FUNDAMENTALS OF DATA MINING

Australian Sign Language Signs

SEMİHA HAZEL ÇAVUŞ 150115045

ZEYNEP KUMRALBAŞ 150115051

Problem Definition

- Observation and comparison of performances of 5 classifiers on Australian sign language signs:
  - C4.5
  - Naive Bayes
  - Neural Networks with 1 hidden layer
  - Neural Networks with 2 hidden layers
  - Support vector machines

Data Collection

Nintendo PowerGlove

Data Collection Methodology

- Examples of 95 signs were collected from five signers with a total of 6650 sign samples.

Signer	Description	Sessions	Total samples/sign
Adam	Sign linguist - PhD completed in area.	2	8
Andrew	Natural signer - signing since youth	3	8
John	Professional Auslan interpreter	5	18
Stephen	Professional Auslan interpreter	4	16
Waleed	The researcher. Novice signer	4	20

- For example, each sign is repeated totally 8 times in 2 sessions by Adam.

Data set characteristics: Multivariate, Time-Series

Total number of instances: 6650

Attribute Characteristics: Categorical, Real

Number of attributes: 15

0.000000	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.750000	0.500000	0.750000	0.750000	0x0	0x1	0x0	0x3F
0.015625	0.007812	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.750000	0.500000	0.750000	0.750000	0x0	0x1	0x0	0x3F
0.031250	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.750000	0.500000	0.750000	0.750000	0x0	0x1	0x0	0x3F
0.031250	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.750000	0.500000	0.750000	0.750000	0x0	0x1	0x0	0x3F
0.023438	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.750000	0.500000	0.750000	0.750000	0xFF	0x1	0x0	0x3F
0.023438	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.750000	0.500000	0.750000	0.750000	0xFF	0x1	0x0	0x3F
0.023438	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.750000	0.500000	0.750000	0.750000	0xFF	0x1	0x0	0x3F
0.007812	-0.007812	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.750000	0.500000	0.750000	0.750000	0xFF	0x1	0x0	0x3F
0.023438	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.500000	0.500000	0.500000	0.500000	0xFF	0x1	0x0	0x3F
0.031250	0.000000	0.007812	0.000000	-1.000000	-1.000000	0.250000	0.500000	0.500000	0.500000	0.500000	0xFF	0x1	0x0	0x3F
0.015625	-0.007812	0.007812	0.000000	-1.000000	-1.000000	0.250000	0.500000	0.500000	0.500000	0.500000	0xFF	0x1	0x0	0x3F
0.031250	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.500000	0.500000	0.500000	0.500000	0xFF	0x1	0x0	0x3F
0.031250	0.000000	0.000000	0.083333	-1.000000	-1.000000	0.250000	0.500000	0.500000	0.500000	0.500000	0xFF	0x1	0x0	0x3F
0.031250	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.500000	0.500000	0.750000	0.750000	0xFF	0x1	0x0	0x3F
0.031250	0.000000	0.000000	0.083333	-1.000000	-1.000000	0.250000	0.250000	0.750000	0.750000	0.750000	0xFF	0x1	0x0	0x3F
0.031250	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.000000	0.000000	0.000000	0.000000	0xFF	0x0	0x0	0x3F
0.015625	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.000000	0.750000	0.000000	0.000000	0xFF	0x0	0x0	0x3F
0.007812	0.007812	0.000000	0.083333	-1.000000	-1.000000	0.250000	0.000000	0.000000	0.000000	0.000000	0xFF	0x0	0x0	0x3F
0.015625	0.007812	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.000000	0.000000	0.000000	0.000000	0xFF	0x0	0x0	0x3F
0.000000	0.039062	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.000000	0.000000	0.000000	0.000000	0xFF	0x0	0x0	0x3F
[illegible]	0.085938	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.000000	0.000000	0.000000	0.000000	0xFF	0x0	0x0	[illegible]
[illegible]	[illegible]	0.000000	0.000000	-1.000000	-1.000000	0.250000	0.000000	0.000000	0.000000	0.000000	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	-1.000000	-1.000000	0.250000	0.000000	0.000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

x, y, z, roll, pitch, yaw, thumb, fore, index, ring, little, keycode, gs1, gs2, receiver values

SIGNER: ADAM   SIGN: ALIVE

Data Preprocessing

- Missing values
- Feature Creation
- Normalization
- Training set, test set

- A sign file collected from Stephen was empty.
- We deleted this file.

Data Preprocessing

- Missing values
- Feature Creation
- Normalization
- Training set, test set

- **Number of frames:** number of rows
- **Distance:** a measure of the total distance covered by the sign
- **Energy:** some signs can be more energetic (momentarily faster motion)
- **Bounding boxes:** some signs can be bigger than others, (x_min, y_min, z_min) (x_max, y_max, z_max)
- **Simple Time Division**: try a pattern-matching approach at a very low level, x, y, z positions, rotation value, finger positions

Data Preprocessing

- Missing values
- Feature Creation
- Normalization
- Training set, test set

- All values are normalized into 0-1.

Data Preprocessing

- Missing values
- Feature Creation
- Normalization
- Training set, test set

- Test size = 0.15
- Cross validation 5-fold

Python Implementation

- **IDE/environment**
- Implementation
- Libraries

- Environment :
- IDE :

Python Implementation

- IDE/environment
- **Implementation**
- Libraries

Decision Tree (C4.5)

- Libraries : ChefBoost, Sklearn, Matplotlib
- The C4.5 decision tree is directly built with model function of ChefBoost.
- Model evaluation metrics are calculated with Sklearn functions.

Python Implementation

- IDE/environment
- **Implementation**
- Libraries

Naive Bayes

- Libraries : Sklearn, Matplotlib
- The Naive Bayes classifier is implemented as GaussianNB() function.
- Model evaluation metrics are calculated with Sklearn functions.

Python Implementation

- IDE/environment
- **Implementation**
- Libraries

Neural Networks

- Libraries: Keras, Sklearn
- Categorical data are transformed into numbers.
- One Hidden Layer NN
  - Input layer has 49 neurons.
  - Hidden layer has 95 neurons and activation function is hyperbolic tangent (tanh).
  - Output layer has 95 neurons and activation function is softmax.

Python Implementation

- IDE/environment
- **Implementation**
- Libraries

Neural Networks

- Two Hidden Layer NN
  - Input layer has 49 neurons
  - First hidden layer has 95 neurons and activation function is hyperbolic tangent (tanh).
  - Second hidden layer has 95 neurons and activation function is hyperbolic tanget (tanh).
  - Output layer has 95 neurons and activation function is softmax.

Python Implementation

- IDE/environment
- **Implementation**
- Libraries

Support Vector Machine

- Libraries: Sklearn
- Best parameters are searched among

  C = [0.001, 0.01, 1, 10], gamma = [0.001,0.01, 0.1, 1]

  using GridSearchCV function from sklearn
- Best parameters are found and they are used:

  kernel='poly', C=0.1, gamma=1

Python Implementation

- IDE/environment
- Implementation
- **Libraries**

NumPy

pandas

scikit learn

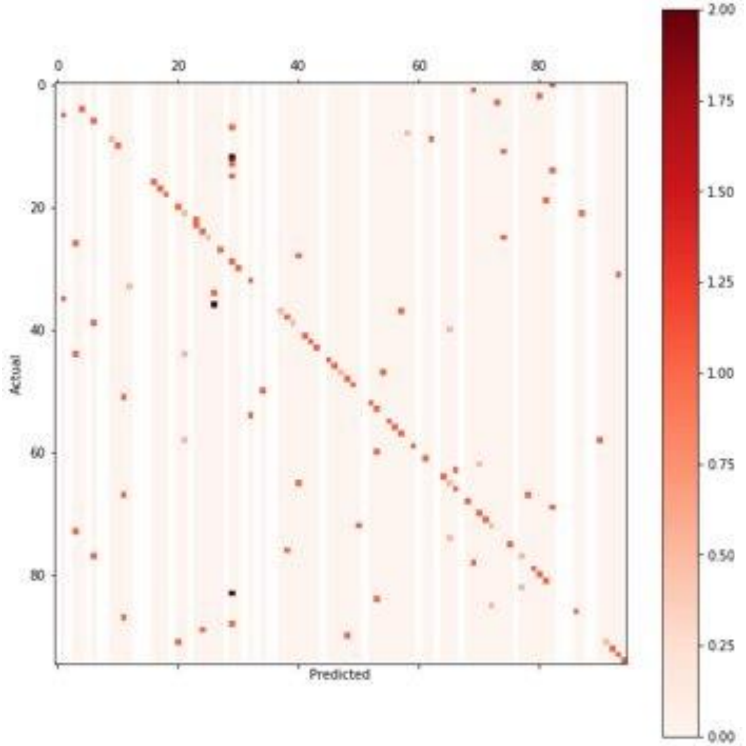
matplotlib

chefboost

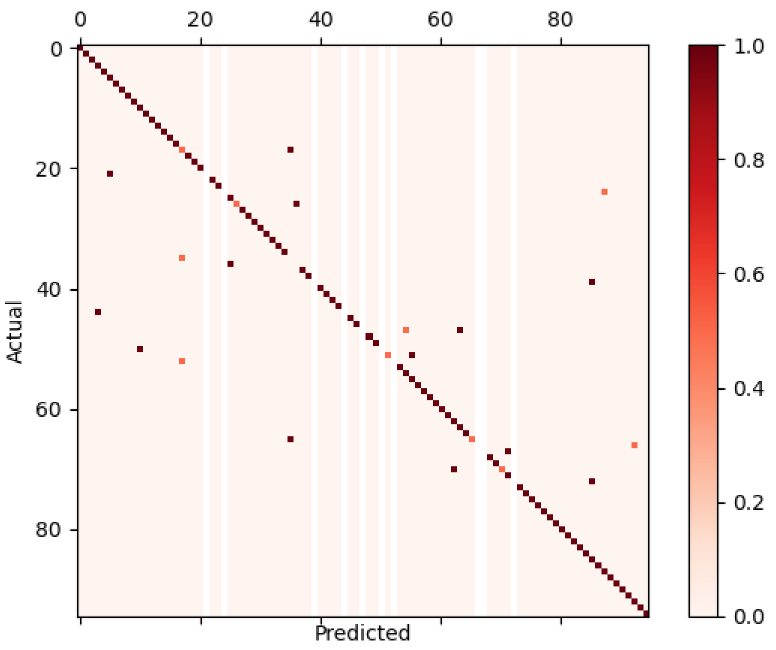
Performance Results

ADAM

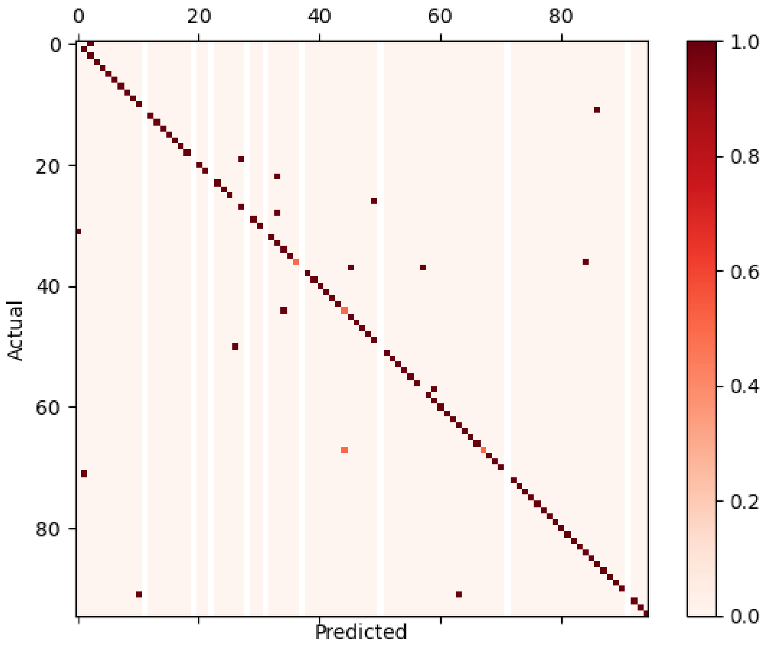
Confusion Matrices



C4.5



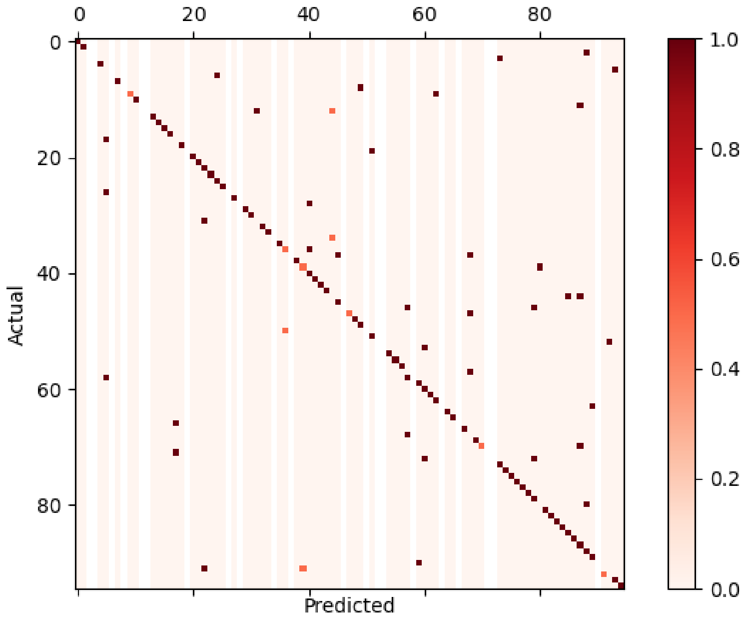
NN-1



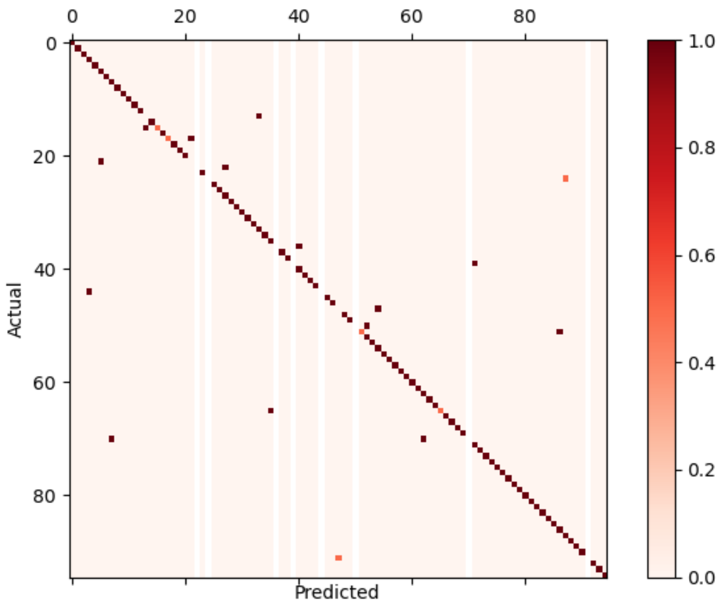
NN-2

ADAM

Confusion Matrices



NV



SVM

ADAM

	C4.5	NB	NN-1	NN-2	SVM
Accuracy	0.47	0.60	0.79	0.79	0.80
Error	0.53	0.40	0.21	0.21	0.20
Sensitivity	0.48	1.20	1.59	1.57	1.61
Specificity	0.99	1.99	1.99	1.99	1.99
Precision	0.70	0.79	0.87	0.87	0.88
Recall	0.48	0.60	0.79	0.78	0.81
F1 score	0.43	0.53	0.74	0.73	0.75

$$ACC_{SVM} > ACC_{NN1} > ACC_{NN2} > ACC_{NB} > ACC_{C4.5}$$

ADAM

Performance Comparison - Error (%)

Our result

Dataset	C4.5	NB	NN-1	NN-2	SVM
Adam	52.6	40.18	20.53	21.0	19.65

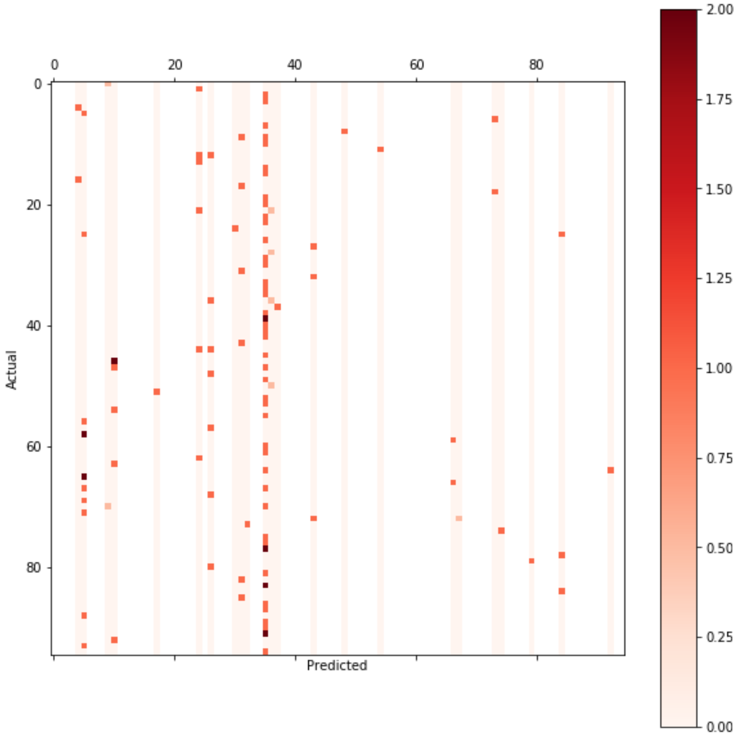
Paper result

Dataset	IBL1	C4.5
Adam	12.6	51.4

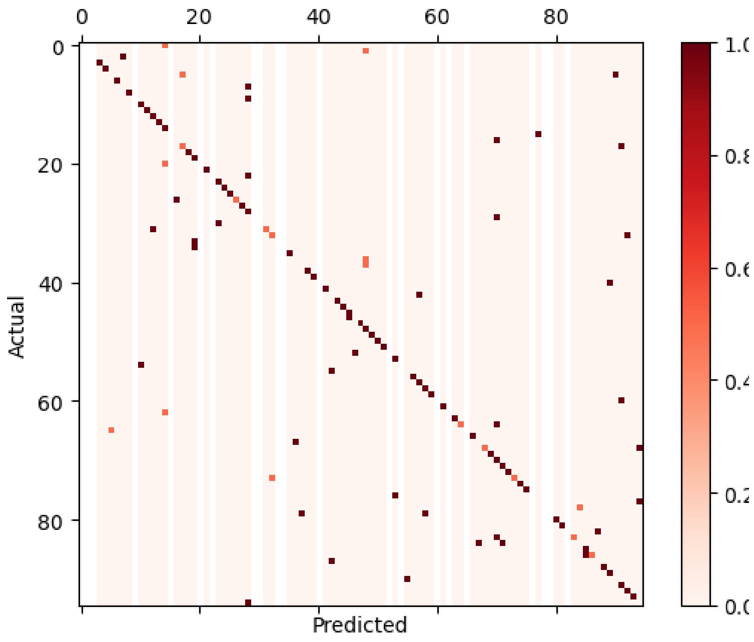
Results of learning algorithms on most effective attributes combined

ANDREW

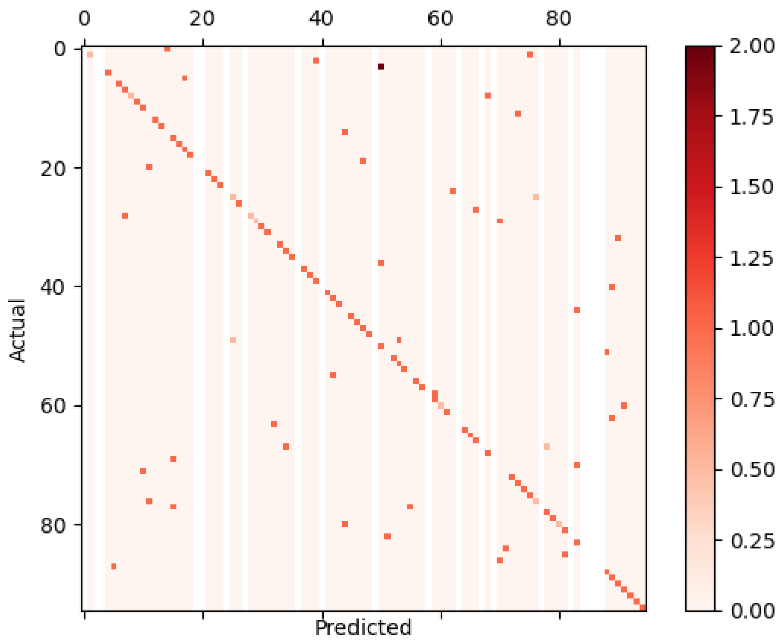
Confusion Matrices



C4.5



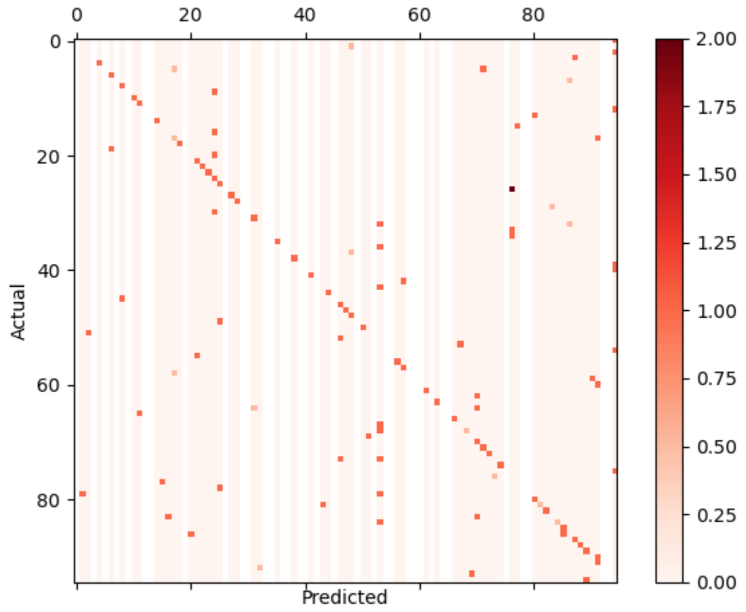
NN-1



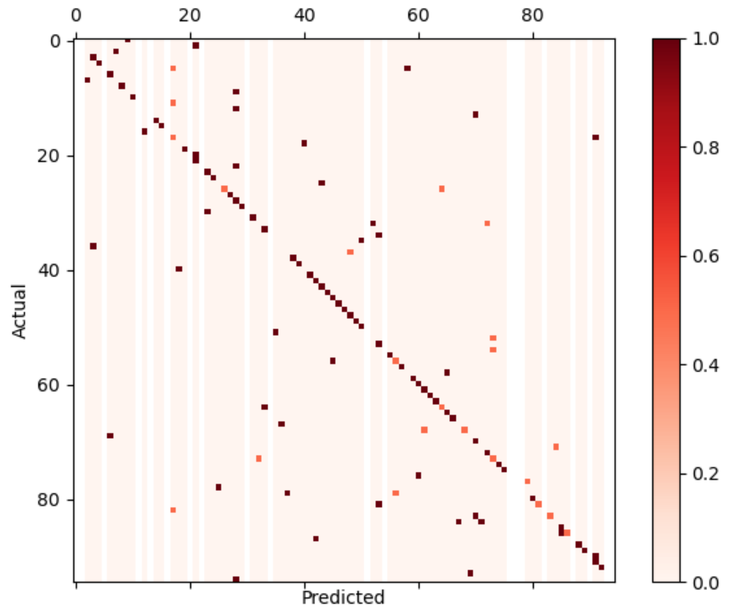
NN-2

ANDREW

Confusion Matrices



NV



SVM

ANDREW

	C4.5	NB	NN-1	NN-2	SVM
Accuracy	0.10	0.39	0.57	0.59	0.52
Error	0.90	0.61	0.43	0.41	0.48
Sensitivity	0.10	0.78	1.14	1.19	1.04
Specificity	0.99	1.99	1.99	1.99	1.99
Precision	0.81	0.66	0.73	0.73	0.67
Recall	0.10	0.39	0.57	0.60	0.52
F1 score	0.06	0.34	0.50	0.54	0.45

$$ACC_{NN2} > ACC_{NN1} > ACC_{SVM} > ACC_{NB} > ACC_{C4.5}$$

ANDREW

Performance Comparison - Error (%)

Our result

Dataset	C4.5	NB	NN-1	NN-2	SVM
Andrew	90.3	61.05	42.98	40.53	48.07

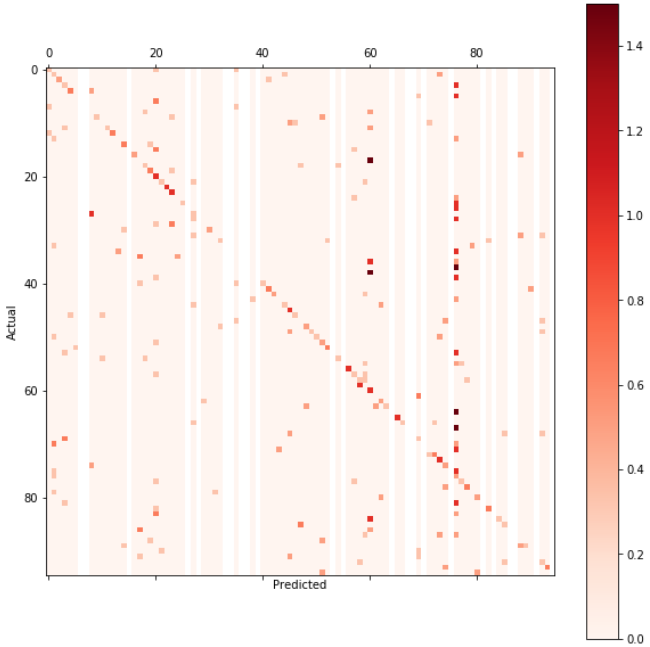
Paper result

Dataset	IBL1	C4.5
Andrew	41.5	66.1

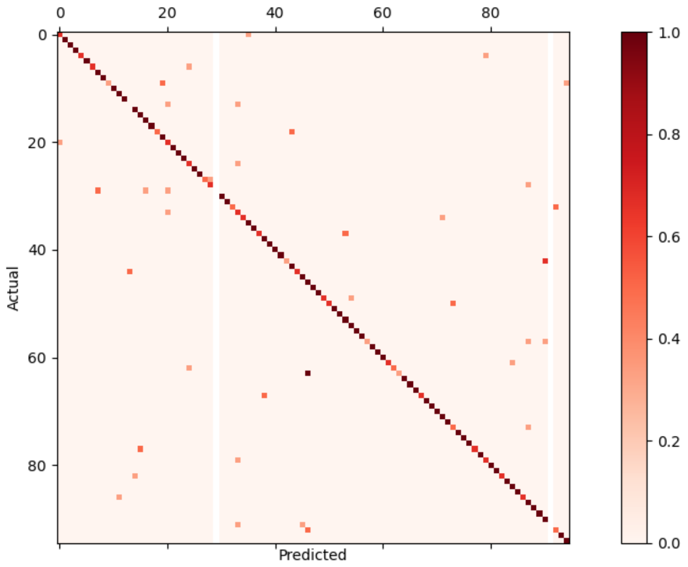
Results of learning algorithms on most effective attributes combined

JOHN

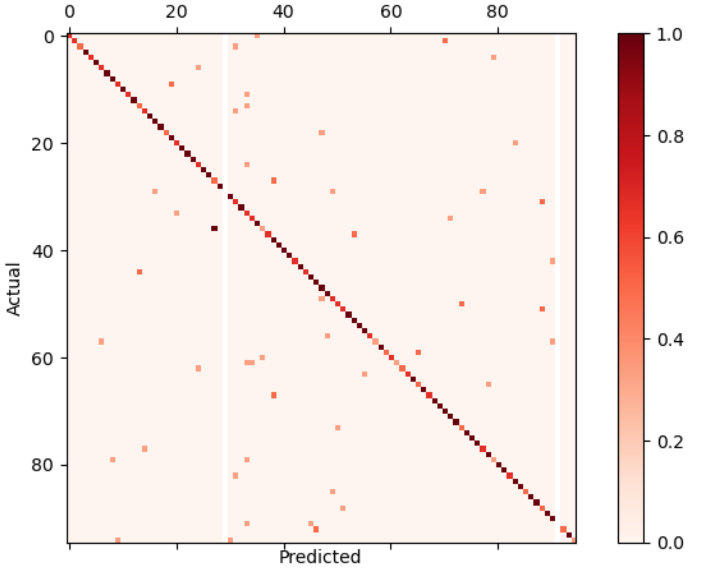
Confusion Matrices



C4.5



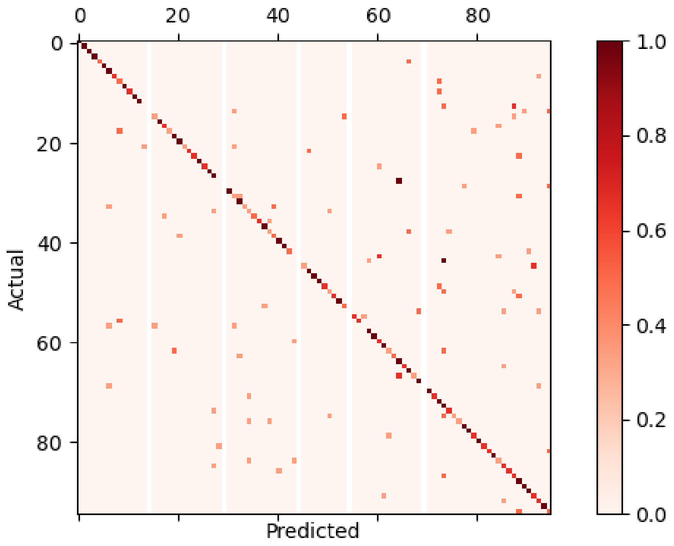
NN-1



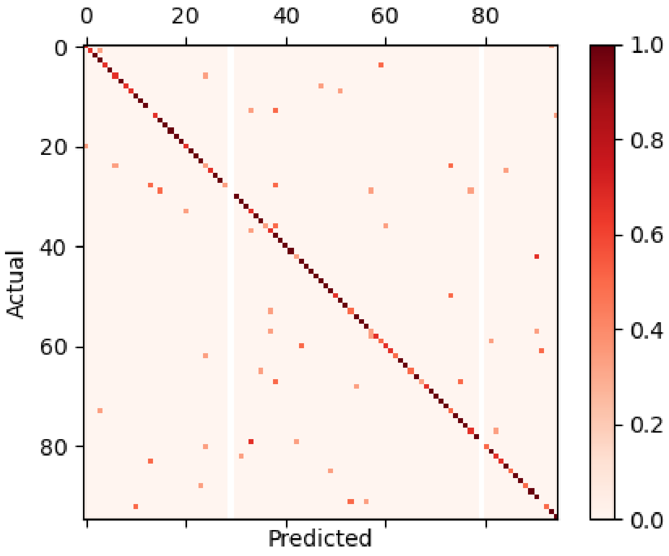
NN-2

JOHN

Confusion Matrices



NV



SVM

JOHN

	C4.5	NB	NN-1	NN-2	SVM
Accuracy	0.29	0.64	0.82	0.81	0.79
Error	0.71	0.36	0.18	0.19	0.21
Sensitivity	0.30	1.29	1.65	1.61	1.59
Specificity	0.99	1.99	2.00	1.99	1.99
Precision	0.59	0.74	0.86	0.85	0.83
Recall	0.30	0.65	0.82	0.81	0.80
F1 score	0.28	0.63	0.81	0.80	0.78

$$ACC_{NN1} > ACC_{NN2} > ACC_{SVM} > ACC_{NB} > ACC_{C4.5}$$

JOHN

Performance Comparison - Error (%)

Our result

Dataset	C4.5	NB	NN-1	NN-2	SVM
John	70.8	35.72	17.66	19.38	20.70

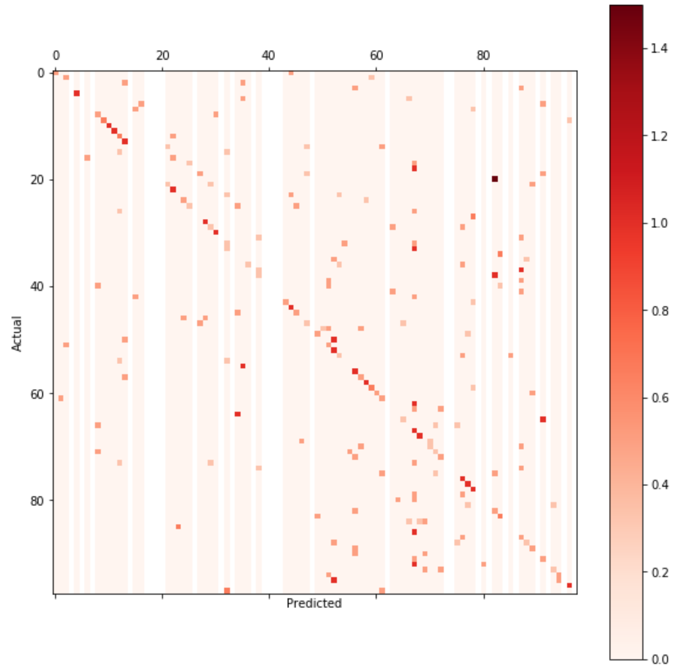
Paper result

Dataset	IBL1	C4.5
John	19.4	45.4

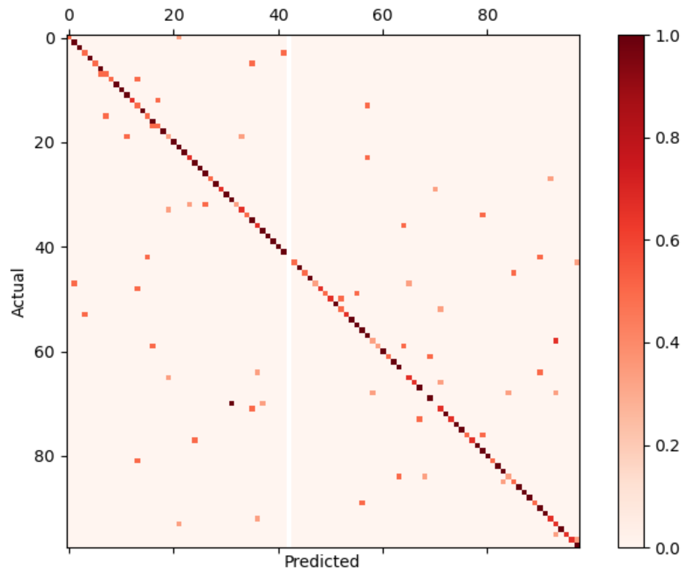
Results of learning algorithms on most effective attributes combined

STEPHEN

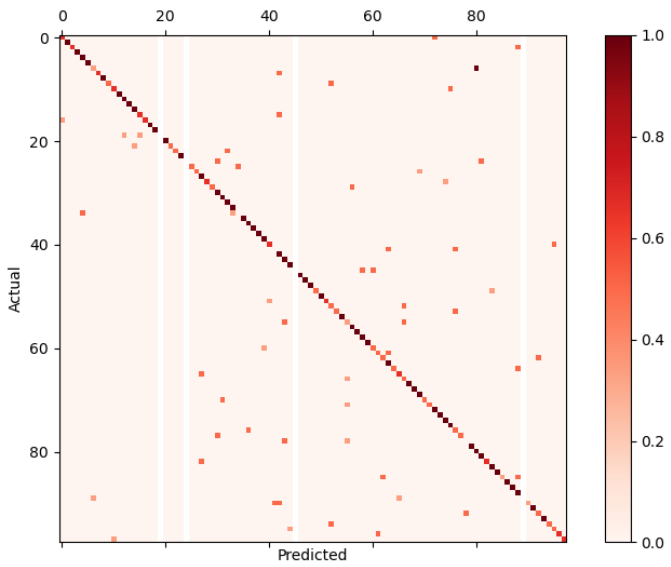
Confusion Matrices



C4.5



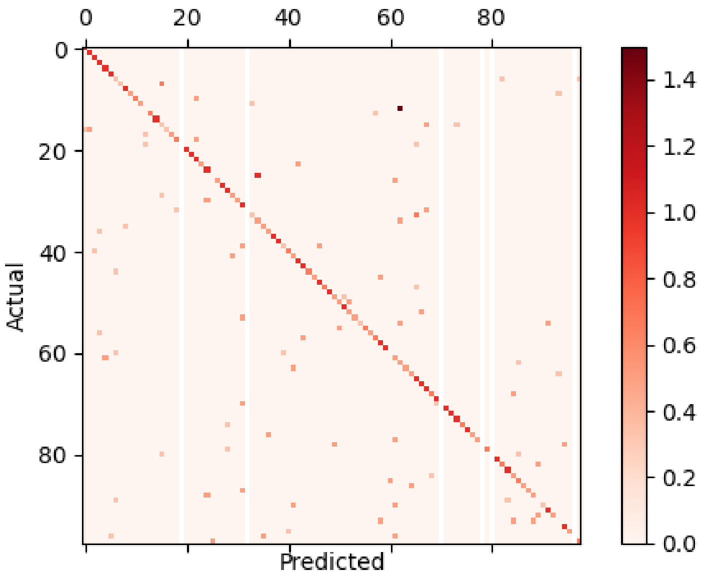
NN-1



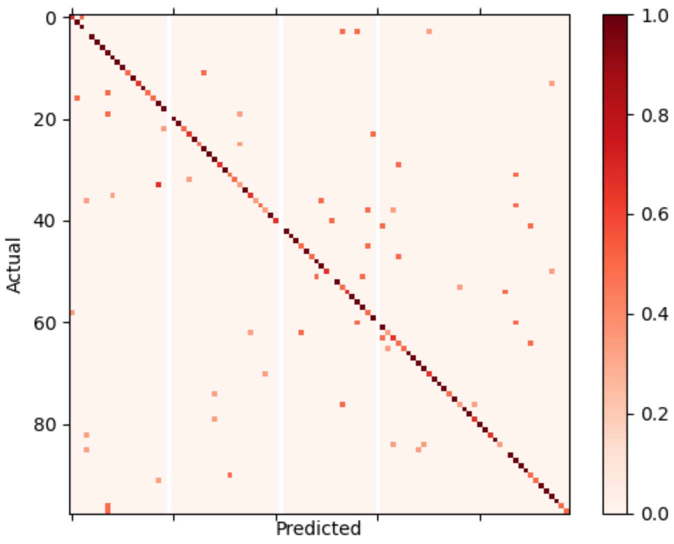
NN-2

STEPHEN

Confusion Matrices



NV



SVM

STEPHEN

	C4.5	NB	NN-1	NN-2	SVM
Accuracy	0.33	0.59	0.76	0.74	0.74
Error	0.67	0.41	0.24	0.26	0.26
Sensitivity	0.33	1.17	1.52	1.47	1.47
Specificity	0.99	1.99	1.99	1.99	1.99
Precision	0.54	0.70	0.80	0.79	0.79
Recall	0.33	0.59	0.76	0.73	0.73
F1 score	0.26	0.56	0.74	0.72	0.72

$$ACC_{NN1} > ACC_{NN2} > ACC_{SVM} > ACC_{NB} > ACC_{C4.5}$$

STEPHEN

Performance Comparison - Error (%)

Our result

Dataset	C4.5	NB	NN-1	NN-2	SVM
Stephen	67.2	41.19	24.0	26.38	26.13

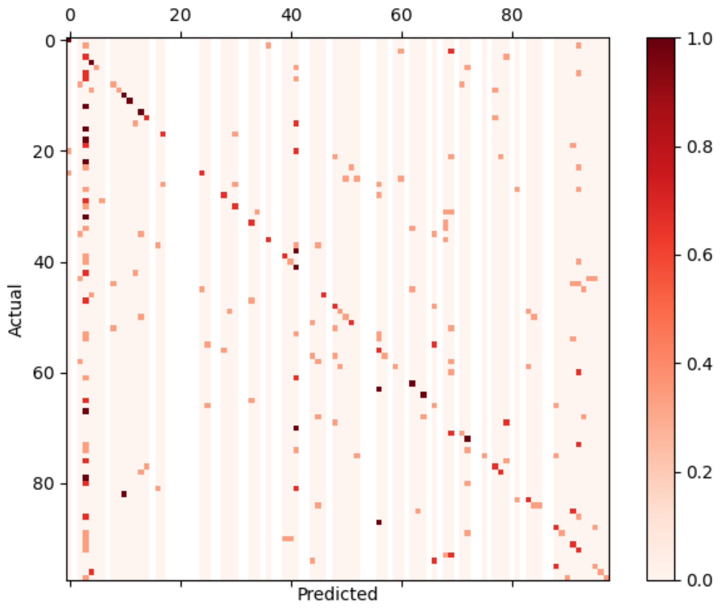
Paper result

Dataset	IBL1	C4.5
Stephen	18.4	52.8

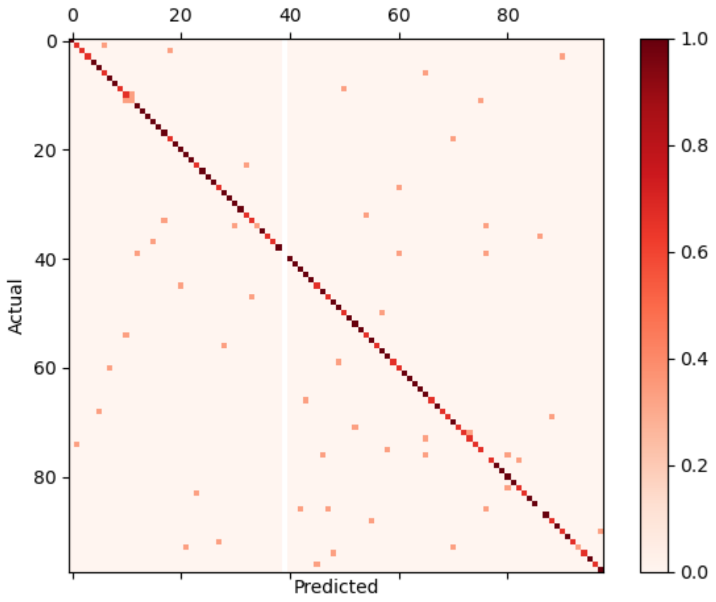
Results of learning algorithms on most effective attributes combined

WALEED

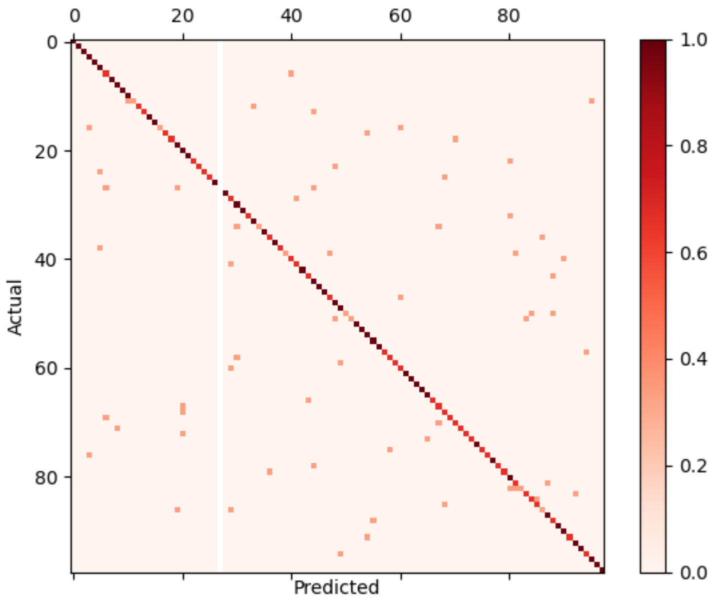
Confusion Matrices



C4.5



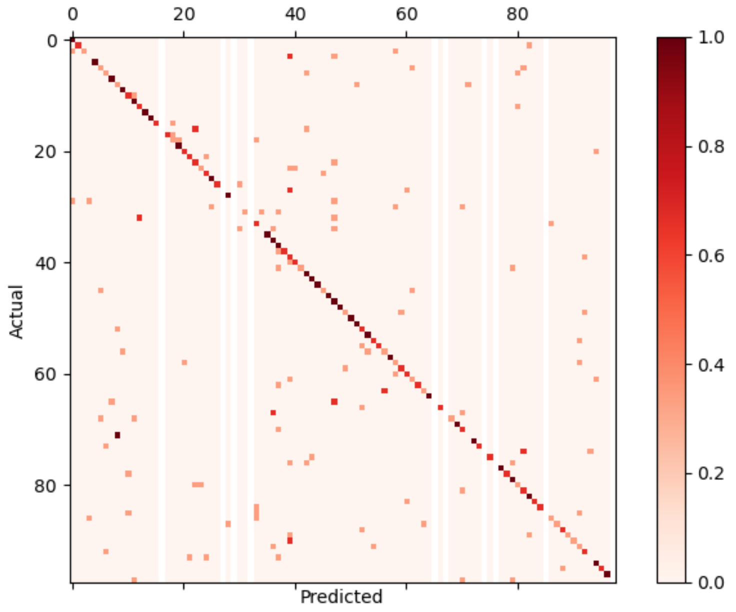
NN-1



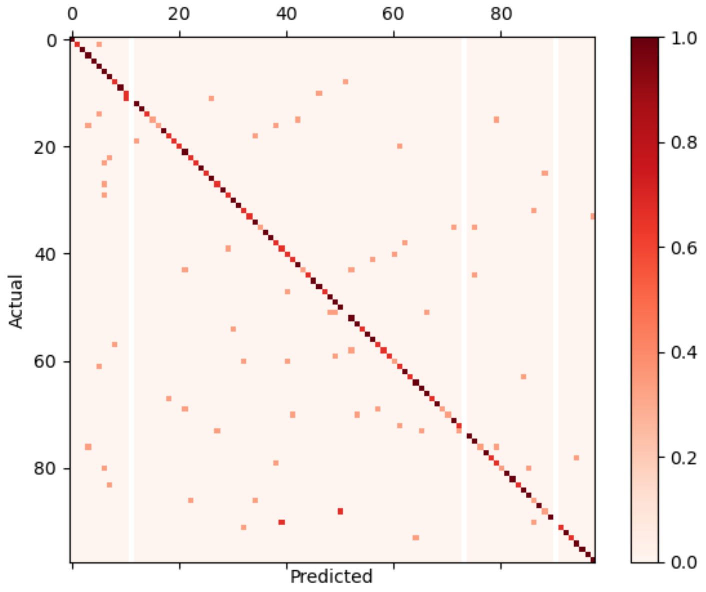
NN-2

WALEED

Confusion Matrices



NV



SVM

WALEED

	C4.5	NB	NN-1	NN-2	SVM
Accuracy	0.28	0.57	0.80	0.79	0.76
Error	0.72	0.43	0.20	0.22	0.24
Sensitivity	0.28	1.14	1.59	1.56	1.52
Specificity	0.99	1.99	1.99	1.99	1.99
Precision	0.61	0.69	0.83	0.81	0.80
Recall	0.28	0.57	0.80	0.79	0.76
F1 score	0.24	0.55	0.79	0.77	0.76

$$ACC_{NN1} > ACC_{NN2} > ACC_{SVM} > ACC_{NB} > ACC_{C4.5}$$

WALEED

Performance Comparison - Error (%)

Our result

Dataset	C4.5	NB	NN-1	NN-2	SVM
Waleed	72.45	43	20.34	22.0	24

Paper result

Dataset	IBL1	C4.5
Waleed	17.0	48.5

Results of learning algorithms on most effective attributes combined

Conclusion

Conclusion

- Our C4.5 decision tree gives worse results than the C4.5 decision tree in the paper.
- Constructing C4.5 decision tree for each person takes too much time.
- Other classifiers give better results than the decision tree in the paper.
- For each person, C4.5 decision trees give the worst accuracy results.
- In general, the accuracies in descending order: one hidden layer neural network, two hidden layer neural network, SVM, NB.

References

- KADOUS, Mohammed Waleed. Auslan sign recognition using computers and gloves. In: *Deaf Studies Research Symposium*. 1998.
- M. W. Kadous, GRASP: Recognition of Australian Sign Language using Instrumented Gloves, Honours thesis, School of Computer Science and Engineering, University of New South Wales, 1995.

Supplementary

$$\Delta x_i = x_i - x_{i-1}$$
$$\Delta y_i = y_i - y_{i-1}$$
$$\Delta z_i = z_i - z_{i-1}$$

$$\Delta^2 x_i = \Delta x_i - \Delta x_{i-1}$$
$$\Delta^2 y_i = \Delta y_i - \Delta y_{i-1}$$
$$\Delta^2 z_i = \Delta z_i - \Delta z_{i-1}$$
$$\Delta_i^2 = \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}$$

$$\Delta_i = \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}$$

$$distance = \sum_{i=1}^{n} \Delta_i$$

$$W = \int_0^T F ds$$

and a discrete approximation of this equation is given by:

$$W = \sum_{i=0}^{n} F \Delta s_i$$

Since $F = ma$ and m is a constant, and converting to the naming we have used then we can say:

$$W \propto \sum_{i=1}^{n} a \Delta_i$$

But $a = \frac{\Delta v}{\Delta t}$ and we have assumed that Δt is a constant. Also Δv is synonymous with Δ_i^2 then we have that[10]:

$$energy = \sum_{i=2}^{n} \Delta_i^2 \Delta_i$$

Supplementary

Mathematically, we can express this in the following way. For the moment, consider only the average value of the x position. Let s_i be the x position in the ith segment, and let d be the number of segments we want to divide the sign into. The the value of s_i is given by:

$$s_i = \sum_{j=l(i)}^{u(i)} \frac{x_j}{u(i) - l(i) + 1}, \ \forall i, 0 \leq i < d$$

where n is the number of frames and

$$l(i) = \lfloor i\tfrac{n}{d} \rfloor + 1$$
$$u(i) = \lfloor (i+1)\tfrac{n}{d} \rfloor$$