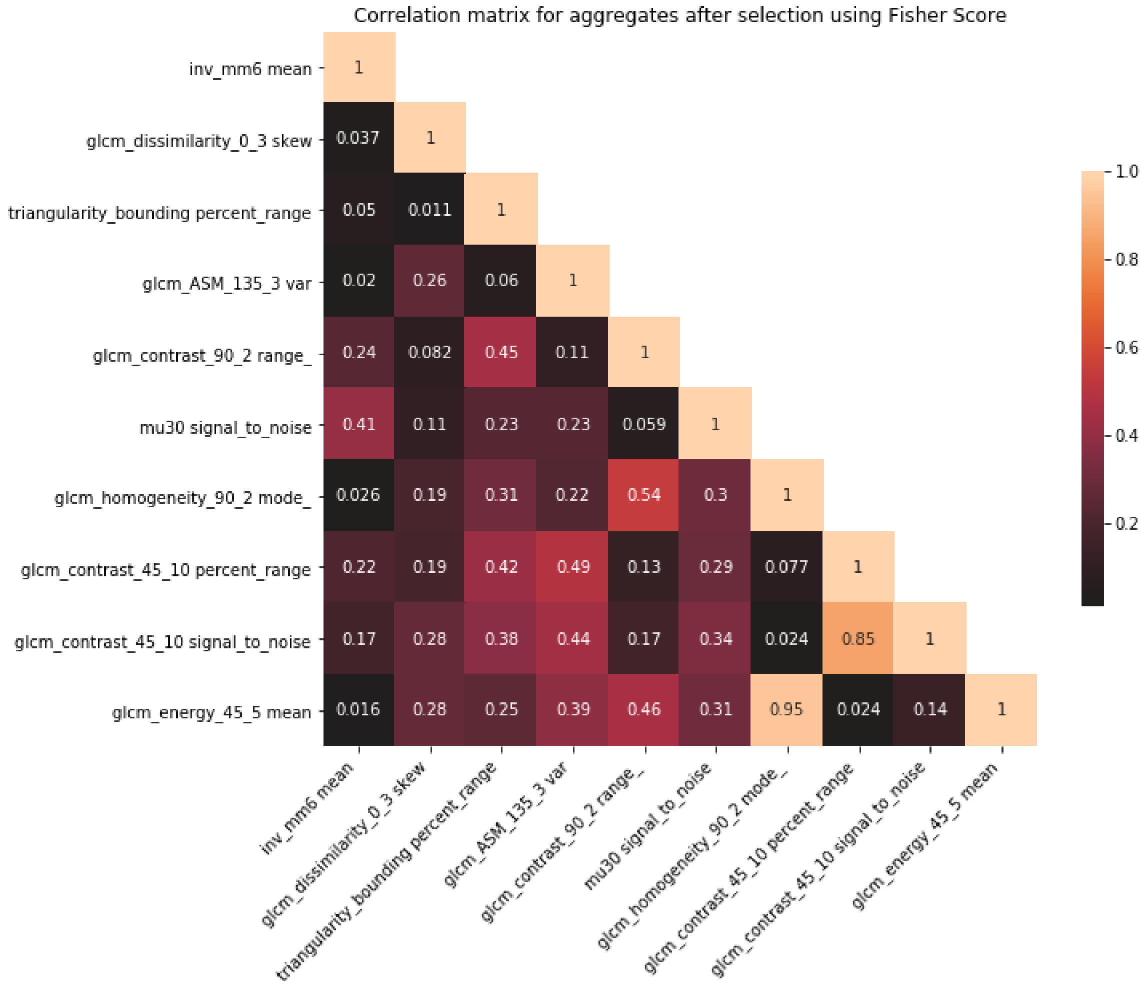


Figure 36 : Feature correlation after Feature Selection using Fisher Score

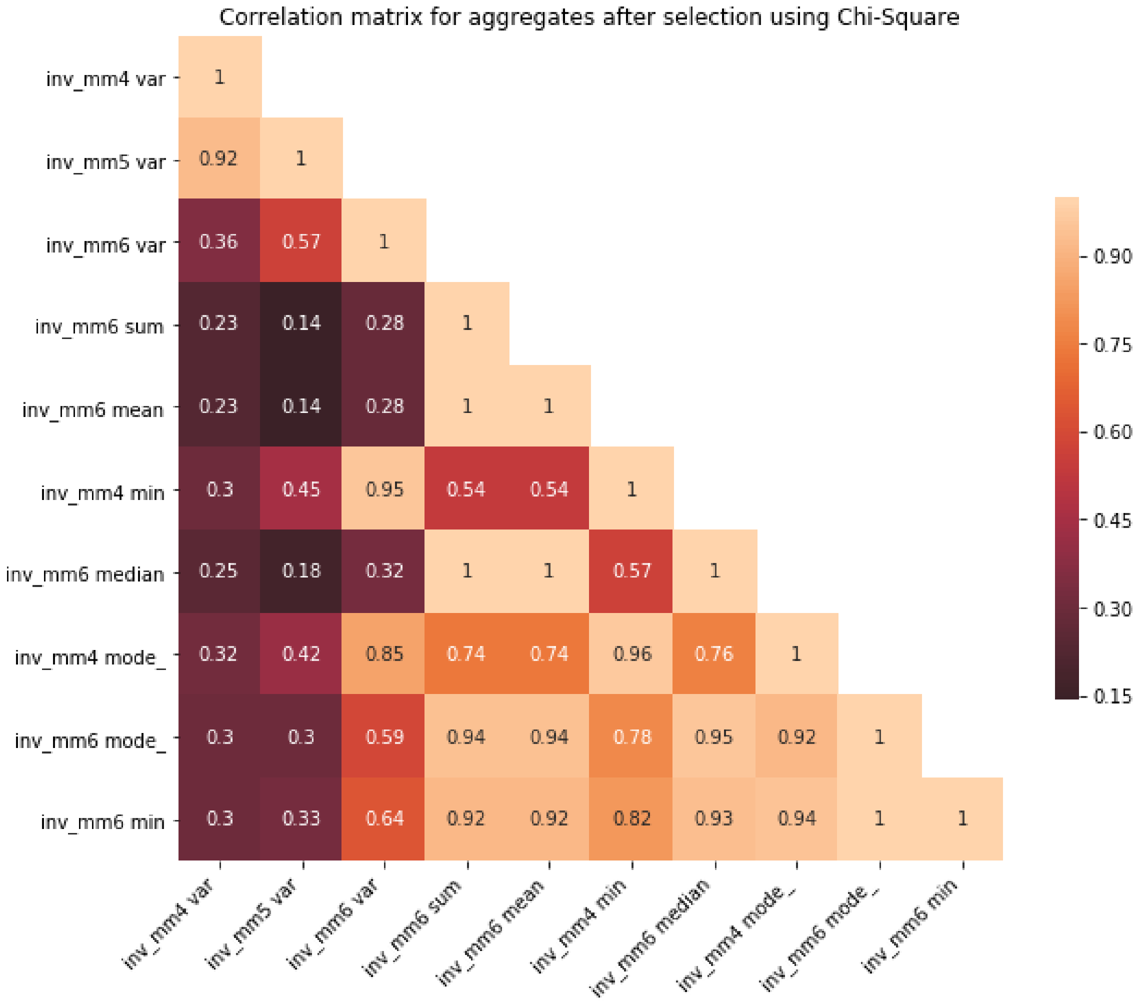


Figure 37 : Correlation matrix after Feature Selection using Chi-Square

Feature selection methods were assessed by comparing performance across all estimators. Since the prediction target is an angle Mean Absolute Error (MAE) can be measured regardless if a classifier or regressor is used to make a prediction. Performance with respect to the MAE for all estimators is shown in Figure 38.

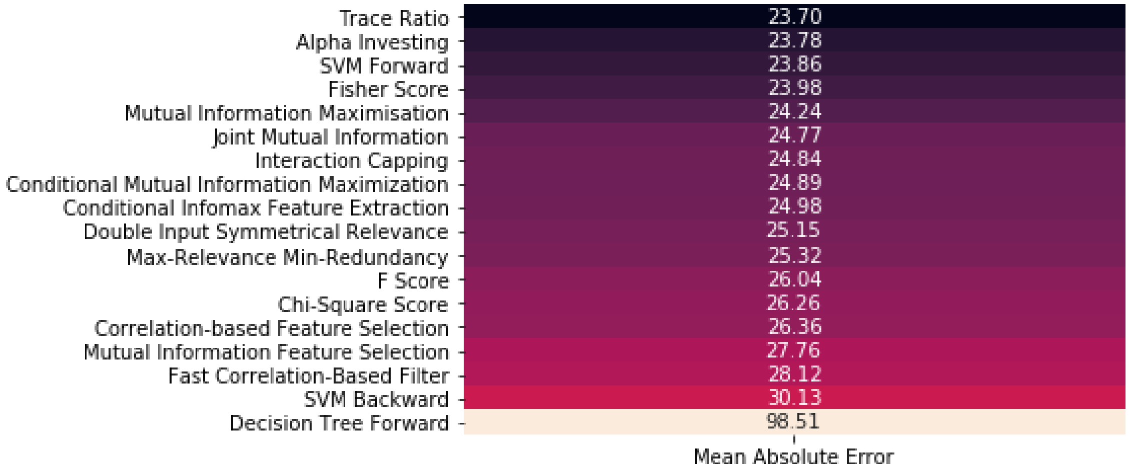


Figure 38 : Mean Absolute Error with respect to feature selection method

As practical concerns regarding training time are important considerations for the use of a method average training time for each feature selection method is shown in Figure 39 for each method as implemented in scikit-feature [88] and run on a computer with 4 CPU cores, 16GB RAM. As forward and backward feature selection methods proved quite slow in some cases (not completing after over a day), these methods were not suitable for other tests. Also, some methods were either not robust in their applicability to all data considerations or produced similar feature subsets; these methods were also not considered past this stage of testing.

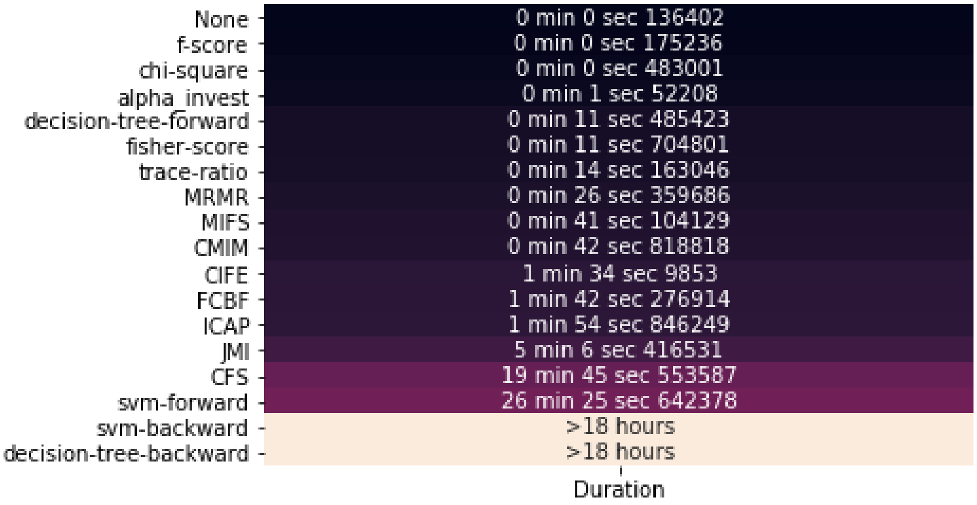


Figure 39 : Training time for each feature selection method

5.7 **Discussion**

As seen in Figure 38 and Figure 55 Fisher Score and f-score feature selection methods produce good predictions while still being quick to select as seen in Figure 39. Originally after feature generation there were a great deal of redundant features. As seen in Figure 36 these methods reduce redundant features – this may explain their usefulness compared to Chi-Square seen in Figure 37 selecting highly redundant features which give poor performance.

However, ‘None’ still produces reasonable results in algorithms such as gradient boosted trees which have heavy internal feature selection and regularization. The effect is likely amplified by the relatively small size of the dataset compared to the number of features. While useful in this situation perhaps with a very large dataset prior feature selection would become less important.

Chapter 6 Models

While some categories of predictive algorithms become popular or fade with time there is no perfect approach to all problems. The ‘No Free Lunch’ theorem suggests that for each problem a predictive algorithm is good at, there are several datasets on which is will perform poorly [85], [110]. Due to this problem there have been a large variety of predictive algorithms created. They each have different levels of applicability to a given problem and some level of trial and error may be required to find an appropriate model for a dataset.

Several of the techniques used in this research are explained in Friedman, Tibshirani, and Hastie's Elements of Statistical Learning [111] and are implemented in the python machine learning framework called Scikit-Learn [59] which was used in this analysis.

6.1 Traditional single instance estimators

6.1.1 Linear models

In the early 1800's Statisticians like Gauss and Legendre created methods of fitting equations to datasets using methods such as ordinary least squares[112].

Ordinary least squares (OLS) fits a linear model to data by minimizing the sum of squared residuals. Since then many other models have been created which extend this method to fit linear models to datasets. [111]

Ridge Regression[113], also known as Tikhonov regularization [114] or weight decay[115] depending on context, is a method of dealing with input matrices where OLS results in over or under-determined systems of equations perhaps from multicollinearity in small samples. An L_2 regularization term is added to the OLS system in the form of a weighted identity matrix (the 'ridge' from which it gets one of the names. The effect is a weighting of the input features (a form of feature selection) and a matrix which is invertible, allowing for solutions to OLS via orthogonal decomposition.

The LASSO (Least Absolute Shrinkage and Selection Operator), invented by Santosa [116] and popularized by Tibshirani [117] is a compromise between ridge regression's shrinking of coefficients and deletion of features via subset selection. It uses an L_1 regularization penalty. The 2 from L_2 denotes a squared distance metric while the 1 from L_1 denotes an absolute value of magnitude. L_1 penalizes coefficients to zero to help underfit and provide hard feature selection, while L_2 only scales the feature weights, severely punishes large values more than small values and is preferable if your data is not sparse.

Least Angle Regression (LARS) is a method for computing the LASSO (weights + feature selection) via coordinate descent making it computationally faster for some problems. [118]

Linear and Quadratic Discriminant Analysis (LDA, QDA) are method which fit multivariate gaussians to a dataset to find linear combinations which best explain the dataset. LDA uses a discriminant first defined by Fisher to compare the between class and within class variance.[119] This discriminant is used to position the component axes of the data to maximize this separation, usually using eigendecomposition of the covariance matrix. It is similar to Principal Component Analysis (PCA) which simply aligns the axes with maximum variance irrespective of class.

Rosenblatt's single layer Percepton used an analog to biological neurons to find separating hyperplanes in dataset [120]. It works for data which is linearly separable and is a precursor to modern neural networks.

6.1.2 Decision trees

Classification And Regression Trees (CART) and related algorithms by Scott Quinlan are decision tree algorithms[121]. They build a tree data structure from a dataset using information entropy measurements to separate data optimally using gini impurity (a measure of inequality related to probability of misclassification)[122] or information gain (a difference of entropy defined as $-\sum_i P_{ilog} P_i$) . The 'root' decision node of the tree separates the data in the broadest possible way into 2 groups. Additional nodes record further separation of those groups and so on until a fine degree of isolation is reached. The final ‘leaf’ node in the tree stores information about the target variable kept from the training data. When classifying a new data point it presents as a series of ‘questions’ about the data as the tree is traversed from the root node until arriving at the stored information about the target. Rotation trees are a type of decision tree where PCA is used internally as a dimension reduction technique [123]–[125]

6.1.3 Support vector machines

Support Vector Machines work by projecting a dataset into a higher-dimensional space where the data becomes linearly separable. Similar to previous algorithms the SVM places a deciding hyperplane in such a way to be as far as possible from the nearest training point of any class.[126]

6.1.4 Nearest neighbors

The nearest centroid model is a type of ‘nearest neighbour’ algorithm which simply finds the nearest training sample to a given test data and assigns it the same label as the training observation [127]. It is sensitive to the specific training dataset and to drift, but extremely effective when samples are similar, and class distribution is balanced.

6.1.5 Results from traditional single instance estimators

Figure 40 shows a summary of the cross-validation accuracy scores for each of the estimators described in this chapter. Each estimator was used with several feature selection methods as described in Chapter 5. For a guide on how to read the chart refer to Chapter 3.8. Accuracy was computed using Scikit-learn [59] and describes the fraction of samples where the estimator returned exactly the correct result out of the total number of samples. This assumes the cases are equally likely and this metric is not useful when this is not the case (when the classes are imbalanced).

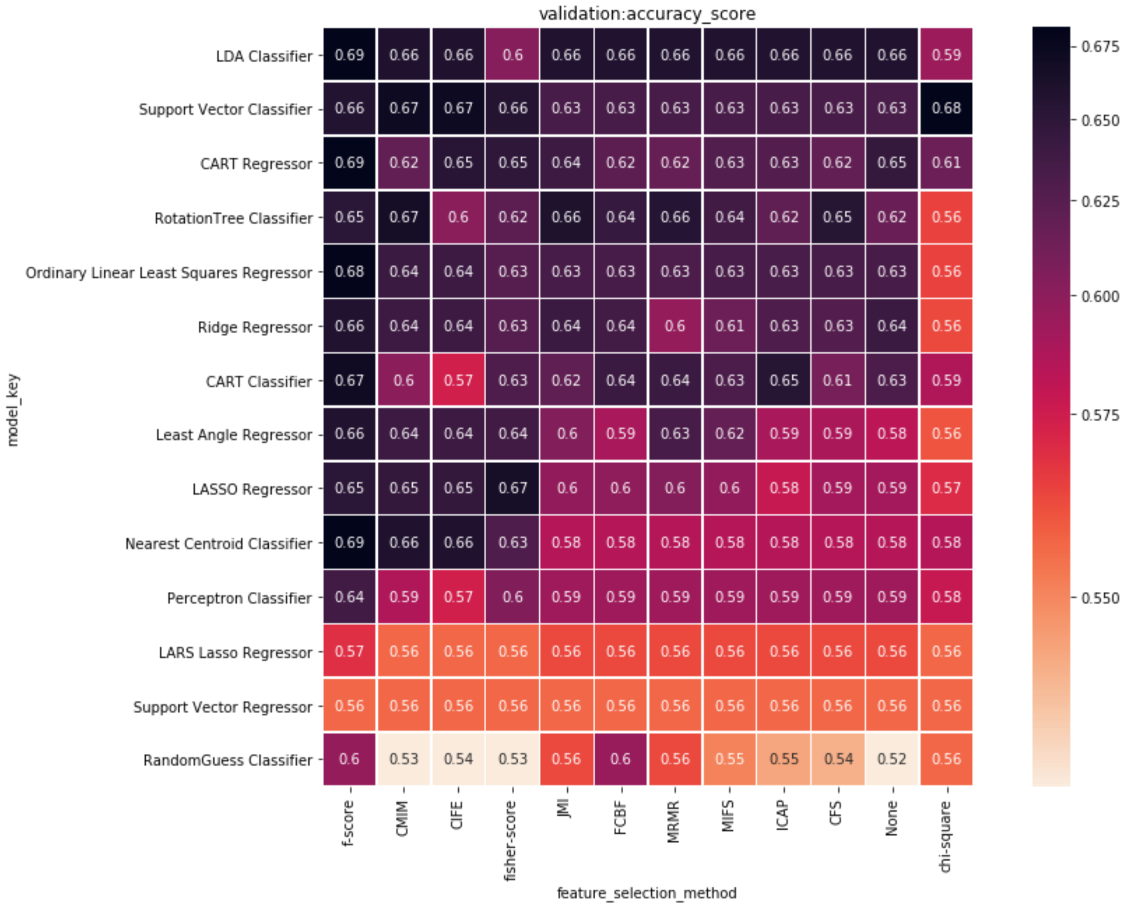


Figure 40 : Validation accuracy for single instance estimators

Figure 41 shows the mean of the area under the precision-recall curve (AUC-PR) for each estimator / feature selection method pair. Precision trades off with recall as decision thresholds are adjusted. The area under the PR curve gives a summary of that tradeoff and is useful in assessing performance in situations where true positives vs false positives are a concern.

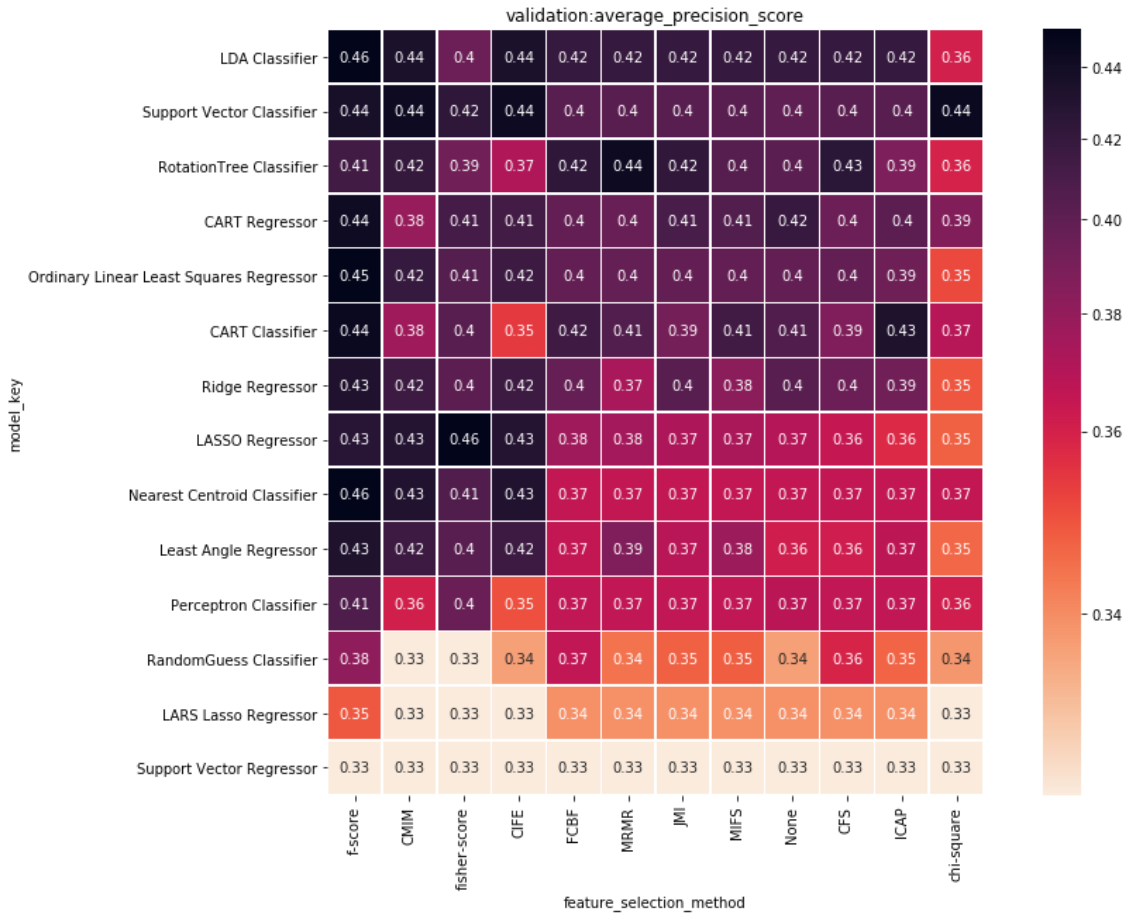


Figure 41 : Validation AUCPR for single instance estimators

6.2 Ensemble estimators

If one is to flip a slightly weighted coin and get 'heads' it's difficult to determine which way the coin is weighted. Additional tosses add further evidence to the true nature of the weighting until, after many independent observations one can become more confident in saying that the coin is weighted toward heads or tails. A collection of weak predictors can be used in a similar way. Even though one prediction on its own is not likely to be correct, by combining many weakly correlated observations of the same sample the quality of predictions can increase. There are many ways to do this combination such as boosting, stacking and voting. Gradient boosting machines involve building a model step by step using a differentiable loss function [128]–[131]. Usually the base model is a decision tree, the training process is some form of gradient descent and the method of combining or generating the estimators will vary slightly between implementations. There are many methods in the space and the following algorithms were used:

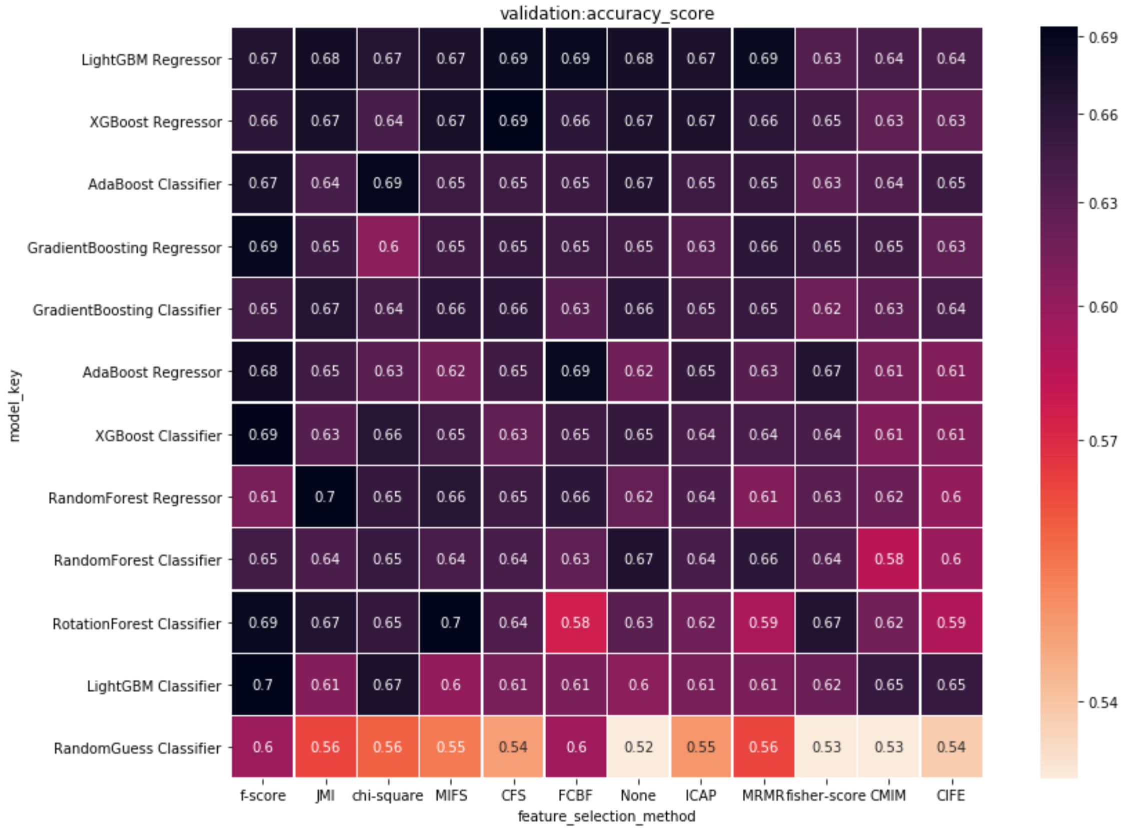
- Random Forests [132]
- RotationForests[123]
- Gradient Boosting [128], [129]
- Adaboost [133]
- XGBoost[134]
- LightGBM [135]

6.2.1 Results from ensemble estimators

Figure 42 gives cross-validation accuracy scores for the various ensemble estimators described in this chapter, Figure 43 gives the AUC-PR scores from cross-validation. A description of the cross-validation process and figures is given in Chapter 3.

Figure 44 shows the area under the curve for the receiver operating characteristic (ROC-AUC). The ROC shows the tradeoff of true positive rate vs false positive rate as decision thresholds are adjusted. The area varies between 0 and 1 with .5 signifying a random guess. As this metric is commonly used with binary classification the comparison is one-vs-rest. For regressor predictions the predicted class is determined by the closest class number to the output (ex: a prediction of 44 would be treated as the 45°class). This is useful if the algorithm is used for detection of a concerning condition in a monitoring application.

Figure 45 shows the Mean Absolute Error (MAE) from cross validation. In addition to accuracy and AUCPR which are usually given in binary classification tasks, the MAE describes how incorrect the angle prediction was (in degrees) vs the true torso angle of the subject. This is useful because the ‘angle’ classes are quite similar from a medical perspective and are actually part of a continuous spectrum.



model_key	f-score	JMI	chi-square	MIFS	CFS	FCBF	None	ICAP	MRMR	fisher-score	CMIM	CIFE
LightGBM Regressor	0.67	0.68	0.67	0.67	0.69	0.69	0.68	0.67	0.69	0.63	0.64	0.64
XGBoost Regressor	0.66	0.67	0.64	0.67	0.69	0.66	0.67	0.67	0.66	0.65	0.63	0.63
AdaBoost Classifier	0.67	0.64	0.69	0.65	0.65	0.65	0.67	0.65	0.65	0.63	0.64	0.65
GradientBoosting Regressor	0.69	0.65	0.6	0.65	0.65	0.65	0.65	0.63	0.66	0.65	0.65	0.63
GradientBoosting Classifier	0.65	0.67	0.64	0.66	0.66	0.63	0.66	0.65	0.65	0.62	0.63	0.64
AdaBoost Regressor	0.68	0.65	0.63	0.62	0.65	0.69	0.62	0.65	0.63	0.67	0.61	0.61
XGBoost Classifier	0.69	0.63	0.66	0.65	0.63	0.65	0.65	0.64	0.64	0.64	0.61	0.61
RandomForest Regressor	0.61	0.7	0.65	0.66	0.65	0.66	0.62	0.64	0.61	0.63	0.62	0.6
RandomForest Classifier	0.65	0.64	0.65	0.64	0.64	0.63	0.67	0.64	0.66	0.64	0.58	0.6
RotationForest Classifier	0.69	0.67	0.65	0.7	0.64	0.58	0.63	0.62	0.59	0.67	0.62	0.59
LightGBM Classifier	0.7	0.61	0.67	0.6	0.61	0.61	0.6	0.61	0.61	0.62	0.65	0.65
RandomGuess Classifier	0.6	0.56	0.56	0.55	0.54	0.6	0.52	0.55	0.56	0.53	0.53	0.54

Figure 42 : Validation accuracy for ensembles made of multiple estimators

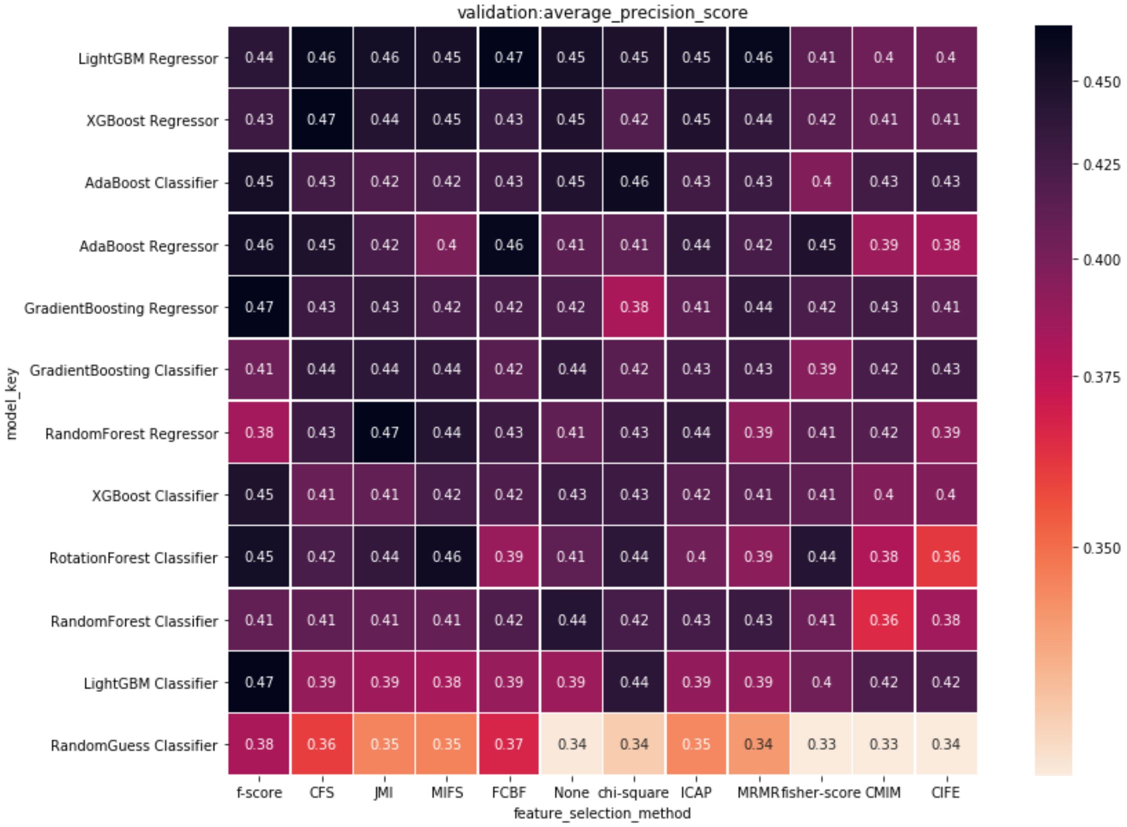


Figure 43 : Validation AUCPR for ensembles made of multiple estimators

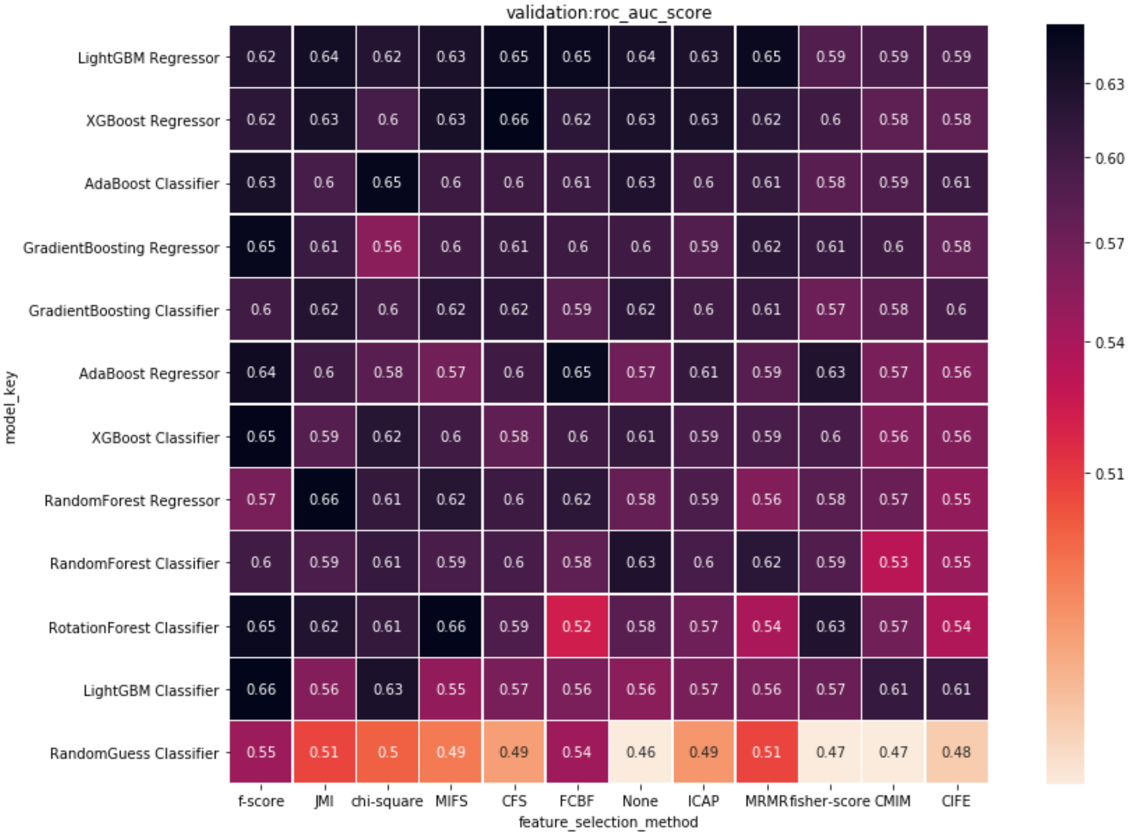


Figure 44 : ROC AUC for ensembles made of multiple estimators

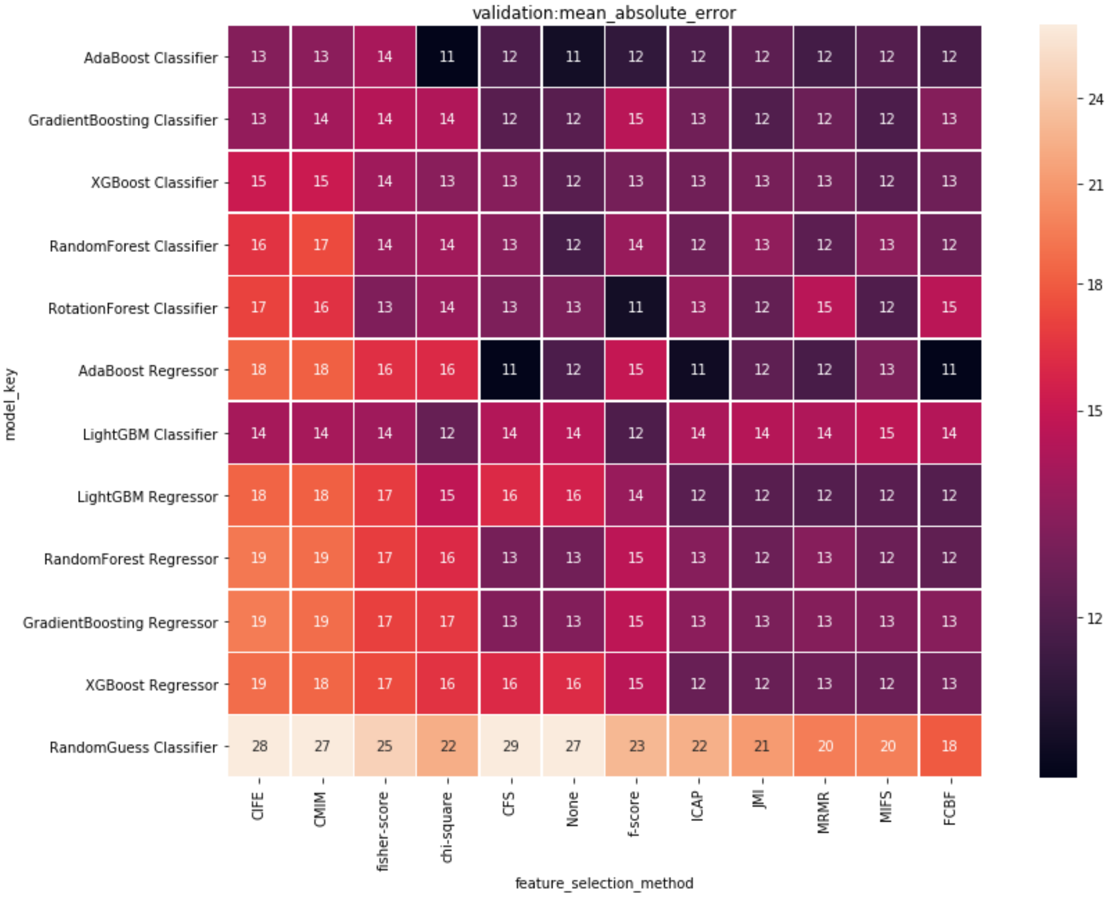


Figure 45 : Mean Absolute Error for ensembles made of multiple estimators

6.3 Neural networks

In a biological brain there are cells called ‘neurons’. When the sensitive ‘dendrites’ of a neuron are stimulated a chemical/electrical impulse travels down the length of the cell [120]. Once it reaches the end of the cell it may reach a gap called a ‘synapse’ which it can cross to stimulate the dendrites of another cell providing the conditions at the junction permit it to do so. This gated linking of many cells allows for the creation of complex neural pathways for the storage and processing of information.

Artificial neural networks simulate this process. They link artificial neurons which transmit signals according to a linear function of the sum of inputs to each node. If the aggregate input to a node is above a certain threshold input passes to the next node. Information usually passes from a set of input nodes to a set of output nodes with several 'hidden' nodes between. As input data ('training data') is passed to the system the weights controlling activation of certain pathways get adjusted in proportion to the strength of the response at the output. With sufficient training the network is able to respond to a given input such that output nodes respond in a useful way.

Common 'architectures' include classic 'feed forward networks' or 'perceptrons' in which nodes connect from input to output without any cycles or complex interaction. The nodes are usually grouped into 'layers' of connections. Single layer perceptrons can learn most linearly separable patterns and are identical to logistic regression models. Using multiple 'hidden layers' of nodes they can learn any Boolean function.

The deep belief network model by Hinton et al. 2006 [136] allowed networks to add increasing numbers of layers creating so-called 'deep neural networks' by solving the 'vanishing gradient problem' where information could not flow from input to output.

In the 2010's a flavor of network called a 'convolutional neural network' became popular for applications such as image processing [48]. These networks use a convolution operation common in digital signal processing to learn filters for capturing arbitrarily complex two-dimensional patterns.

Recurrent neural networks are used to process sequential data [137]. They share parameters between nodes, passing input from previous steps in a sequence to affect the prediction at the current step. Modern variants using ‘Long Short Term Memory’[138] and ‘Gated Recurrent Units’ [139] help control the focus of the network, weight new inputs effectively and are less susceptible to the vanishing gradient problem. LSTMs are more powerful but for small datasets GRUs are more effective [140].

6.3.1 One-dimensional convolutional neural network (CNN)

This network operates on the first 150 frames of a video clip (due to the repetitive nature of the clip even less might be reasonable, longer sequences increase the amount of data and compute required). For each frame in the sequence N features are computed. Which N are used is selected by the feature selection methods described above and ranges between 1 and 10. If used for multiclass classification the final softmax layer allows selection of the largest ‘probability’ from 3 target classes representing angles 0, 30 and 45. If used for regression all angles are used for training/testing and this layer is replaced with one using a linear activation.

Variations of this model are tried in an internal cross validation on the training data. The selected parameters are then used to build a model on the training data and pass the model off for testing on unseen data. This internal hyperparameter optimization changes the number of filters (2,4,8,16), kernel size (3,5,7,9) and learning rate scheduling (0.001 to 0.5 initial rate, halving every 10 epochs).

An example of a well performing model is given in Figure 46 which uses 4 features selected by F-Score (Circularity, Area in Pixels, Square Root Area, and a Hu moment), 50% dropout before max pooling, a kernel size of 7, 16 filters for 1 dimensional convolution with ReLU activation and an initial learning rate of 0.0015.

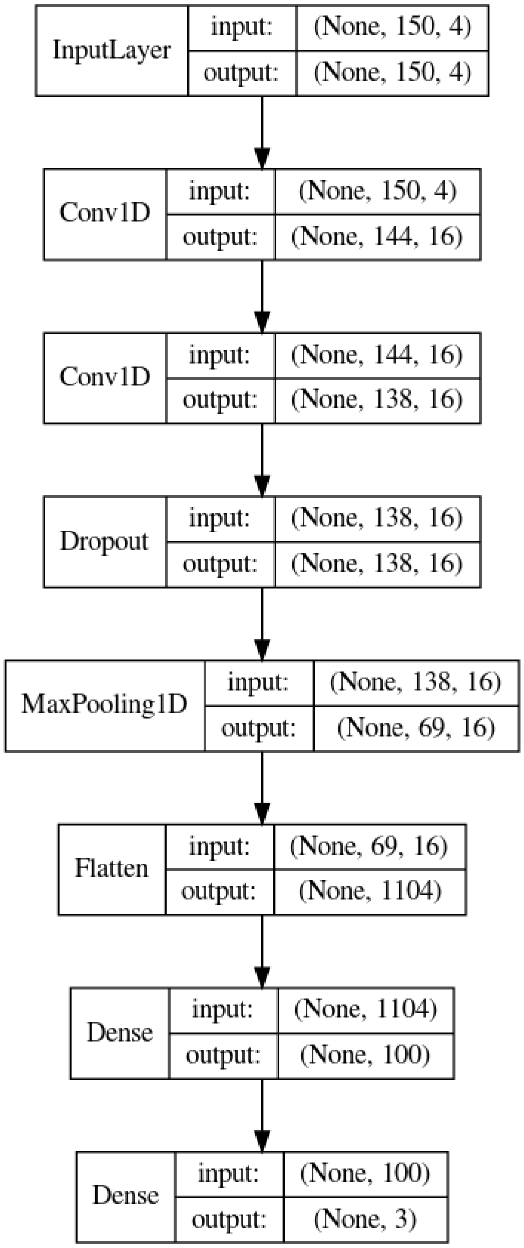


Figure 46 : Convolutional Neural Network Architecture

6.3.2 One-dimensional CNN + recurrent neural network (RNN)

A similar process was used to produce a model using a recurrent neural network. Consuming the same features selected by the same methods, the input is the same. To do model comparison the datasets were scaled with sklearn.RobustScaler (median removed and IQR used to scale) and split once with each training session completed in separate parallel pipelines using separate containerized environments for later reproducibility. The hyperparameters of filters and kernel size for 1D convolution were tuned as before. The difference is that instead of additional convolution layers this architecture instead used a Bidirectional RNN[53] with 32 Gated Recurrent Units[139] and no spatial dropout.

An example is shown with the same 4 f-score selected features as were shown in the example convolutional network, a kernel size of 9, 4 filters for the 1D Convolutional layer with ReLU activation, MaxPooling layer with stride 2 and pool size 2, 32 pairs of GRUs (64 elements total), and an initial learning rate of 0.0010.

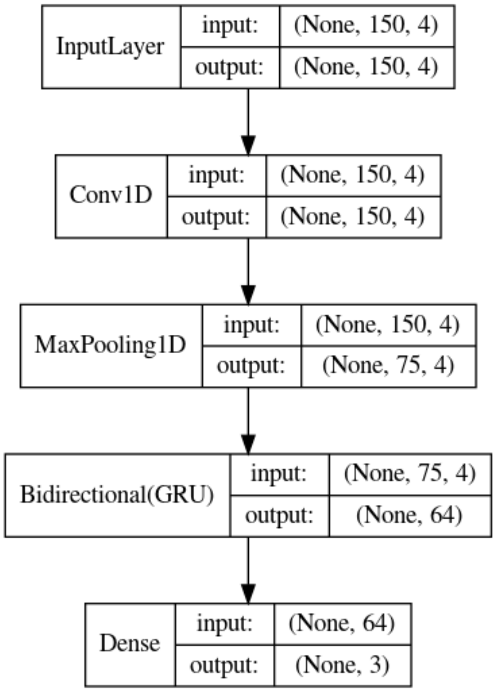


Figure 47 : Recurrent Neural Network Architecture

6.3.3 Results from neural networks

A key consideration in using neural networks is that they were long considered infeasible due to computer hardware constraints. Figure 48 and Figure 49 respectively show the cross-validation inference and training times for the different neural networks. The networks were trained on a CPU using hardware from Amazon Web Services. The training computer had 4 CPUs and 16 GB of memory.

Figure 50 shows the training accuracy to determine whether the networks were overfitting while Figure 51 shows the performance on unseen data. Similarly Figure 52 gives MAE for the training data while Figure 53 gives the same metric for unseen data using 3 fold cross-validation.

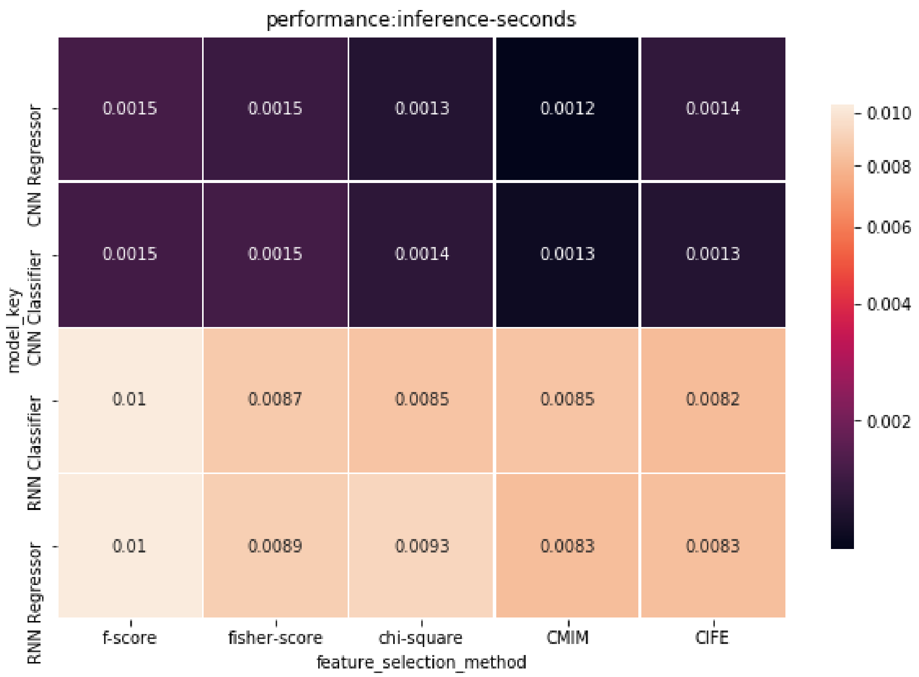


Figure 48 : Inference time in seconds for neural network estimators

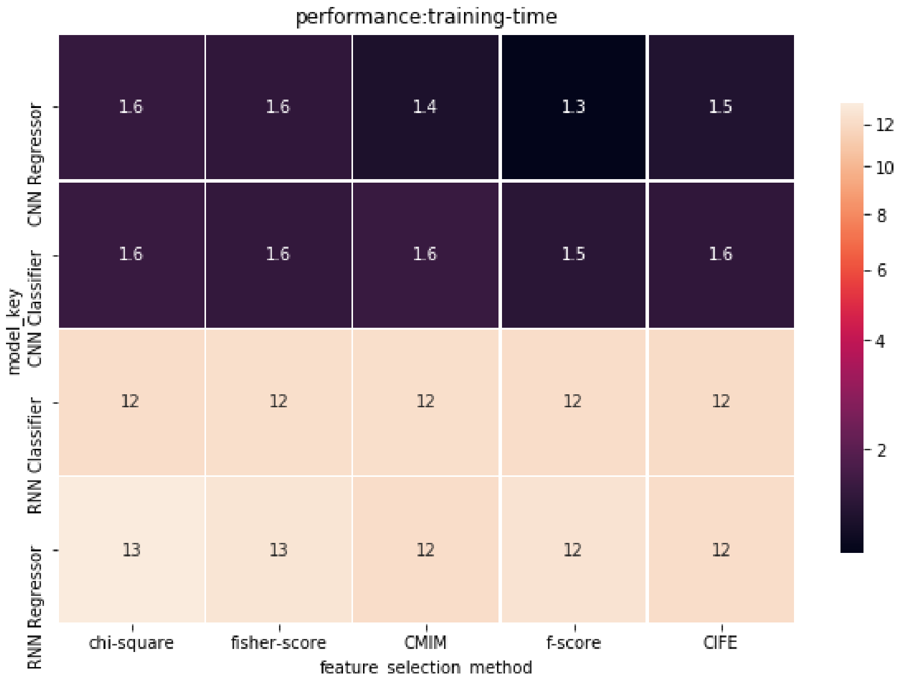


Figure 49 : Training time in seconds for neural network estimators

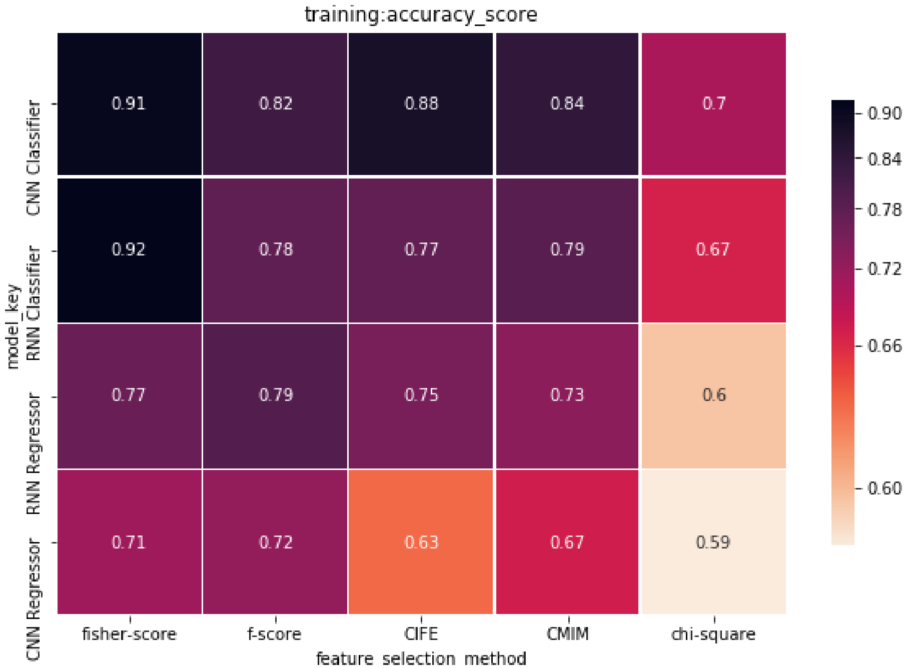


Figure 50 : Training accuracy for neural network estimators

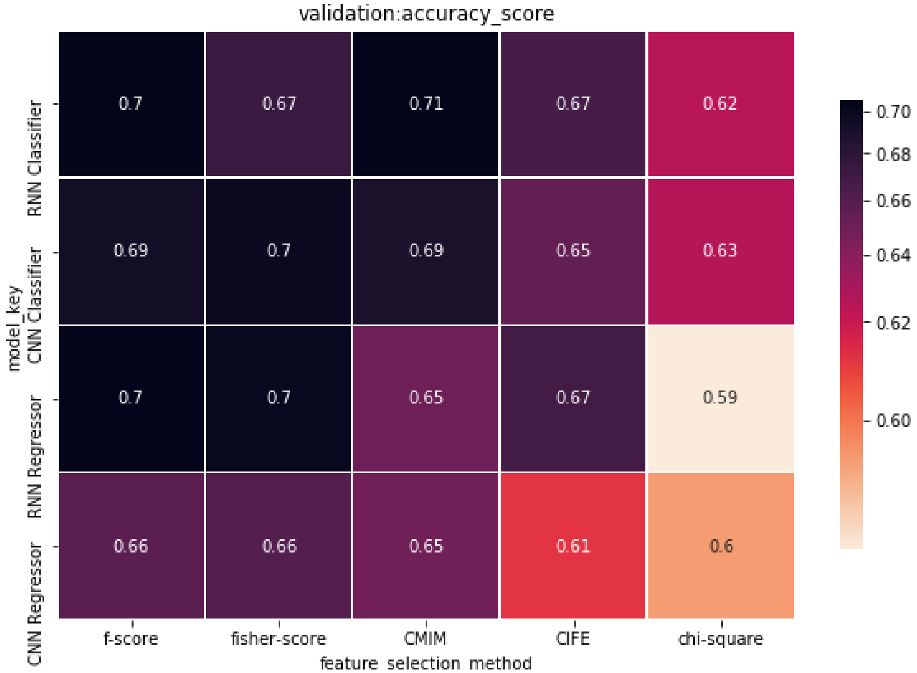


Figure 51 : Validation accuracy for neural network estimators

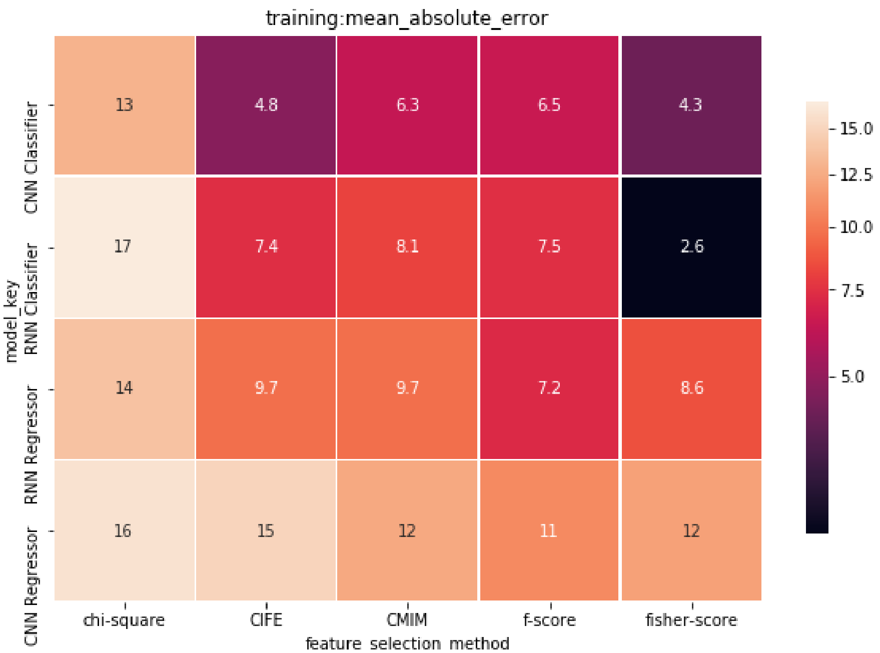


Figure 52 : Training MAE for neural network estimators

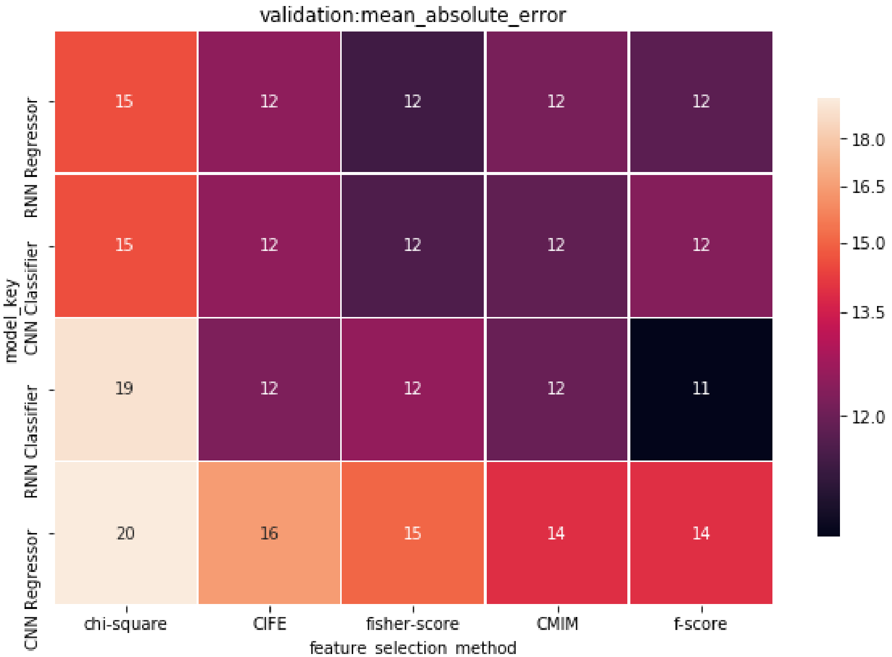


Figure 53 : Validation MAE for neural network estimators

Chapter 7 Results

7.1 Treatment as regression

Figure 54 shows the average distance between the angle which was predicted and the actual angle of the subject as a mean absolute error (MAE). Each cell of the grids represents the average of the target metric across 3 folds of cross validation on all cross-sectional data. The y axis is sorted by the mean across a row (the mean for a given regressor) and the x axis is sorted by mean across a column (the mean for a given feature selection method). This sorting is continued in later figures. Training scores are provided for reference (bias vs variance tradeoff). Validation scores represent the expected performance on unseen data.

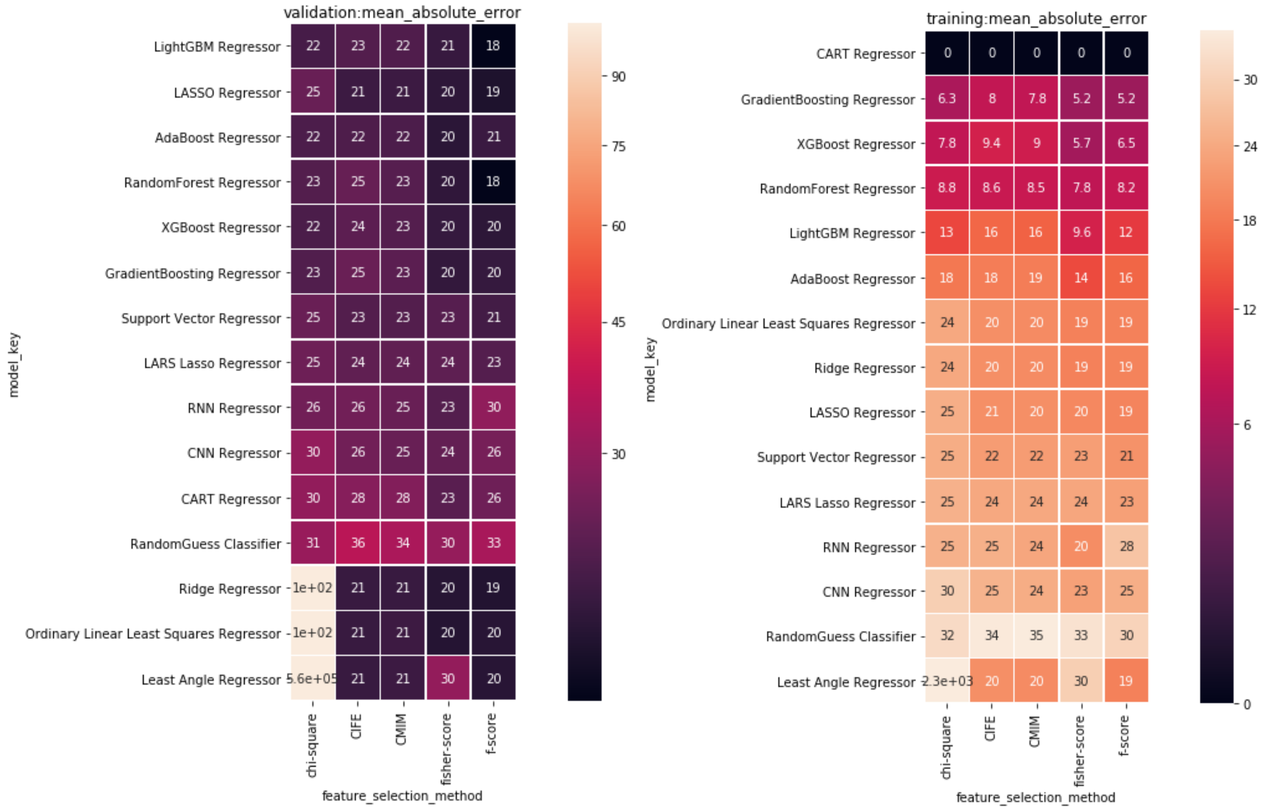


Figure 54 : Mean Absolute Error when using all angles

7.2 Treatment as classification

There are multiple use cases for the predictions being presented. The previous regression estimates angle directly as a proxy for measuring the continuous value of JVP as a height, perhaps as an input to another system. However, the objective may be to classify a subject to quickly make an automated decision. For this classification task the angles of 0, 30 and 45 degrees were chosen as they are common in literature. To produce classification metrics the results were created using a one-vs-rest binarization [141]. For algorithms which did not produce an enumerated result for one of the three categories the closest center point for each bin was used. For example: if 31 was returned that is close to 30 so the class 30 was assigned, 44 would likewise assign to 45.

Figure 55 shows the area under the Receiver Operating Characteristic curve (ROC AUC).This curve presents the tradeoff between True Positive Rate (TPR) vs False Positive Rate (FPR). ROC AUC varies between 0 and 1 with .5 representing random guessing. Figure 56 shows the average precision score which represents the area under the Precision Recall curve (AUC PR). AUCPR has the same 0 to 1 range, represents the tradeoff between selecting all values of interest vs not showing irrelevant values. Figure 57 shows accuracy, the percentage of correctly classified samples. Figure 58 shows inference time measured in seconds. Figure 59 shows time to train an instance. Both performance measures were taken from Python[142] implementations of the algorithms in a ‘containerized’ environment provided by Docker [143] running on a host with 4 CPUs and 16 GB of memory, optimized for parallel execution where possible.

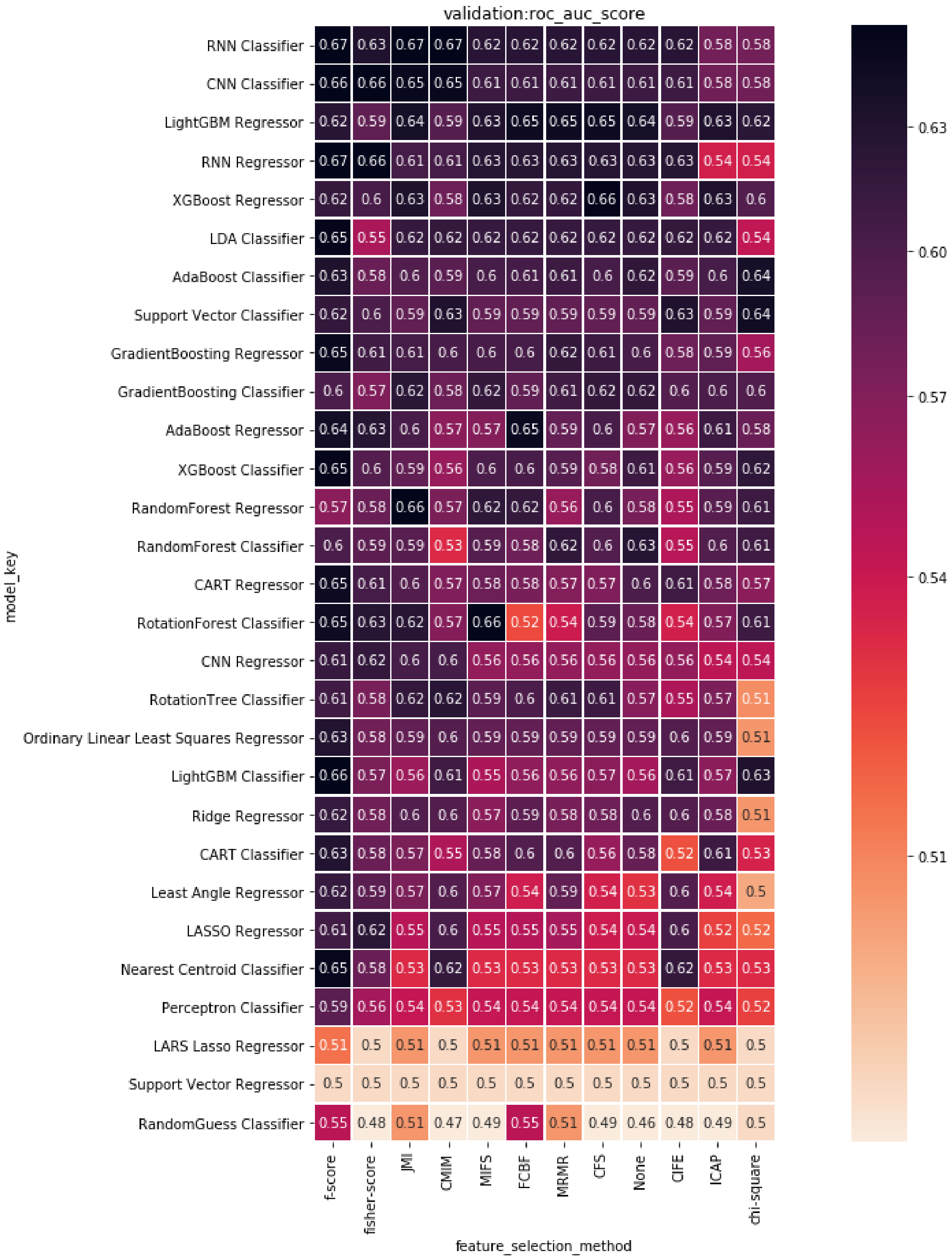


Figure 55 : ROC AUC for classification

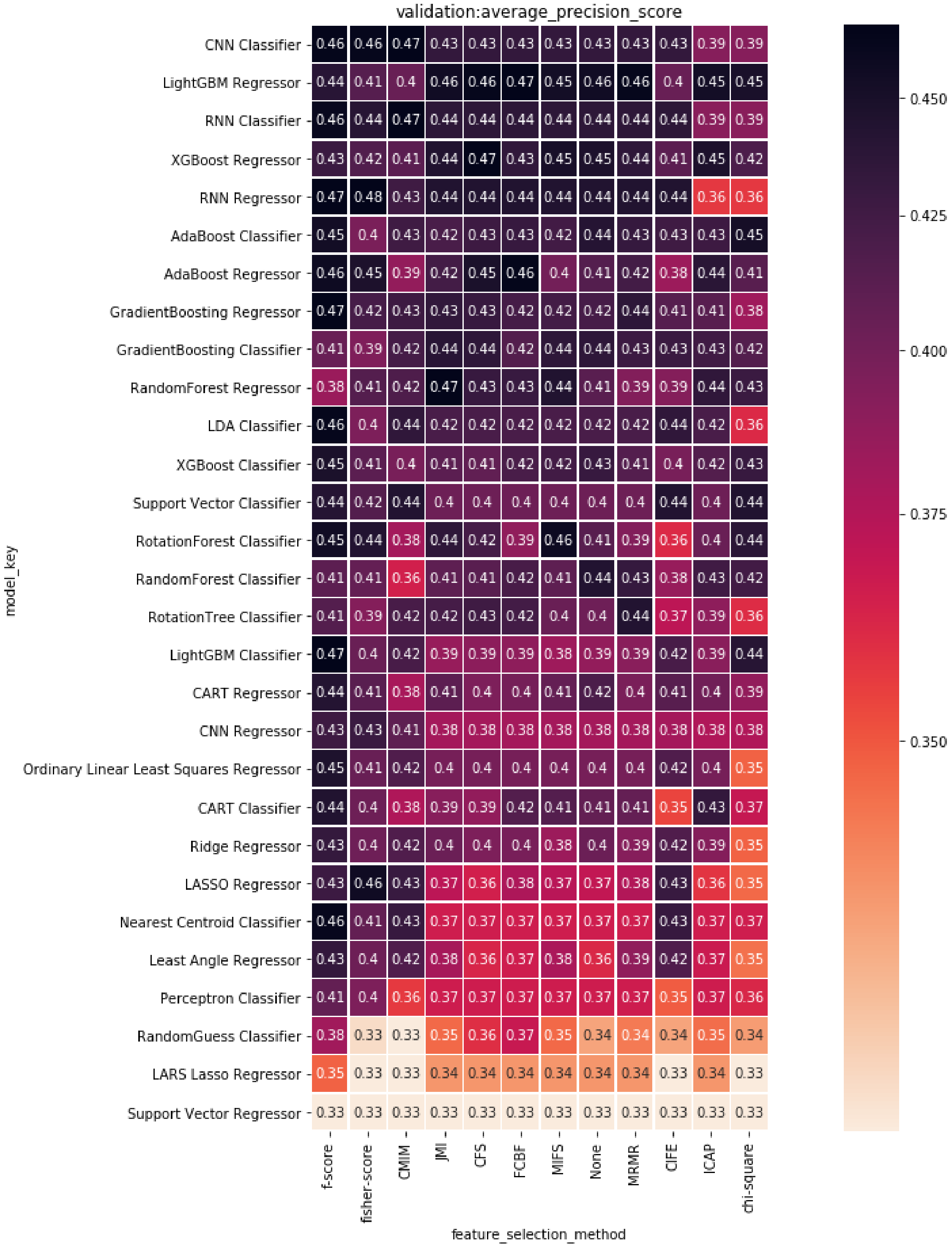


Figure 56 : Average Precision Score (AUC PR) for classification

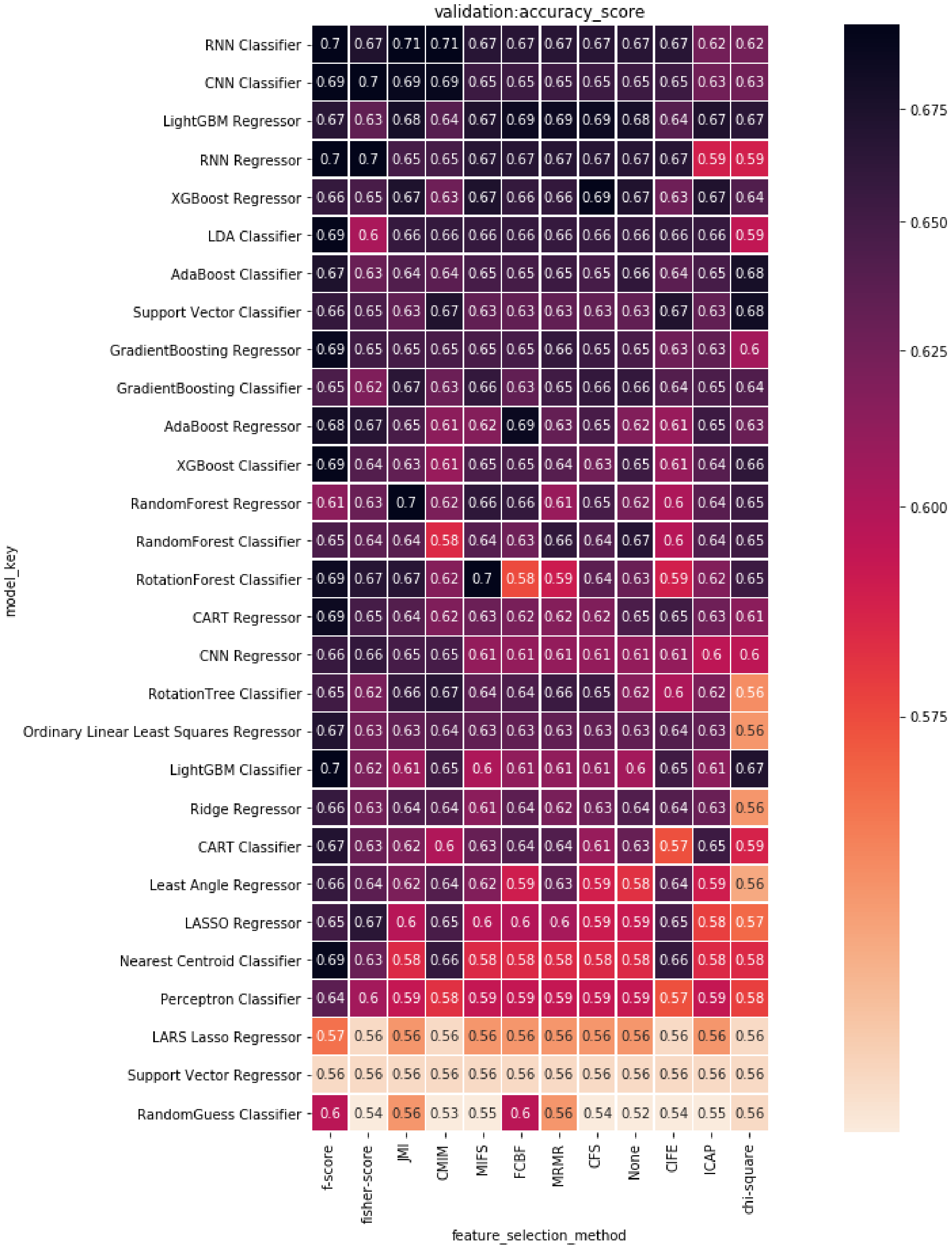


Figure 57 : Accuracy score for classification.

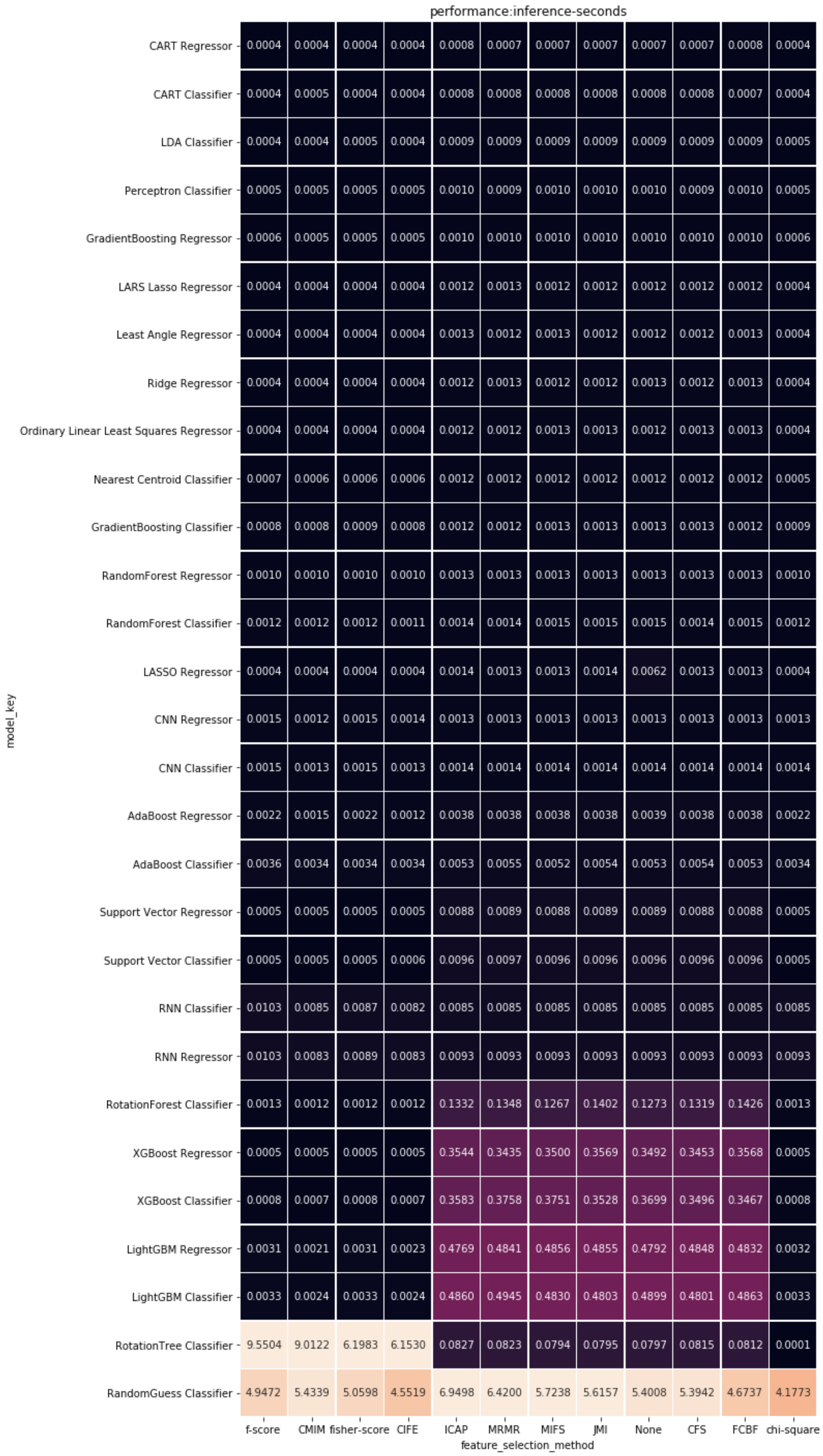
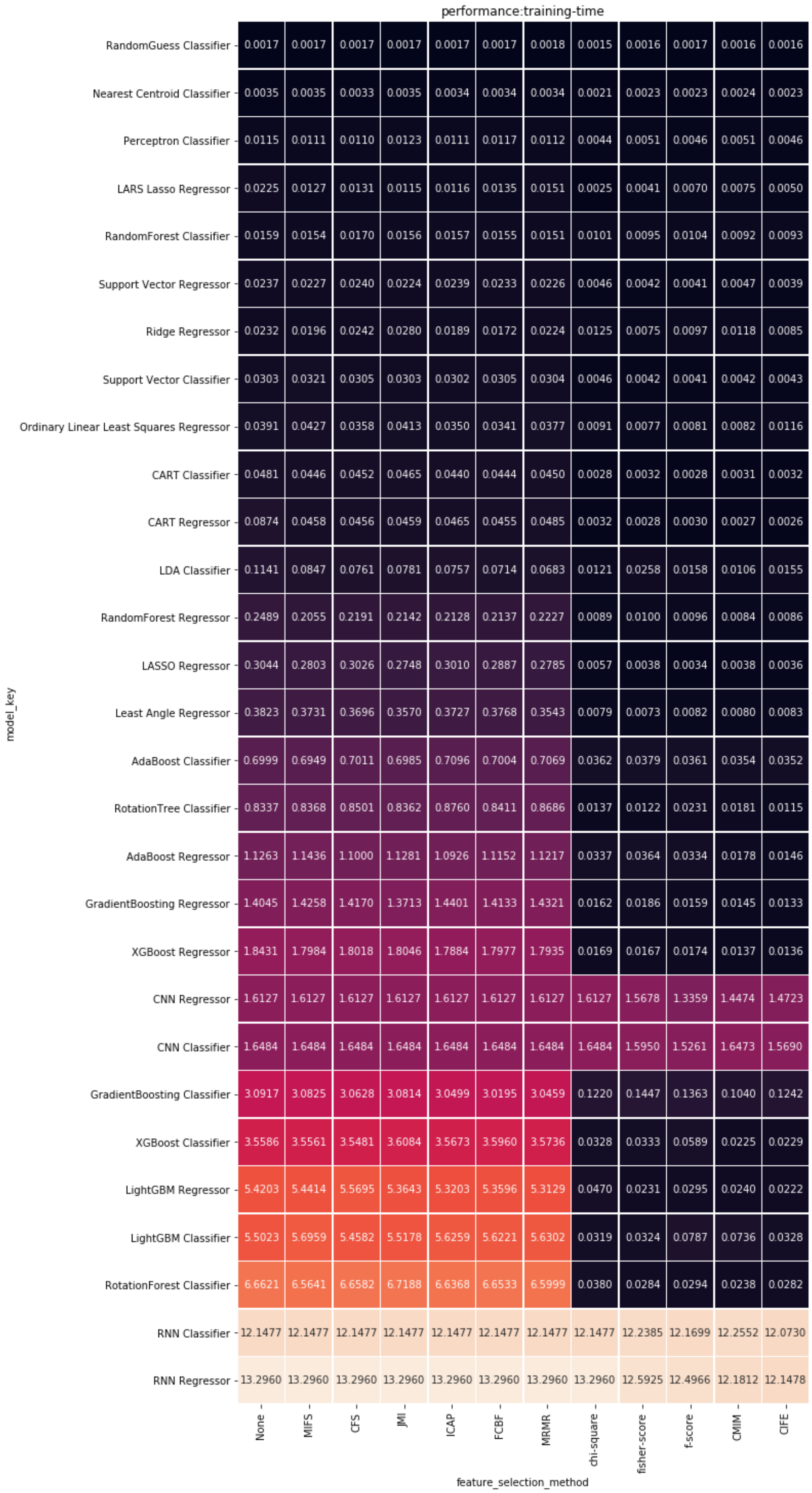


Figure 58 : Performance time - Inference in seconds



model_key	None	MIFS	CFS	JMI	ICAP	FCBF	MRMR	chi-square	fisher-score	f-score	CMIM	CIFE
RandomGuess Classifier	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0018	0.0015	0.0016	0.0017	0.0016	0.0016
Nearest Centroid Classifier	0.0035	0.0035	0.0033	0.0035	0.0034	0.0034	0.0034	0.0021	0.0023	0.0023	0.0024	0.0023
Perceptron Classifier	0.0115	0.0111	0.0110	0.0123	0.0111	0.0117	0.0112	0.0044	0.0051	0.0046	0.0051	0.0046
LARS Lasso Regressor	0.0225	0.0127	0.0131	0.0115	0.0116	0.0135	0.0151	0.0025	0.0041	0.0070	0.0075	0.0050
RandomForest Classifier	0.0159	0.0154	0.0170	0.0156	0.0157	0.0155	0.0151	0.0101	0.0095	0.0104	0.0092	0.0093
Support Vector Regressor	0.0237	0.0227	0.0240	0.0224	0.0239	0.0233	0.0226	0.0046	0.0042	0.0041	0.0047	0.0039
Ridge Regressor	0.0232	0.0196	0.0242	0.0280	0.0189	0.0172	0.0224	0.0125	0.0075	0.0097	0.0118	0.0085
Support Vector Classifier	0.0303	0.0321	0.0305	0.0303	0.0302	0.0305	0.0304	0.0046	0.0042	0.0041	0.0042	0.0043
Ordinary Linear Least Squares Regressor	0.0391	0.0427	0.0358	0.0413	0.0350	0.0341	0.0377	0.0091	0.0077	0.0081	0.0082	0.0116
CART Classifier	0.0481	0.0446	0.0452	0.0465	0.0440	0.0444	0.0450	0.0028	0.0032	0.0028	0.0031	0.0032
CART Regressor	0.0874	0.0458	0.0456	0.0459	0.0465	0.0455	0.0485	0.0032	0.0028	0.0030	0.0027	0.0026
LDA Classifier	0.1141	0.0847	0.0761	0.0781	0.0757	0.0714	0.0683	0.0121	0.0258	0.0158	0.0106	0.0155
RandomForest Regressor	0.2489	0.2055	0.2191	0.2142	0.2128	0.2137	0.2227	0.0089	0.0100	0.0096	0.0084	0.0086
LASSO Regressor	0.3044	0.2803	0.3026	0.2748	0.3010	0.2887	0.2785	0.0057	0.0038	0.0034	0.0038	0.0036
Least Angle Regressor	0.3823	0.3731	0.3696	0.3570	0.3727	0.3768	0.3543	0.0079	0.0073	0.0082	0.0080	0.0083
AdaBoost Classifier	0.6999	0.6949	0.7011	0.6985	0.7096	0.7004	0.7069	0.0362	0.0379	0.0361	0.0354	0.0352
RotationTree Classifier	0.8337	0.8368	0.8501	0.8362	0.8760	0.8411	0.8686	0.0137	0.0122	0.0231	0.0181	0.0115
AdaBoost Regressor	1.1263	1.1436	1.1000	1.1281	1.0926	1.1152	1.1217	0.0337	0.0364	0.0334	0.0178	0.0146
GradientBoosting Regressor	1.4045	1.4258	1.4170	1.3713	1.4401	1.4133	1.4321	0.0162	0.0186	0.0159	0.0145	0.0133
XGBoost Regressor	1.8431	1.7984	1.8018	1.8046	1.7884	1.7977	1.7935	0.0169	0.0167	0.0174	0.0137	0.0136
CNN Regressor	1.6127	1.6127	1.6127	1.6127	1.6127	1.6127	1.6127	1.6127	1.5678	1.3359	1.4474	1.4723
CNN Classifier	1.6484	1.6484	1.6484	1.6484	1.6484	1.6484	1.6484	1.6484	1.5950	1.5261	1.6473	1.5690
GradientBoosting Classifier	3.0917	3.0825	3.0628	3.0814	3.0499	3.0195	3.0459	0.1220	0.1447	0.1363	0.1040	0.1242
XGBoost Classifier	3.5586	3.5561	3.5481	3.6084	3.5673	3.5960	3.5736	0.0328	0.0333	0.0589	0.0225	0.0229
LightGBM Regressor	5.4203	5.4414	5.5695	5.3643	5.3203	5.3596	5.3129	0.0470	0.0231	0.0295	0.0240	0.0222
LightGBM Classifier	5.5023	5.6959	5.4582	5.5178	5.6259	5.6221	5.6302	0.0319	0.0324	0.0787	0.0736	0.0328
RotationForest Classifier	6.6621	6.5641	6.6582	6.7188	6.6368	6.6533	6.5999	0.0380	0.0284	0.0294	0.0238	0.0282
RNN Classifier	12.1477	12.1477	12.1477	12.1477	12.1477	12.1477	12.1477	12.1477	12.2385	12.1699	12.2552	12.0730
RNN Regressor	13.2960	13.2960	13.2960	13.2960	13.2960	13.2960	13.2960	13.2960	12.5925	12.4966	12.1812	12.1478

Figure 59 : Performance - time in seconds to train instance of model

7.3 Treatment as ‘high vs. low’ classification

If angles 0 and 15 are treated as condition ‘low’ and angles above these are treated as ‘high’ the problem becomes binary classification and it is possible to produce receiver operating characteristic curves as seen in Figure 60. The accuracies for each fold at the threshold of 15 are 78%, 86% and 89% respectively.

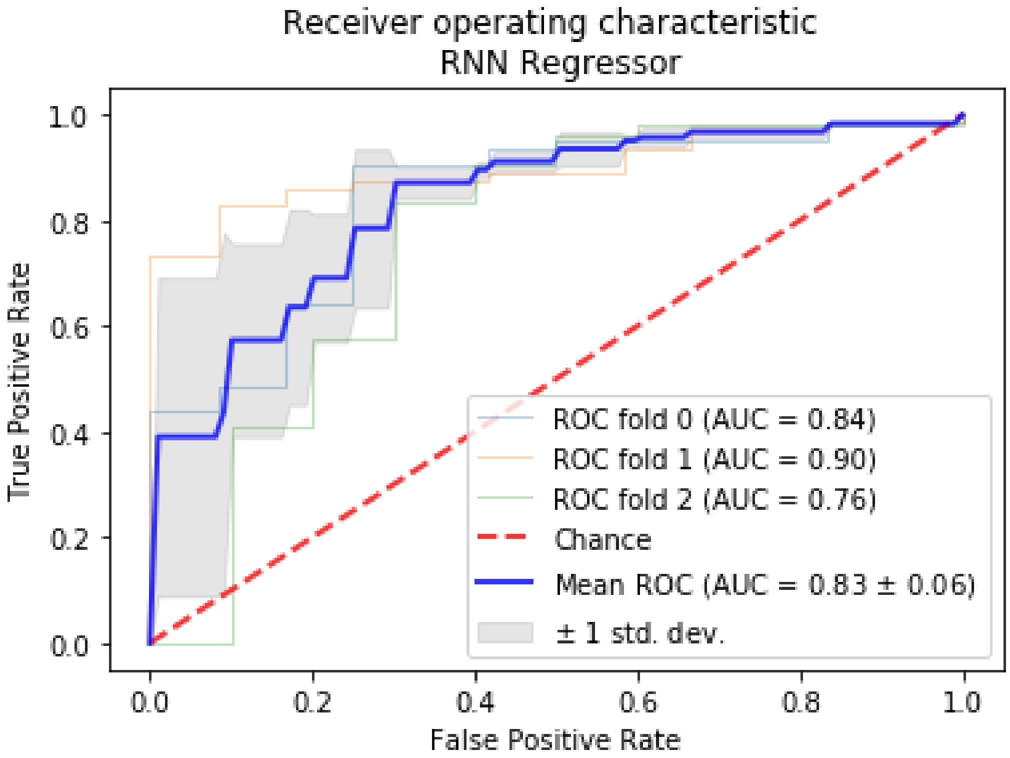


Figure 60 : RNN regressor used at binary classifier.

Chapter 8 Discussion

8.1 Overview

Hilbert 2016 found good correlation of IJV ultrasound with central venous pressure [10]. A study of medical residents estimating JVP and comparing their success to estimating right atrial pressure with ultrasound showed JVP can be quite hard to measure for untrained persons but still gave 60% overall accuracy for JVP and 71% with ultrasound [11]. As seen in Figure 57 our automated process give similar accuracies fairly consistently and exceeded their score in some folds.

When treated as a regression problem a dumb regressor which always returns the mean angle of 45 degrees will have a maximum error of 45 degrees. The random guess classifier gave MAE of 30 to 35. LightGBM Regressor gave a mean error of 18 degrees when assessing all angles with the outliers included and in the limited angle case an MAW of 11. A visual representation of an 11 degree angle is shown in Figure 61. The wooden bed used during some recordings allowed for up to 15 degrees of error and small changes in ultrasound probe position may have a big impact on data quality. Taking that into consideration the regression predicts reasonably well. As shown in referenced publications, if outliers are removed or the problem is simplified to ‘high/low’ type classification as is commonly seen for JVP estimation accuracy of these techniques approaches 100%

11° angle
18° angle

Figure 61 : Visualization of angles

8.2 Interpretability

For something which may be deployed in a medical setting it is important that it is possible to understand the predictions being made by a classifier and know that the algorithms are predicting as they do for reasons which make some intuitive sense.

Referring to the neural network architecture defined in Figure 46, data flows from the input layer, then through 2 convolutional layers, is processed with dropout, max pooled, flattened and passed to a fully connected dense layer before the actual classification is provided by a final softmax operation in the last layer.

Figure 62 shows the activation of each of these layers as 150 frames of input is passed through the network (with 5 features per frame). As can be seen in the plots of cross-sectional area vs time (Figure 30) and frequency analysis (Figure 22), pulse is a noticeable part of some classes of samples. The activations actually appear to augment the pulse and summarize it in a simple way compared to manually processing feature aggregates and running feature selection.

In a similar way Figure 47 gives RNN architecture and Figure 63 gives activation. The RNN has an internal concept of a sequence and so is a bit more compact.

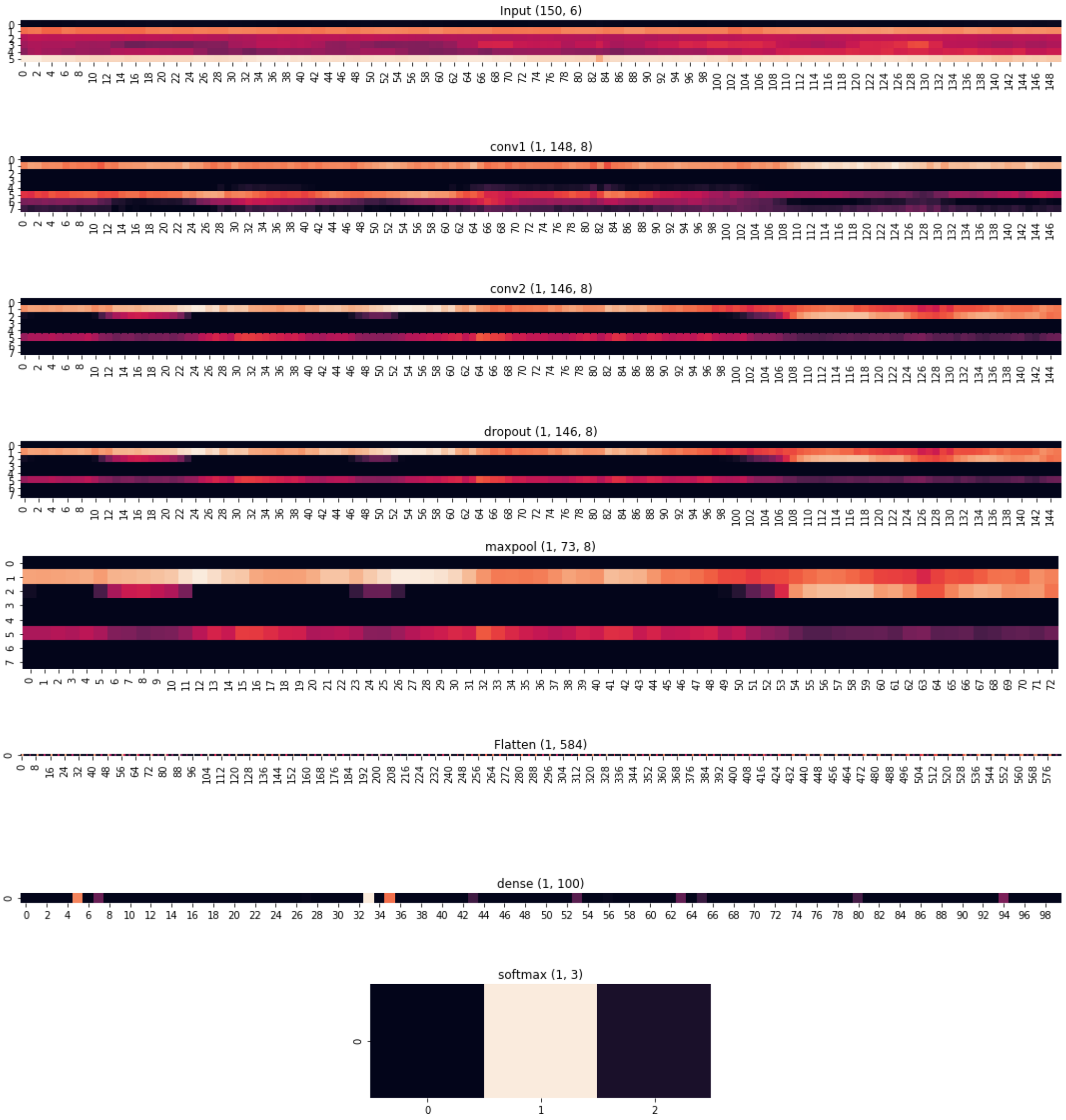


Figure 62: Activation of CNN layers

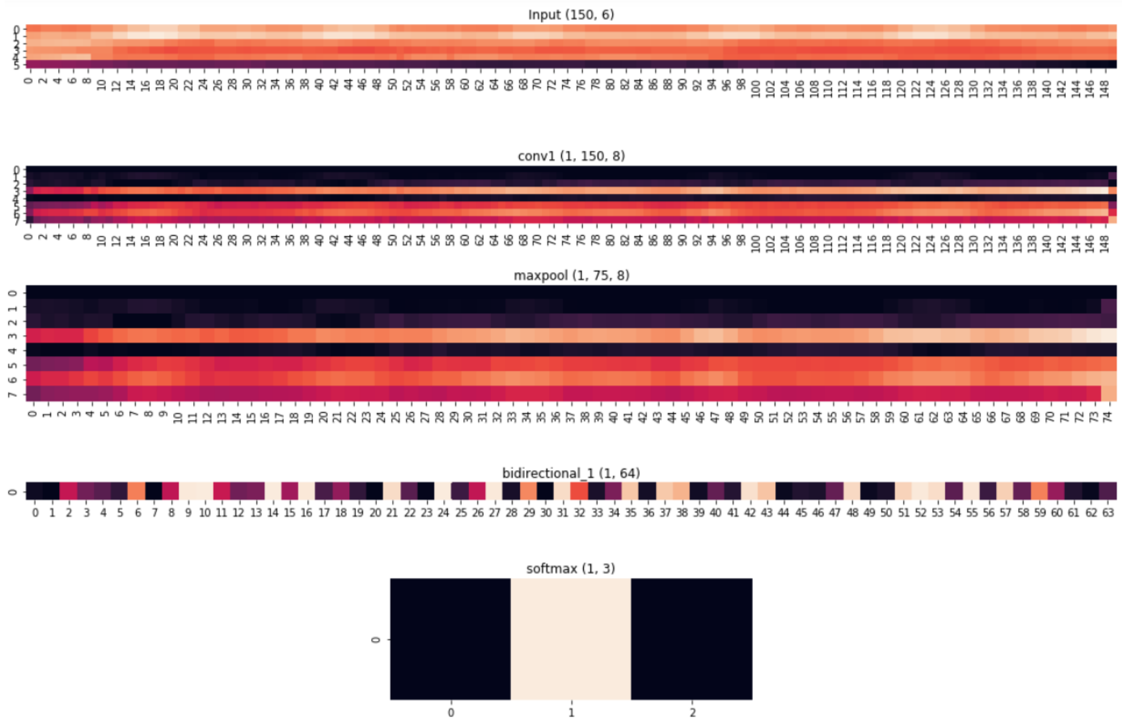


Figure 63 : Activation of RNN layers

Figure 64 shows the output of the SHAP library applied to the predictions of a gradient boosting classifier [80]–[82]. For a single sample it explains which features contributed to the score. For example: the compactness of the contour pushed the prediction higher (at higher angles the IJV collapses), while the energy across the gradient lowered the score.

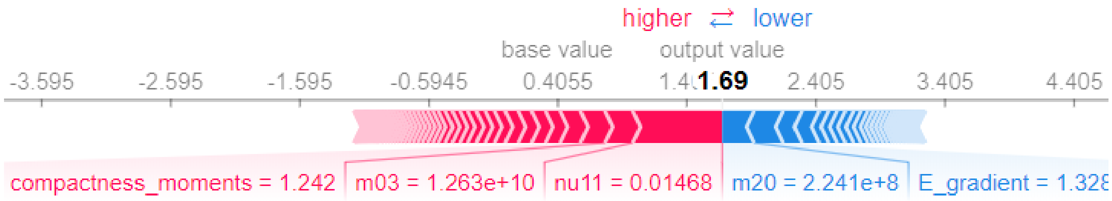


Figure 64 : Single inference explanation using SHAP on a Gradient Boosted Tree estimator

8.3 Computational performance

With the advent of modern cloud computing and point of care devices computational performance and deployment is not a concern. All estimators train in seconds with a worst case of under 13 seconds without using specialized processors. Neural network segmentation can be achieved in real time using graphic or neural processor acceleration and inference could be deployed live.

Chapter 9 Conclusions and further research

The dataset collected was quite small given anatomical variation between people. A learning curve for a single subject with longitudinal data collection is shown in Figure 65. As can be seen relatively few samples from a single variation in anatomy is enough for a classifier to learn, but error may remain high.

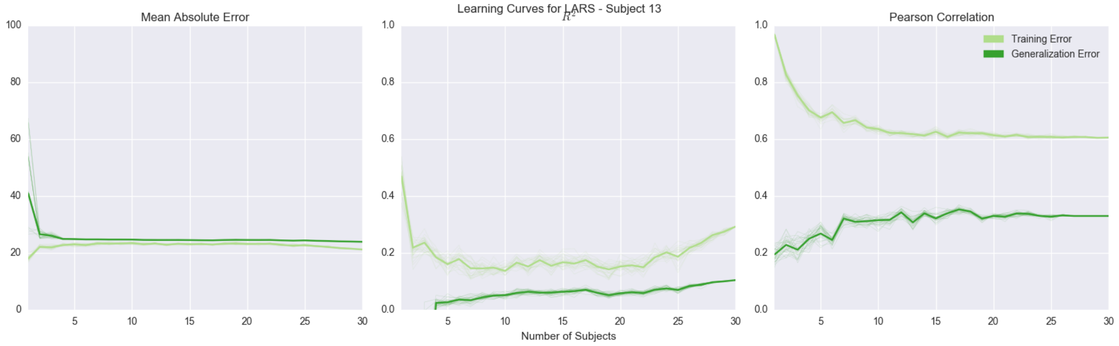


Figure 65 : Learning curve - performance vs samples

Future steps would include a large increase in the number of subjects beyond the 34 collected. The population studied should include subjects with varied habitus such that a range of thin and thick necks are included. The patient positioning should be varied to compensate for the practicalities of using the method with various subject posture, probe positioning and the impact of rotation of the head on the IJV as described in [144]–[146] .

The limit to which the model can predict angle in simulation is bounded by the size of the effect. If there is a large amount of error intrinsic to the physical process, the estimator will not be able to overcome this gap. While assistance is most useful to inexperienced clinicians, inexperienced clinicians would also be persons most likely to malposition the patient or the ultrasound probe in such a way that the estimator would be unable to compensate. One way to make the system more robust would be to collect data which finely tracks these variations in subject or probe positioning and incorporate these as additional features in the model. Also, error could be reduced by modifying the collection process itself to make it more robust to the physical positioning of the probe, perhaps with a type of probe which scans the entirety of the neck in 3D or which helps consistently place the probe through mechanical means.

A further study should be undertaken which compares an assisted approach with unassisted approaches. A clinician measuring IJV diameters directly with the ultrasound unit could be compared to automated methods. The gold standard for comparison would be a third measurement of volume status, the central venous pressure (CVP) resulting from central line measurement against the assisted and unassisted approaches.

As shown in the training metric sets the neural network and GBM estimators do not completely fit to the training data. It may be possible that they have reached a limit due to insufficient data. In most situations greater accuracies are possible with additional data collection.

Given that UNet can reliably segment images as a preprocessing step and is built in the same framework as the later estimators an approach which reads video frames directly to produce predictions and explanations might be a good direction for future research.

References

[1] C. R. Gracia and K. T. Barnhart, “Diagnosing ectopic pregnancy: decision analysis comparing six strategies,” *Obstet. Gynecol.*, vol. 97, no. 3, pp. 464–470, Mar. 2001.

[2] A. H. Miller, B. A. Roth, T. J. Mills, J. R. Woody, C. E. Longmoor, and B. Foster, “Ultrasound Guidance versus the Landmark Technique for the Placement of Central Venous Catheters in the Emergency Department,” *Acad. Emerg. Med.*, vol. 9, no. 8, pp. 800–805, Aug. 2002.

[3] G. D. Ludwig and F. W. Struthers, *Considerations underlying the use of ultrasound to detect gallstones and foreign bodies in tissue*. Naval Medical Research Institute, 1949.

[4] R. R. Steuer, D. C. H. Harris, and J. M. Conis, “A new optical technique for monitoring hematocrit and circulating blood volume: its application in renal dialysis.” 1993.

[5] R. H. Rosenwasser, E. Garrido, M. H. Freed, and R. C. Shupak, “Monitoring of Activated Clotting Time during Carotid Endarterectomy,” *Neurosurgery*, vol. 9, no. 5, pp. 521–523, Nov. 1981.

[6] H. Kasuya, H. Onda, T. Yoneyama, T. Sasaki, and T. Hori, “Bedside Monitoring of Circulating Blood Volume After Subarachnoid Hemorrhage,” *Stroke*, vol. 34, no. 4, pp. 956–960, Apr. 2003.

[7] M. Yashiro, Y. Hamada, H. Matsushima, and E. Muso, “Estimation of Filtration

Coefficients and Circulating Plasma Volume by Continuously Monitoring Hematocrit during Hemodialysis," *Blood Purif.*, vol. 20, no. 6, pp. 569–576, 2002.

[8] F. Bremer, A. Schiele, J. Sagkob, T. Palmaers, and K. Tschaikowsky, "Perioperative monitoring of circulating and central blood volume in cardiac surgery by pulse dye densitometry," *Intensive Care Med.*, vol. 30, no. 11, pp. 2053–2059, Nov. 2004.

[9] K. Kalantari, J. N. Chang, C. Ronco, and M. H. Rosner, "Assessment of intravascular volume status and volume responsiveness in critically ill patients," *Kidney Int.*, vol. 83, no. 6, pp. 1017–1028, Jun. 2013.

[10] T. Hilbert, R. K. Ellerkmann, S. Klaschik, C. Putensen, and M. Thudium, "The Use of Internal Jugular Vein Ultrasonography to Anticipate Low or High Central Venous Pressure During Mechanical Ventilation," *J. Emerg. Med.*, vol. 50, no. 4, pp. 581–587, Apr. 2016.

[11] J. M. Brennan *et al.*, "A comparison by medicine residents of physical examination versus hand-carried ultrasound for estimation of right atrial pressure.," *Am. J. Cardiol.*, vol. 99, no. 11, pp. 1614–6, Jun. 2007.

[12] E. Karami, M. Shehata, P. Mcguire, and A. Smith, "Ultrasound Image Segmentation Techniques for Tracking and Measurement of the Internal Jugular Vein," *energy*, vol. 2, no. 1, p. 6, 2015.

[13] E. Karami, M. Shehata, and A. Smith, "Segmentation and tracking of inferior vena cava in ultrasound images using a novel polar active contour algorithm," in *2017*

IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2017, pp. 745–749.

[14] E. Karami, M. S. Shehata, and A. Smith, “Estimation and tracking of AP-diameter of the inferior vena cava in ultrasound images using a novel active circle algorithm,” *Comput. Biol. Med.*, vol. 98, pp. 16–25, 2018.

[15] E. Karami, M. Shehata, P. McGuire, and A. Smith, “A semi-automated technique for internal jugular vein segmentation in ultrasound images using active contours,” in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, 2016, pp. 184–187.

[16] E. Karami, M. S. Shehata, and A. Smith, “Semi-automatic Algorithms for Estimation and Tracking of AP-diameter of the IVC in Ultrasound Images,” *J. Imaging*, vol. 5, no. 1, p. 12, 2019.

[17] P. J. Armstrong, R. Sutherland, and D. H. Scott, “The effect of position and different manoeuvres on internal jugular vein diameter size,” *Acta Anaesthesiol. Scand.*, vol. 38, no. 3, pp. 229–231, Apr. 1994.

[18] U. A. Baumann, C. Marquis, C. Stoupis, T. A. Willenberg, J. Takala, and S. M. Jakob, “Estimation of central venous pressure by ultrasound,” *Resuscitation*, vol. 64, no. 2, pp. 193–199, Feb. 2005.

[19] S. Bellows, M. Shehata, J. Smith, P. Mcguire, and A. Smith, “Validation of a Computerized Technique for Automatically Tracking and Measuring the Inferior Vena Cava in Ultrasound Imagery,” *Biosyst. Eng.*, vol. 138, pp. 4–10, Oct. 2015.

[20] S. Bellows, J. Smith, P. Mcguire, and A. Smith, “Validation of a computerized technique for automatically tracking and measuring the inferior vena cava in ultrasound imagery.,” *Stud. Health Technol. Inform.*, vol. 207, pp. 183–92, 2014.

[21] X. Qian, J. Wang, S. Guo, and Q. Li, “An active contour model for medical image segmentation with application to brain CT image,” *Med. Phys.*, vol. 40, no. 2, p. 21911, 2013.

[22] C. Azzopardi, Y. A. Hicks, and K. P. Camilleri, “Automatic Carotid ultrasound segmentation using deep Convolutional Neural Networks and phase congruency maps,” in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, 2017, pp. 624–628.

[23] Y. M. Li, “Segmentation of Medical Ultrasound Images Using Convolutional Neural Networks with Noisy Activating Functions,” 2016.

[24] J. Smith, M. Shehata, P. McGuire, and A. Smith, “Augmentation of Point of Care Ultrasound,” in *IEEE Newfoundland Electrical and Computer Engineering Conference (NECEC)*, 2016.

[25] J. Smith, M. Shehata, P. McGuire, and A. Smith, “Texture Features for Classification of Vascular Ultrasound,” *International Association for Pattern Recognition (IAPR) 3rd International Workshop on Pattern Recognition for Healthcare Analytics*. 2016.

[26] J. P. Smith, M. Shehata, R. G. Powell, P. F. McGuire, and A. J. Smith, “Physiological Features of the Internal Jugular Vein from B-Mode Ultrasound

Imagery," in *International Symposium on Visual Computing (ISVC)*, 2016, pp. 312–319.

[27] E. Karami, M. S. Shehata, and A. Smith, "Tracking of the internal jugular vein in ultrasound images using optical flow," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017, pp. 1–4.

[28] E. Karami, M. S. Shehata, and A. Smith, "Adaptive polar active contour for segmentation and tracking in ultrasound videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1209–1222, 2018.

[29] H. Gray, *Gray's anatomy: with original illustrations by Henry Carter*. Arcturus Publishing, 2009.

[30] H. Gray, *Anatomy of the human body*. 1918.

[31] D. Van Wynsberghe, C. R. Noback, and R. Carola, *Human anatomy and physiology*. McGraw-Hill College, 1995.

[32] B. Lipton, "Estimation of central venous pressure by ultrasound of the internal jugular vein," *Am. J. Emerg. Med.*, vol. 18, no. 4, pp. 432–434, Jul. 2000.

[33] A. Meyerson and J. Loman, "Internal Jugular Venous Pressure in Man," *Arch. Neurol. Psychiatry*, vol. 27, no. 4, p. 836, Apr. 1932.

[34] S. R. McGee, "Physical examination of venous pressure: A critical review," *Am. Heart J.*, vol. 136, no. 1, pp. 10–18, Jul. 1998.

[35] Dr.S.Venkatesan, "How do you differentiate 'a' wave from 'v' in JVP?" [Online].

Available: https://drsvenkatesan.com/2012/05/20/how-do-you-differentiate-a-wave-from-v-in-jvp/. [Accessed: 10-Nov-2019].

[36] Wapcaplet, “Diagram of the human heart (cropped).svg -,” *Wikimedia Commons*, 2006. [Online]. Available: https://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_(cropped).svg. [Accessed: 10-Nov-2019].

[37] L. S. Bickley, P. G. Szilagyi, and B. Bates, *Bates’ guide to physical examination and history taking*. Lippincott Williams & Wilkins, 2009.

[38] H. Semat and R. Katz, “*Physics*, Chapter 8: Hydrostatics (Fluids at Rest),” *Robert Katz Publ.*, 1958.

[39] G. Ciuti, D. Righi, L. Forzoni, A. Fabbri, and A. M. Pignone, “Differences between internal jugular vein and vertebral vein flow examined in real time with the use of multigate ultrasound color Doppler,” *AJNR.American J. Neuroradiol.*, vol. 34, no. 10, pp. 2000–2004, Oct. 2013.

[40] J. M. Valdueza, T. von Munster, O. Hoffman, S. Schreiber, and K. M. Einhaupl, “Postural dependency of the cerebral venous outflow,” *Lancet*, vol. 355, no. 9199, pp. 200–201, Jan. 2000.

[41] A. J. Layon, A. Gabrielli, and W. A. Friedman, *Textbook of Neurointensive Care*. London: Springer London, 2013.

[42] R. M. Elias *et al.*, “Relationship of pharyngeal water content and jugular volume with severity of obstructive sleep apnea in renal failure,” *Nephrol. Dial.*

Transplant., vol. 28, no. 4, pp. 937–944, Apr. 2013.

[43] F. Guarracino, B. Ferro, F. Forfori, P. Bertini, L. Magliacano, and M. R. Pinsky, "Jugular vein distensibility predicts fluid responsiveness in septic patients," *Crit. Care*, vol. 18, no. 6, p. 647, Dec. 2014.

[44] S. P. Donahue, J. P. Wood, B. M. Patel, and J. V. Quinn, "Correlation of sonographic measurements of the internal jugular vein with central venous pressure," *Am. J. Emerg. Med.*, vol. 27, no. 7, pp. 851–855, Sep. 2009.

[45] H. M. Jassim *et al.*, "IJV collapsibility index vs IVC collapsibility index by point of care ultrasound for estimation of CVP: a comparative study with direct estimation of CVP.," *Open Access Emerg. Med.*, vol. 11, pp. 65–75, 2019.

[46] K. Killu, V. Coba, Y. Huang, T. Andrezejewski, and S. Dulchavsky, "Internal jugular vein collapsibility index associated with hypovolemia in the intensive care unit patients," *Crit. Ultrasound J.*, vol. 2, no. 1, pp. 13–17, Jul. 2010.

[47] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[48] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[49] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.

[50] S. Liu *et al.*, “Deep learning in medical ultrasound analysis: A review,” *Engineering*, 2019.

[51] J. Chen, J. Li, X. Ding, C. Chang, X. Wang, and D. Ta, “Automated Identification and Localization of the Inferior Vena Cava Using Ultrasound: An Animal Study,” *Ultrason. Imaging*, vol. 40, no. 4, pp. 232–244, Jul. 2018.

[52] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[53] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[54] P. Domingos, “Mining Social Networks for Viral Marketing.”

[55] B. Liu, “Sentiment Analysis and Opinion Mining,” *Synth. Lect. Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, May 2012.

[56] R. Upstill-Goddard, D. Eccles, J. Fliege, and A. Collins, “Machine learning approaches for the discovery of gene-gene interactions in disease data,” *Brief. Bioinform.*, vol. 14, no. 2, pp. 251–260, Mar. 2013.

[57] R. Berkow, M. H. Beers, and A. J. Fletcher, “Abnormal heart rhythms,” *Merck Man. Med. information, home Ed. Merck Res. Lab. New Jersey*, vol. 79, 1997.

[58] S. R. Braun, *Respiratory Rate and Pattern*. Butterworths, 1990.

[59] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn.*

Res., vol. 12, pp. 2825–2830, 2011.

[60] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inf. theory*, vol. 8, no. 2, pp. 179–187, 1962.

[61] J. Flusser and T. Suk, "Rotation moment invariants for recognition of symmetric objects," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3784–3790, 2006.

[62] F. Podczeck, "A shape factor to assess the shape of particles using image analysis," *Powder Technol.*, vol. 93, no. 1, pp. 47–53, 1997.

[63] A. M. Bouwman, J. C. Bosma, P. Vonk, J. H. A. Wesselingh, and H. W. Frijlink, "Which shape factor (s) best describe granules?," *Powder Technol.*, vol. 146, no. 1, pp. 66–72, 2004.

[64] R. M. Rangayyan, N. M. El-Faramawy, J. E. L. Desautels, and O. A. Alim, "Measures of acutance and shape for classification of breast tumors," *IEEE Trans. Med. Imaging*, vol. 16, no. 6, pp. 799–810, 1997.

[65] C. F. Mora and A. K. H. Kwan, "Sphericity, shape factor, and convexity measurement of coarse aggregate for concrete using digital image processing," *Cem. Concr. Res.*, vol. 30, no. 3, pp. 351–358, 2000.

[66] J. Flusser and T. Suk, "Pattern recognition by affine moment invariants," *Pattern Recognit.*, vol. 26, no. 1, pp. 167–174, 1993.

[67] E. Welzl, "Smallest enclosing disks (balls and ellipsoids)," in *New results and new trends in computer science*, Springer, 1991, pp. 359–370.

[68] J. O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin, "An optimal algorithm for finding minimal enclosing triangles," *J. Algorithms*, vol. 7, no. 2, pp. 258–269, 1986.

[69] G. T. Toussaint, "Solving geometric problems with the rotating calipers," in *Proc. IEEE Melecon*, 1983, vol. 83, p. A10.

[70] K. S. Vidya, E. Y. K. Ng, U. R. Acharya, S. M. Chou, R. San Tan, and D. N. Ghista, "Computer-aided diagnosis of Myocardial Infarction using ultrasound images with DWT, GLCM and HOS methods: A comparative study," *Comput. Biol. Med.*, vol. 62, pp. 86–93, 2015.

[71] X. Yang *et al.*, "Ultrasound GLCM texture analysis of radiation-induced parotid-gland injury in head-and-neck cancer radiotherapy: an in vivo study of late toxicity," *Med. Phys.*, vol. 39, no. 9, pp. 5732–5739, 2012.

[72] T. Yun and H. Shu, "Ultrasound image segmentation by spectral clustering algorithm based on the curvelet and GLCM features," in *Electrical and Control Engineering (ICECE), 2011 International Conference on*, 2011, pp. 920–923.

[73] D. Mitrea, M. Socaciu, R. Badea, and A. Golea, "Texture based characterization and automatic diagnosis of the abdominal tumors from ultrasound images using third order GLCM features," in *Image and Signal Processing (CISP), 2011 4th International Congress on*, 2011, vol. 3, pp. 1558–1562.

[74] R. M. Haralick, K. Shanmugam, and others, "Textural features for image classification," *IEEE Trans. Syst. Man. Cybern.*, no. 6, pp. 610–621, 1973.

[75] J. Nunez-Iglesias, “Scikit-Image Dev Docs: Feature Module,” *Scikit-image.org*. 2016.

[76] S. van der Walt *et al.*, “scikit-image: image processing in {P}ython,” *PeerJ*, vol. 2, p. e453, 2014.

[77] D.-C. He, L. Wang, and J. Guibert, “Texture discrimination based on an optimal utilization of texture features,” *Pattern Recognit.*, vol. 21, no. 2, pp. 141–146, 1988.

[78] D. Zwillinger, S. Kokoska, B. Raton, L. New, and Y. Washington, “standard probability and Statistics tables and formulae CRC,” 2000.

[79] C. P. Chung *et al.*, “Jugular venous hemodynamic changes with aging,” *Ultrasound Med. Biol.*, vol. 36, no. 11, pp. 1776–1782, Nov. 2010.

[80] Scott Lundberg, “Interpretable Machine Learning with XGBoost - Towards Data Science.” [Online]. Available: https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27. [Accessed: 28-Jul-2019].

[81] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions.” pp. 4765–4774, 2017.

[82] S. M. Lundberg *et al.*, “Explainable AI for Trees: From Local Explanations to Global Understanding,” May 2019.

[83] J. P. Smith, M. Shehata, P. F. McGuire, and A. J. Smith, “Features of Internal Jugular Vein Contours for Classification,” in *International Symposium on Visual Computing (ISVC)*, 2016, vol. 10073 LNCS, pp. 421–430.

[84] R. Bellman, *Dynamic programming*. Princeton University Press, 2010.

[85] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[86] J. Li *et al.*, “Feature Selection: A Data Perspective,” *arXiv Prepr. arXiv1601.07996*, 2016.

[87] K. C. Jundong Li, “jundongl/scikit-feature: open-source feature selection repository in python (DMML Lab@ASU).” [Online]. Available: https://github.com/jundongl/scikit-feature. [Accessed: 21-Jul-2019].

[88] Chapman Siu, “chappers/scikit-feature.” [Online]. Available: https://github.com/chappers/scikit-feature. [Accessed: 21-Jul-2019].

[89] C. M. Bishop and C. M., *Neural networks for pattern recognition*. Clarendon Press, 1995.

[90] R. O. Duda, P. E. (Peter E. Hart, and D. G. Stork, *Pattern classification*. Wiley, 2001.

[91] J. W. Pratt, “F. Y. Edgeworth and R. A. Fisher on the Efficiency of Maximum Likelihood Estimation,” *The Annals of Statistics*, vol. 4. Institute of Mathematical Statistics, pp. 501–514.

[92] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, “Trace Ratio Criterion for Feature Selection.”

[93] M. Robnik-Šikonja and I. Kononenko, “Theoretical and Empirical Analysis of

ReliefF and RReliefF," *Mach. Learn.*, vol. 53, no. 1/2, pp. 23–69, 2003.

[94] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[95] D. D. Lewis, "Feature Selection and Feature Extract ion for Text Categorization."

[96] Hanchuan Peng, Fuhui Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[97] R. Battiti, "Using Mutual Information for Selecting Features in Supervised Neural Net Learning," *IEEE Trans. neural networks*, vol. 5, no. 4, pp. 537–550, 1994.

[98] D. Lin and X. Tang, "Conditional Infomax Learning: An Integrated Framework for Feature Extraction and Fusion," in *European Conference on Computer Vision*, 2006, pp. 68–82.

[99] H. H. Yang and J. Moody, "Data Visualization and Feature Selection: New Algorithms for Nongaussian Data."

[100] F. Fleuret, "Fast Binary Feature Selection with Conditional Mutual Information," *J. Mach. Learn. Res.*, vol. 5, no. Nov, pp. 1531–1555, 2004.

[101] A. Jakulin, "Machine learning based on attribute interactions," Univerza v Ljubljani, 2005.

[102] P. E. Meyer, C. Schretter, and G. Bontempi, "Information-Theoretic Feature Selection in Microarray Data Using Variable Complementarity," *IEEE J. Sel. Top.*

Signal Process., vol. 2, no. 3, pp. 261–274, 2008.

[103] L. Yu and H. Liu, “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution,” in *International Conference on Machine Learning*, 2003, vol. 3, pp. 856–863.

[104] W. J. Adams, *The life and times of the central limit theorem*. American Mathematical Society, 2009.

[105] Huan Liu and R. Setiono, “Chi2: feature selection and discretization of numeric attributes,” in *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pp. 388–391.

[106] S. Wright, “The Interpretation of Population Structure by F-Statistics with Special Regard to Systems of Mating,” *Evolution (N. Y).*, vol. 19, no. 3, p. 395, Sep. 1965.

[107] M. A. Hall, “Correlation-based feature selection for machine learning,” The University of Waikato, 1999.

[108] J. Zhou, D. Foster, R. Stine, and L. Ungar, “Streaming Feature Selection using Alpha-investing,” 2005.

[109] I. Guyon and A. M. De, “An Introduction to Variable and Feature Selection André Elisseeff,” 2003.

[110] J. McCarthy, “Artificial intelligence: a paper symposium: Professor Sir James Lighthill, FRS. Artificial Intelligence: A General Survey. In: Science Research Council, 1973,” *Artif. Intell.*, vol. 5, no. 3, pp. 317–322, Sep. 1974.

[111] T. Hastie, R. Tibshirani, and J. Friedman, “Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction.”

[112] S. M. Stigler, “Gauss and the Invention of Least Squares,” *Ann. Stat.*, vol. 9, no. 3, pp. 465–474, May 1981.

[113] A. E. Hoerl and R. W. Kennard, “Ridge Regression: Applications to Nonorthogonal Problems,” *Technometrics*, vol. 12, no. 1, pp. 69–82, Feb. 1970.

[114] A. N. Tikhonov, “On the stability of inverse problems,” in *Dokl. Akad. Nauk SSSR*, 1943, vol. 39, pp. 195–198.

[115] A. Krogh· and J. A. Hertz, “A Simple Weight Decay Can Improve Generalization.”

[116] F. Santosa and W. W. Symes, “Linear Inversion of Band-Limited Reflection Seismograms,” *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 4, pp. 1307–1330, Oct. 1986.

[117] R. Tibshirani, “Regression Shrinkage and Selection Via the Lasso,” *J. R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[118] R. Tibshirani, I. Johnstone, T. Hastie, and B. Efron, “Least angle regression,” *Ann. Stat.*, vol. 32, no. 2, pp. 407–499, Apr. 2004.

[119] R. A. Fisher and others, “138: The Use of Multiple Measurements in Taxonomic Problems.,” 1936.

[120] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and

organization in the brain.," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[121] R. Quinian and J. Ross, *C4.5 : programs for machine learning.* Morgan Kaufmann, 1993.

[122] C. Gini, "Concentration and dependency ratios," *Riv. di Polit. Econ.*, vol. 87, pp. 769–792, 1997.

[123] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation Forest: A New Classifier Ensemble Method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, Oct. 2006.

[124] J. Loyal, "joshloyal/RotationForest: Implementation of the Rotation Forest by Rodriques et al. 2006." [Online]. Available: https://github.com/joshloyal/RotationForest. [Accessed: 21-Jul-2019].

[125] "digital-idiot/RotationForest: Implementation of the Rotation Forest by Rodriques et al. 2006." [Online]. Available: https://github.com/digital-idiot/RotationForest. [Accessed: 21-Jul-2019].

[126] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[127] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proc. Natl. Acad. Sci.*, vol. 99, no. 10, pp. 6567–6572, May 2002.

[128] L. Breiman, "Arcing The Edge."

[129] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Ann. Stat.*, pp. 1189–1232, 2001.

[130] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, Apr. 2000.

[131] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," 1996.

[132] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[133] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[134] T. Chen and C. Guestrin, "XGBoost," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016, pp. 785–794.

[135] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." pp. 3146–3154, 2017.

[136] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[137] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.

[138] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[139] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” *arXiv Prepr. arXiv1406.1078*, Jun. 2014.

[140] G. Weiss, Y. Goldberg, and E. Yahav, “On the practical computational power of finite precision RNNs for language recognition,” *arXiv Prepr. arXiv1805.04908*, 2018.

[141] C. M. Bishop, *Pattern recognition and machine learning*. Springer Science+ Business Media, 2006.

[142] “Welcome to Python.org.” [Online]. Available: https://www.python.org/. [Accessed: 28-Jul-2019].

[143] “Enterprise Container Platform | Docker.” [Online]. Available: https://www.docker.com/. [Accessed: 28-Jul-2019].

[144] T. Lorchirachoonkul *et al.*, “Anatomical variations of the internal jugular vein: implications for successful cannulation and risk of carotid artery puncture,” *Singapore Med. J.*, vol. 53, no. 5, pp. 325–328, May 2012.

[145] E. Erbabacan, G. M. Koksal, B. Ekici, G. Kaya, and F. Altindas, “Effects of different modes of ventilation and head position on the size of the vena jugularis interna,” *Swiss Medical Weekly*, vol. 144. Apr-2014.

[146] T. Saitoh *et al.*, “Ultrasound analysis of the relationship between right internal jugular vein and common carotid artery in the left head-rotation and head-flexion

position," *Heart Vessels*, vol. 28, no. 5, pp. 620–625, Sep. 2013.