

# Modulation classification assignment

**Colab Link:**

<https://colab.research.google.com/drive/1Ait7cozVrKOEb-N7yj6KgC6Rn91Ft7ML?usp=sharing>

---

Eman Rafik Abdelkader (11)

## Dataset

We downloaded the dataset and uploaded it to Google drive then we loaded it in Google Colab using. The raw data is of shape (1200000, 2, 128) which describes 1200000 each having 2 functions specified by 128 coefficients each with 2 labels for each entry specifying the SNR and the MOD.

```
Xd = pickle.load(open("/content/RML2016.10b.dat",'rb'), encoding="latin1")
snrs,mods = map(lambda j: sorted(list(set(map(lambda x: x[j], Xd.keys())))), [1,0])
X = []
lbl = []
for mod in mods:
    for snr in snrs:
        X.append(Xd[(mod,snr)])
        for i in range(Xd[(mod,snr)].shape[0]): lbl.append((mod,snr))
X = np.vstack(X)
print(X.shape)
labels = np.array(lbl)
print(labels.shape)
print(np.array(mods).shape)
del Xd
del lbl

(1200000, 2, 128)
(1200000, 2)
(10,)
```

Then we got the integrals and the derivatives using Poly1d and formed all possible combinations of the raw data, the derivatives and the integrals.

```
X_integ = []
for i in range(len(X)):
    first_func = X[i][0]
    second_func = X[i][1]
    p1 = np.poly1d(first_func)
    d1 = np.polyint(p1)
    p2 = np.poly1d(second_func)
    d2 = np.polyint(p2)
    first_func_integ = np.array(d1)
    second_func_integ = np.array(d2)
    current = []
    current.append(first_func_integ)
    current.append(second_func_integ)
    X_integ.append(current)
X_integ = np.array(X_integ)
print(X_integ.shape)

(1200000, 2, 129)

X_deriv = []
for i in range(len(X)):
    first_func = X[i][0]
    second_func = X[i][1]
    p1 = np.poly1d(first_func)
    d1 = np.polyder(p1)
    p2 = np.poly1d(second_func)
    d2 = np.polyder(p2)
    first_func_derivative = [0]
    first_func_derivative = np.concatenate(([first_func_derivative, np.array(d1)])]
    second_func_derivative = [0]
    second_func_derivative = np.concatenate(([second_func_derivative, np.array(d2)]))
    current = []
    current.append(first_func_derivative)
    current.append(second_func_derivative)
    X_deriv.append(current)
X_deriv = np.array(X_deriv)
print(X_deriv.shape)

(1200000, 2, 128)
```



Then we split the data into 30% for the test data. And 5% of the other 70% will be for the validation data. We used the built in Sklearn split function for that.

## Classification models overall accuracies

### 1. Raw data

Model	Overall accuracy
Logistic regression	0.1580555555555556
Decision Trees	0.3097055555555557
Random Forest	0.4389861111111113
Fully connected dense layer	0.3816
CNN	0.4216
LSTM	0.4595

### 2. Derivatives of the data

Model	Overall accuracy
Logistic regression	0.1602277777777777
Decision Trees	0.3077
Random Forest	0.4402861111111111
Fully connected dense layer	0.4122
CNN	0.4777
LSTM	0.4915

### 3. Integrals of the data

Model	Overall accuracy
Logistic regression	0.1346166666666666



Decision Trees	0.31201944444444446
Random Forest	0.4287361111111111
Fully connected dense layer	0.3293
CNN	0.3023
LSTM	0.3493

#### 4. Using a combination of raw data and the derivatives

Model	Overall accuracy
Logistic regression	0.16105
Decision Trees	0.3112805555555556
Random Forest	0.4411111111111111
Fully connected dense layer	0.4185
CNN	0.4900
LSTM	0.4979

#### 5. Using a combination of raw data and the integrals

Model	Overall accuracy
Logistic regression	0.1574222222222222
Decision Trees	0.308625
Random Forest	0.4430083333333334
Fully connected dense layer	0.3667
CNN	0.4241
LSTM	0.4970

#### 6. Using a combination of the derivatives and the integrals

Model	Overall accuracy



Logistic regression	0.1612444444444444
Decision Trees	0.3102138888888889
Random Forest	0.4412138888888889
Fully connected dense layer	0.4088
CNN	0.4483
LSTM	0.5069

## 7. Using a combination of the raw data, the derivatives and the integrals

Model	Overall accuracy
Logistic regression	0.1626194444444446
Decision Trees	0.3116944444444444
Random Forest	0.4451416666666666
Fully connected dense layer	0.4165
CNN	0.4774
LSTM	0.5078

## The results at SNR = 0dB and confusion matrices

### 1. Raw data

a) Logistic Regression:

```

SNR ==> 0
0.1762524218101301
[[ 188   95   97   199   221   180   228   225   182   202]
 [  9 1170    6   14   16   26    5   76   38  390]
 [ 160  106  166  166  252  160  192  214  172  198]
 [ 152  151  163  206  215  151  169  179  164  234]
 [ 103  318  117  178  188  186   71  223  142  336]
 [ 142  141  165  180  190  173  179  235  156  290]
 [ 133  114  173  173  209  148  219  233  149  224]
 [ 155  126  159  167  204  152  235  256  158  221]
 [ 160   91  183  178  193  156  230  208  169  193]
 [  8 1147   19   22   31    9    4   77   60  449]]
Accuracy: 0.1762524218101301

```

b) Decision Trees:



```
SNR ==> 0
0.2871298090229726
[[291 57 166 193 123 185 278 291 272 61]
 [ 20 766 33 15 133 21 14 18 14 716]
 [185 69 340 131 89 312 226 183 190 61]
 [246 66 137 358 167 97 211 183 235 84]
 [150 128 95 181 517 92 122 138 162 197]
 [ 89 31 230 52 40 959 169 168 83 30]
 [247 35 101 106 59 142 414 447 190 34]
 [215 24 85 83 59 158 425 529 217 38]
 [234 60 185 182 110 149 245 257 281 58]
 [ 30 708 20 38 167 30 35 30 36 732]]
Accuracy: 0.2871298090229726
```

c) Random forest:

```
SNR ==> 0
0.5061233578267647
[[ 419 3 40 44 33 25 489 427 233 10]
 [ 0 1112 0 0 52 0 0 0 0 649]
 [ 144 4 832 16 19 491 125 110 90 5]
 [ 299 8 9 898 57 2 199 177 180 9]
 [ 18 49 0 25 1552 1 12 15 14 142]
 [ 8 1 78 1 4 1521 62 84 10 2]
 [ 129 1 5 6 6 30 610 912 63 0]
 [ 100 2 5 6 8 29 571 1016 44 2]
 [ 457 3 44 61 18 23 418 485 294 7]
 [ 0 801 0 2 158 0 1 0 0 838]]
Accuracy: 0.5061233578267647
```

d) Fully connected layer:

```
SNR ==> 0
562/562 [=====] - 1s 2ms/step - loss: 1.1970 - accuracy: 0.5072
[[ 45 3 40 329 7 19 228 614 484 1]
 [ 0 1532 0 0 0 0 0 0 0 312]
 [ 25 9 259 174 32 649 160 236 257 7]
 [ 11 4 2 1560 36 0 34 71 48 10]
 [ 0 9 0 5 1721 0 0 0 0 77]
 [ 2 3 78 32 18 1247 18 368 19 3]
 [ 2 4 54 107 7 130 81 1221 136 1]
 [ 9 5 44 84 2 161 77 1298 136 0]
 [ 35 1 46 318 4 27 194 605 578 5]
 [ 0 931 0 2 72 0 0 0 0 793]]
Accuracy: 0.5072350845948352
```

e) CNN:

```
SNR ==> 0
564/564 [=====] - 1s 2ms/step - loss: 1.0465 - categorical_accuracy: 0.5488
[[ 448 6 95 112 58 28 84 660 311 8]
 [ 0 1300 40 0 5 6 0 0 0 440]
 [ 86 9 1228 39 31 168 10 52 145 19]
 [ 125 3 37 1368 54 6 18 89 103 6]
 [ 3 25 4 21 1674 1 1 9 2 112]
 [ 36 7 253 18 19 1279 14 151 30 9]
 [ 205 2 24 41 28 27 97 1337 90 4]
 [ 149 2 25 36 26 47 65 1298 80 3]
 [ 366 3 151 135 52 17 71 489 482 5]
 [ 0 955 28 2 95 9 0 0 0 723]]
Accuracy: 0.5487967173117445
```

f) LSTM:

```
SNR ==> 0
561/561 [=====] - 2s 4ms/step - loss: 0.7035 - categorical_accuracy: 0.6683
[[ 442   2   22  227   2   1   2   64 1105   0]
 [  0 1707   0   0   0   0   0   0   0 124]
 [  6   2 1757   2   1   6   0   0   5   0]
 [ 16   0   0 1648   5   0   0   1  95   0]
 [  0   2   0   7 1783   0   0   0   0  29]
 [  3   5  352   3   2 1417   0   8   0   1]
 [ 155   2   11  30   3   4  23 1277  257   0]
 [ 109   3   4  25   3   10  15 1452 141   0]
 [ 340   3   22 214   4   0   1  72 1127   1]
 [  0 1086   0   0  58   0   0   0   0 630]]
Accuracy: 0.6682649420160571
```

## 2. Derivatives of the data

a) Logistic Regression:

```
SNR ==> 0
0.18533078737504213
[[ 234   99  209  218  155  190  167  208  165  182]
 [ 12 1156   11   16   23   13   5   79   22 383]
 [ 197   97 191 193 171 185 160 197 213 208]
 [ 216 175 174 220 191 164 122 196 154 237]
 [ 126 316 137 146 197 109  94 177 139 298]
 [ 190 141 135 163 103 207 148 198 183 244]
 [ 216 116 153 190 149 159 122 260 196 199]
 [ 225 124 161 198 135 160 141 269 168 224]
 [ 216 183 177 201 138 178 162 209 197 187]
 [ 33 1082   22   25   21   20   6   63   35 507]]
Accuracy: 0.18533078737504213
```

b) Decision Trees:

```
SNR ==> 0
0.27917555880040434
[[278   43 152 170 119 164 257 303 266  75]
 [ 20 772   34   17   96   17   22   16   11 715]
 [209   64 336 127   87 300 201 229 202  57]
 [243   60 135 328 176 127 224 212 265  79]
 [161 120   99 164 499   94 139 150 149 164]
 [ 88 19 220   36   37 841 151 222   68   30]
 [280   28  90 108   54 153 376 458 189   24]
 [259   23  96  89   47 152 433 487 192   27]
 [229  55 159 173 115 126 264 272 304  71]
 [ 45 684   21  37 172   29   28  20  28 750]]
Accuracy: 0.27917555880040434
```

c) Random forest:

```
SNR ==> 0
0.5053428317008014
[[ 469   3   28   61   27   23  448  508  268   2]
 [  0 1157   0   0  58   0   0   0   0 552]
 [ 164   3  789   10   16  460  145  118  96   4]
 [ 268   3   12  883   60   6  176  204  168   9]
 [ 22   53   0  25 1511   3   15   14  13 116]
 [  6   0  76   5   3 1575   56  100   5   0]
 [ 147   2   6   5   7  17 583  889   68   0]
 [ 121   2   4   7   3  33 548  974   64   0]
 [ 471   3  33  55  18  14 418  476  324   3]
 [  5 914   0   0 139   0   3   1   0 815]]
Accuracy: 0.5053428317008014
```

d) Fully connected layer:



```
SNR ==> 0
557/557 [=====] - 1s 2ms/step - loss: 1.0916 - accuracy: 0.5366
[[ 72 2 20 194 14 53 649 135 609 3]
 [ 0 1390 0 0 0 0 0 0 0 403]
 [ 16 7 599 168 19 400 89 4 509 10]
 [ 7 2 4 1510 38 1 71 10 109 5]
 [ 0 15 0 8 1725 0 0 0 0 78]
 [ 5 2 61 31 10 1227 216 155 50 5]
 [ 22 1 14 48 10 136 805 564 223 3]
 [ 15 1 15 34 5 157 738 630 182 3]
 [ 35 1 58 222 8 37 480 130 758 3]
 [ 0 866 0 0 52 0 0 0 0 838]]
Accuracy: 0.5366209840485284
```

e) CNN:

```
SNR ==> 0
568/568 [=====] - 1s 2ms/step - loss: 0.7508 - categorical_accuracy: 0.6382
[[ 493 3 42 157 35 5 189 81 780 3]
 [ 0 1611 24 0 5 0 0 0 0 194]
 [ 9 21 1389 19 15 128 19 5 166 8]
 [ 85 2 6 1559 42 0 31 5 137 2]
 [ 11 11 5 33 1621 0 5 0 12 147]
 [ 4 4 89 1 2 1683 14 21 7 5]
 [ 109 4 11 5 7 13 517 1053 114 1]
 [ 71 3 6 8 6 16 489 1146 74 2]
 [ 369 4 52 154 34 5 148 73 970 2]
 [ 0 1079 19 0 55 0 0 0 0 608]]
Accuracy: 0.6381796169931763
```

f) LSTM:

```
SNR ==> 0
562/562 [=====] - 3s 5ms/step - loss: 0.8597 - categorical_accuracy: 0.6596
[[ 825 4 16 81 7 3 119 228 528 1]
 [ 0 1458 1 0 1 0 0 0 0 270]
 [ 4 2 1546 4 5 221 2 9 41 0]
 [ 119 0 1 1532 9 1 14 16 109 0]
 [ 3 3 0 10 1760 1 0 2 6 39]
 [ 3 2 71 1 3 1694 0 8 0 2]
 [ 211 3 7 5 2 12 104 1351 55 3]
 [ 199 0 8 3 1 10 97 1388 35 2]
 [ 685 3 27 89 6 4 107 165 752 0]
 [ 0 992 0 0 65 0 0 0 0 796]]
Accuracy: 0.6596372134431338
```

### 3. Integrals of the data



a) Logistic Regression:

```
SNR ==> 0
0.1485694976346024
[[ 167 485 9 15 155 104 499 94 28 264]
 [ 43 1283 0 1 35 51 32 23 14 255]
 [ 132 469 17 10 145 66 548 100 27 242]
 [ 198 494 4 12 142 119 452 43 39 286]
 [ 191 513 0 10 204 118 360 39 49 281]
 [ 99 531 10 3 179 35 563 118 14 194]
 [ 152 447 6 6 151 120 489 78 21 292]
 [ 141 438 7 6 157 112 511 99 21 282]
 [ 205 469 8 7 115 119 455 91 26 314]
 [ 42 1212 0 2 76 47 68 16 29 306]]
Accuracy: 0.1485694976346024
```

b) Decision Trees:

```
SNR ==> 0
0.2853621645846171
[[257 64 173 184 133 143 246 283 262 58]
 [ 16 833 11 17 109 19 15 23 18 770]
 [197 57 367 102 108 268 186 198 200 62]
 [254 61 135 321 181 111 226 185 229 84]
 [150 115 91 161 499 80 152 150 164 138]
 [ 81 26 225 50 58 848 173 164 91 25]
 [260 25 123 86 57 145 430 441 215 34]
 [237 23 103 91 40 151 450 485 174 21]
 [291 59 166 180 113 121 276 249 280 57]
 [ 41 705 14 45 175 29 31 27 20 774]]
Accuracy: 0.2853621645846171
```

c) Random forest:

```
SNR ==> 0
0.49115791845010137
[[ 452 3 51 81 60 31 422 434 277 9]
 [ 0 1093 1 0 46 0 0 0 0 597]
 [ 150 7 751 14 33 451 111 134 101 4]
 [ 284 5 17 817 105 9 183 166 192 11]
 [ 43 45 1 33 1457 0 21 27 19 119]
 [ 6 0 82 4 4 1490 51 103 6 0]
 [ 198 1 11 15 21 51 562 810 92 1]
 [ 163 1 10 14 20 56 473 956 80 1]
 [ 462 9 49 78 50 26 373 420 338 4]
 [ 2 832 0 1 152 1 2 1 2 805]]
Accuracy: 0.49115791845010137
```

d) Fully connected layer:

```
SNR ==> 0
563/563 [=====] - 1s 2ms/step - loss: 1.4207 - accuracy: 0.4286
[[ 88 13 200 572 34 115 197 382 203 13]
 [ 0 1198 0 0 0 0 0 0 0 542]
 [ 58 10 278 265 69 688 158 161 124 26]
 [ 36 3 28 1243 143 22 70 163 66 26]
 [ 0 3 0 24 1591 0 2 1 0 158]
 [ 12 17 134 98 53 1133 70 277 25 14]
 [ 37 4 135 329 24 186 99 925 78 12]
 [ 24 9 133 328 20 163 65 978 64 9]
 [ 72 8 171 569 41 82 195 388 237 8]
 [ 0 840 0 4 96 0 0 0 0 875]]
Accuracy: 0.42855556789163984
```

e) CNN:

```
SNR ==> 0
558/558 [=====] - 1s 1ms/step - loss: 1.5282 - categorical_accuracy: 0.3873
[[ 54  14 115 418 110 187 279 429 210 21]
 [ 0 982   6   0  77   0   0   0   0 687]
 [ 49  65 514 185  91 369 115 228 110 47]
 [ 40  29  25 981 163  30 142 212  76 43]
 [ 1  46  14 122 1333   9   5   4   0 250]
 [ 8  48 232 102  52 987  71 287  31 20]
 [ 30 11 125 306  97 188 223 645 143 14]
 [ 25 10 142 301  75 226 230 713 110 12]
 [ 54 11 117 482 104  94 253 402 252 23]
 [ 0 647   7   0 253   0   0   0   0 872]]
Accuracy: 0.3873444681089564
```

f) LSTM:

```
SNR ==> 0
563/563 [=====] - 1s 2ms/step - loss: 1.2711 - categorical_accuracy: 0.4575
[[ 56  12  90 391  60  46 205 447 399  7]
 [ 0 1405   0   0   0   1   0   0   0 420]
 [ 19  23 287  60  33 870  93 369  70 14]
 [ 28  3   5 1367  91   1 54  56 121 20]
 [ 1   4   3 50 1584   7   5   7   0 151]
 [ 8  43 175  19  27 1250  27 239  9 19]
 [ 18   2 121 215  37 105 138 896 263  9]
 [ 7   5 127 187  22 114 133 937 255  3]
 [ 40   9 86 415  51  49 210 437 438  8]
 [ 0 1051   0   5 80   0   0   0   0 771]]
Accuracy: 0.4575159766601834
```

## 4. Using a combination of raw data and the derivatives

a) Logistic Regression:

```
SNR ==> 0
0.1829152994827874
[[ 187   81 163 202 206 189 203 181 210 192]
 [ 11 1190   5  20  38   16   12   65   25 432]
 [ 181 108 149 200 217 195 160 202 196 198]
 [ 161 150 140 173 225 145 159 183 204 228]
 [ 134 289   83 126 249 114 140 168 134 343]
 [ 126 128 151 148 187 174 204 204 184 278]
 [ 164 122 167 127 214 163 215 254 160 223]
 [ 167 124 143 147 233 178 214 246 136 252]
 [ 175   97 162 185 224 185 178 178 206 182]
 [ 23 1079   13  19  40  13  25  51  39 506]]
Accuracy: 0.1829152994827874
```

b) Decision Trees:

```
SNR ==> 0
0.2849118513986986
[[ 315  60 157 199 128 123 269 265 235  63]
 [ 20 838  25 18 109 16 14 16 13 745]
 [212  51 376 119 103 269 215 201 192  68]
 [240  61 126 329 171 116 200 204 245  76]
 [[181 135  89 185 479  92 162 130 152 167]
 [ 86 20 241  47 51 836 187 194  88 34]
 [234 23  97  91 66 173 424 476 187 38]
 [260 21  93  66 46 146 447 541 193 21]
 [265 54 137 183 128 156 260 271 250 68]
 [ 26 726  29  35 155  24  20  33  25 735]]
Accuracy: 0.2849118513986986
```

c) Random forest:



```

SNR ==> 0
0.5138834789382895
[[ 458   4   31   46   35   31   426   467   258   5]
 [ 0 1107   0   0   52   0   0   0   0   631]
 [ 128   3  788   7  23  458  126  181  102   0]
 [ 270   3   7  911   64   3  166  159  184   11]
 [ 23   54   1  30 1540   0   16   11  10 148]
 [ 8   1   61   2   2 1588   73   94   6   1]
 [ 154   3   6   6   32  616  944   68   0]
 [ 110   1   7   2   10   31  605 1043   40   0]
 [ 460   3   28   48   33   22  410  401  365   2]
 [ 0 813   0   1 146   0   0   2   0 819]]
Accuracy: 0.5138834789382895

```

d) Fully connected layer:

```

SNR ==> 0
569/569 [=====] - 1s 2ms/step - loss: 1.1009 - accuracy: 0.5404
[[ 630   1   74  157   9   74  238  515   70   3]
 [ 0 1720   0   0   0   0   0   0   0 113]
 [ 277   5  732  137  23  507   34   35   80   3]
 [ 124   1  16 1563   41   5   24   33   7   4]
 [ 0   19   0   3 1752   0   0   0   0 84]
 [ 29   4  129  21   11 1235   66  314   3   2]
 [ 211   2  37  53   5 139 199 1115   20   3]
 [ 167   2  34  53   5 143 194 1219   12   1]
 [ 676   4  179 198   11  58 167 416 150   7]
 [ 0 1116   0   0  38   0   0   0   0 629]]
Accuracy: 0.540351841671248

```

e) CNN:

```

SNR ==> 0
566/566 [=====] - 1s 1ms/step - loss: 0.6989 - categorical_accuracy: 0.6634
[[ 918   3   18  120   35   4   83  113  543   3]
 [ 0 1305   54   0   3   0   0   0   0 427]
 [ 6   10 1454   3   12  141   11   2 136   5]
 [ 105   1   0 1566   29   0   4   5   73   1]
 [ 8   4   2  18 1592   0   1   2   1 115]
 [ 2   1  125   3   6 1729   2   9   2   2]
 [ 152   3   7   5   1 16 201 1325 107   3]
 [ 100   2   6   6   8 17 163 1425   70   0]
 [ 572   3  54  94  28   2   87   56  965   0]
 [ 0 885   23   0  46   0   0   0   0 840]]
Accuracy: 0.6634035728112383

```

f) LSTM:

```

SNR ==> 0
570/570 [=====] - 3s 5ms/step - loss: 0.8619 - categorical_accuracy: 0.6580
[[ 752   5   14   86   5   3  218  217  551   3]
 [ 0 1364   0   0   0   0   0   0   0 477]
 [ 14   3 1438   6   3 303   3   8  34   2]
 [ 105   0   0 1573  10   0  27  14  78   0]
 [ 2   0   4   5 1780   1   1   0   1 47]
 [ 1   1  68   3   5 1742   1  10   0   3]
 [ 143   2   3   9   2 12 215 1329  57   2]
 [ 126   1   5   3   5 14 198 1408  50   0]
 [ 661   5  21  65   7   3 170  157  760   3]
 [ 0 806   0   0  32   0   0   0   0 960]]
Accuracy: 0.657997256515775

```

## 5. Using a combination of raw data and the integrals

a) Logistic Regression:

```
SNR ==> 0
0.17543467322636552
[[ 232   91  174  278  193  139  160  179  166  193]
 [ 13 1212    6   13   27   19    5   66   32   420]
 [ 240  108  161  233  202  168  139  201  173  176]
 [ 225  143  135  219  199  148  130  187  124  237]
 [ 159  312  123  188  204  93   72  209  139  333]
 [ 181  149  163  220  180  139  150  221  159  269]
 [ 222  129  150  220  178  139  167  205  145  204]
 [ 218  134  182  223  191  134  163  189  138  212]
 [ 240  93  161  256  189  166  142  173  169  191]
 [ 15 1089    7   15   40   16    6   64   37  446]]
Accuracy: 0.17543467322636552
```

b) Decision Trees:

```
SNR ==> 0
0.2822720411472019
[[275  56 171 180 123 154 274 253 240  79]
 [ 20 791  24  35  93  17  17  13  10 793]
 [206  53 350 121  99 278 220 232 190  52]
 [222  59 129 321 165 102 222 205 237  85]
 [162 103 130 177 522  92 163 158 144 181]
 [ 87  22 233  47  36 896 156 223  98  33]
 [246  32 105  87  62 143 416 445 186  37]
 [250  28  91  80  47 154 419 487 197  31]
 [264  74 162 170 118 160 267 241 260  64]
 [ 23 681  23  34 142  26  30  24  21 731]]
Accuracy: 0.2822720411472019
```

c) Random forest:

```
SNR ==> 0
0.513490959662031
[[ 466   3   21   53   18   30   412   493   284    9]
 [  0 1120   0   0   51   0   0   0   0 657]
 [136   2  834   7   16  392  122  127  110   3]
 [289   4   8  872   72   5  215  181  197   8]
 [ 21   60   1  16 1569   0   9  14   9 137]
 [ 12   2   67   0   1 1601   71   86   2   0]
 [161   0   6   2   6  24  629  894   60   0]
 [103   1   6   3   3  25  588  999   37   1]
 [484   3   37   37   19   13  440  435  329   4]
 [  2 780   0   0 138   0   0   0   0 811]]
Accuracy: 0.513490959662031
```

d) Fully connected layer:

```
SNR ==> 0
560/560 [=====] - 1s 2ms/step - loss: 1.3056 - accuracy: 0.4732
[[ 16   2  34  398   9   45   53 1034  154   2]
 [  0 1330   0   0   0   0   0   0   0 513]
 [ 36   7  48  254   28  811   46  397  164   3]
 [  4   3  4 1687   59   5   4   64  13   6]
 [  0   9   0   3 1690   0   0   0   0 77]
 [  0   6   8  152   17 1150   20  414   3   5]
 [  0   2  13  251   6   54   2 1415   4   2]
 [  1   7  12  219   6   83   1 1419   0   1]
 [ 17   3  29  374   6   56   47 1139  143   4]
 [  0 740   0   0  79   0   0   0   0 991]]
Accuracy: 0.47317590576676155
```

e) CNN:



```
SNR ==> 0
566/566 [=====] - 1s 2ms/step - loss: 1.0092 - categorical_accuracy: 0.5701
[[ 298   4   62  172   38   24  178  534  504   6]
 [  0 1100  42   0    4   3   0   0   0 689]
 [ 44   18 1271   56   27  159   13   9 245   5]
 [ 69   4   16 1485   24    5   16   64 119   5]
 [ 1   22   0   26 1595   1    2   1   2 135]
 [ 55   4 193   14   18 1354   30 129   37   8]
 [111   3 11   61   26   36 185 1189 188   4]
 [ 87   2   8   56   21   54 155 1249 155   3]
 [151   6 99   170   29   21 124  384  775   3]
 [  0 675   38   1  70    7   0   0   0 1001]]
Accuracy: 0.5701254906296644
```

f) LSTM:

```
SNR ==> 0
564/564 [=====] - 2s 3ms/step - loss: 0.7058 - categorical_accuracy: 0.6708
[[ 969   5   6  228   2   1   8  93  512   1]
 [  0 1542   0   0   0   0   0   0   0 215]
 [  3  3 1536   4   5 234   0   4   2   0]
 [ 74   0   0 1697   0   0   3   4 44   0]
 [  0   0   0   3 1780   0   0   0   0 21]
 [  3   5 229   3   3 1542   1   1   0   1]
 [167   6   1   5   1   2 167 1270 129   0]
 [ 53   6   6   2   1   5 98 1600 76   1]
 [891   4   9 235   4   2   7 97 596   0]
 [  0 1085   0   0  48   0   0   0   0 659]]
Accuracy: 0.6708102108768036
```

## 6. Using a combination of the derivatives and the integrals

a) Logistic Regression:

```
SNR ==> 0
0.18185402117594052
[[ 205 111 179 219 182 146 186 184 238 200]
 [ 23 1179   9  12   21   20   17   57   31 445]
 [190 126 162 194 165 157 152 167 230 188]
 [191 148 147 205 151 144 179 166 203 222]
 [130 351 140 120 208  91 134 154 160 326]
 [184 152 144 167 115 180 145 198 219 236]
 [195 143 160 219 119 163 189 183 248 191]
 [169 148 156 180 120 159 165 198 226 206]
 [200 105 180 209 171 170 160 155 216 188]
 [ 19 1044  38  14  27  20  26  42  43 487]]
Accuracy: 0.18185402117594052
```

b) Decision Trees:

```
SNR ==> 0
0.28829691371930616
[[ 275  58 168 178 145 167 295 239 253  72]
 [ 15 782  16  19 119  14  24  17  17 791]
 [201  53 344 118  97 274 186 208 194  56]
 [241  66 118 348 131 119 214 224 235  60]
 [149 126  97 172 543 121 137 148 149 172]
 [ 82 24 196  53  35 866 170 193  86 35]
 [253  26  92 104  58 153 404 440 250  30]
 [256  11  68  65  46 139 394 550 177  21]
 [251  73 161 163 118 162 238 271 262  55]
 [ 28 655  26  28 164  24  32  25  33 745]]
Accuracy: 0.28829691371930616
```

c) Random forest:

```

SNR ==> 0
0.5107845328565043
[[ 473   4   33   56   28   21   409   481   265   5]
 [  0 1104   0   0   42   0   0   0   0 655]
 [ 147   0  849   11   23  439  141  112  106   3]
 [ 257   7   8  893   71   2  167  171  200   9]
 [ 21   52   0   26 1542   1   7  15   8 124]
 [  8   2   61   2   1 1537   69   96   6   1]
 [ 153   0   7   4   7  30  597  928   73   2]
 [ 100   2   12   8   5  24  579 1014   45   0]
 [ 466   1   39   49   20   13  441  418  336   4]
 [  0 809   0   2 140   0   0   1   0 796]]
Accuracy: 0.5107845328565043

```

#### d) Fully connected layer:

```

SNR ==> 0
567/567 [=====] - 1s 2ms/step - loss: 1.1622 - accuracy: 0.5232
[[ 309   2  108  199   9   93  186  541  382   0]
 [  0 1654   0   0   0   0   0   0   0 125]
 [ 100   7  587  183   24  495   28   54  307   3]
 [ 53   2   28 1535   40   5   25   53   57   2]
 [  0  15   0   2 1788   0   0   0   0 47]
 [ 21   3  130   57   13 1127   75  324   51   1]
 [ 110   3  39   67   4   89  179 1133  171   2]
 [ 102   3  35   57   7  130  191 1174  140   0]
 [ 263   5  180  207   6   72  128  451  512   2]
 [  0 1116   0   0  83   0   0   0   0 624]]
Accuracy: 0.5232423490488006

```

#### e) CNN:

```

SNR ==> 0
568/568 [=====] - 1s 1ms/step - loss: 0.9013 - categorical_accuracy: 0.6093
[[ 778   4   31  153   87   11  347  106  326   7]
 [  0 1536   51   0   1   0   0   0   0 161]
 [  9  12 1212   19   9  376   26   2 129  11]
 [ 122   2   9 1456   86   0   41   5 113   5]
 [ 20   7   6 1607   1   23   2   5 111]
 [  4   8 197   2   7 1550   12  23   8   1]
 [ 236   4  11   27   29   32  407  998   99   3]
 [ 187   3  13   14   19   35  303 1171   70   5]
 [ 469   8  57  169   59   21  283   97  713  10]
 [  0 1055   47   0  99   0   0   0   0 633]]
Accuracy: 0.609330248953514

```

#### f) LSTM:

```

SNR ==> 0
565/565 [=====] - 2s 3ms/step - loss: 0.8457 - categorical_accuracy: 0.6737
[[ 873   3   16   79   4   2  210  196  449   3]
 [  0 1568   0   0   1   0   0   0   0 171]
 [  9   1 1513   2   6  158   5   9  30   3]
 [ 69   0   1 1583   6   0  25  18  61   0]
 [  1   3   0   2 1754   0   0   1   0 50]
 [  5   1  67   1   3 1746   7   6   1   1]
 [ 129   1   4   6   4   6 190 1468   60   3]
 [ 118   3   4   3   5  11 147 1528   36   0]
 [ 686   1  24  86   3   0 158  150  656   1]
 [  0 1054   0   0  34   0   0   0   0 752]]
Accuracy: 0.6737011188656253

```

## 7. Using a combination of the raw data, the derivatives and the integrals

a) Logistic Regression:

```
SNR ==> 0
0.182251538163073
[[ 188   84   180   222   203   144   189   185   206   203]
 [ 11 1179   16   21   33   19    8   84   18 448]
 [ 181 113 158 182 238 169 181 203 195 207]
 [ 160 162 158 211 188 123 155 189 188 246]
 [ 117 337 123 149 232 89 104 162 137 356]
 [ 146 164 146 167 189 200 180 199 166 270]
 [ 188 132 148 201 183 146 185 222 164 217]
 [ 181 119 154 176 172 212 211 228 152 195]
 [ 201 111 170 208 175 165 179 178 196 201]
 [ 12 1058   22   18   55   15   14   56   29 511]]
Accuracy: 0.182251538163073
```

b) Decision Trees:

```
SNR ==> 0
0.28285571753228755
[[278 52 166 177 130 165 259 258 242 77]
 [16 831 17 18 100 17 14 26 24 774]
 [197 53 369 133 98 297 211 216 184 69]
 [257 58 128 316 172 93 208 235 234 79]
 [181 150 91 180 492 102 144 138 147 181]
 [84 16 190 54 44 938 169 200 103 29]
 [250 20 109 85 61 159 396 455 220 31]
 [240 16 106 65 42 171 430 512 190 28]
 [268 63 171 166 116 133 264 282 241 80]
 [21 688 23 37 174 32 34 25 26 730]]
Accuracy: 0.28285571753228755
```

c) Random forest:

```
SNR ==> 0
0.5084279132486796
[[ 459   4   33   48   21   24   478   463   271    3]
 [  0 1076   0   0   40   0   0   0   0   679]
 [ 134   5 815 13 20 427 114 122 119   4]
 [ 322   9   6 888 75   2 150 162 144   9]
 [ 15 54   0 24 1529   0 11   9   7 117]
 [  4   1 66   4   3 1528 66 92   7   0]
 [ 148   0   5 10   6 23 640 895 53   3]
 [ 111   2   5   4   8 30 600 991 35   0]
 [ 442   4 34 48 20 20 415 440 340   6]
 [  1 860   0   0 140   0   0   0   0 783]]
Accuracy: 0.5084279132486796
```

d) Fully connected layer:

```
SNR ==> 0
560/560 [=====] - 1s 2ms/step - loss: 1.1267 - accuracy: 0.5261
[[ 67   1 58 212   9 62 65 862 507   1]
 [  0 1679   0   0   0   0   0   0   0 91]
 [ 35 10 640 180 24 460 18 94 355   2]
 [ 21   5   6 1645 37   1   6 53 76   3]
 [  0 20   0   2 1675   0   0   0   0 64]
 [  6   2 149 34   9 1121 18 439 23   3]
 [ 27   0 38 81   6 91 18 1365 159   1]
 [ 16   2 32 55   5 121 25 1347 107   1]
 [ 60   1 87 248   6 45 27 624 677   4]
 [  0 1163   0   1 60   0   0   0   0 546]]
Accuracy: 0.5260952168082254
```

e) CNN:



```
SNR ==> 0
564/564 [=====] - 1s 1ms/step - loss: 0.7692 - categorical_accuracy: 0.6316
[[ 894    4   59  162   10    8  108  200   348    0]
 [  0 1529    0    0    0    0    0    0    0 261]
 [ 43    2 1588   22   12   80    8   19   50    4]
 [ 205    1   12 1453   20    0    8   26   59    2]
 [ 10   14    4  40 1659    0    0    1    1 111]
 [  1    6   115    3    4 1605    5   37    5    0]
 [ 150    3   18   10    6   21 239 1252 101    1]
 [ 128    1    6    8    7   30 216 1327   62    2]
 [ 942    0   58  139    9    9   81 175 412    1]
 [  0 1069    0    2   50    0    0    0    0 689]]
Accuracy: 0.6315818645382996
```

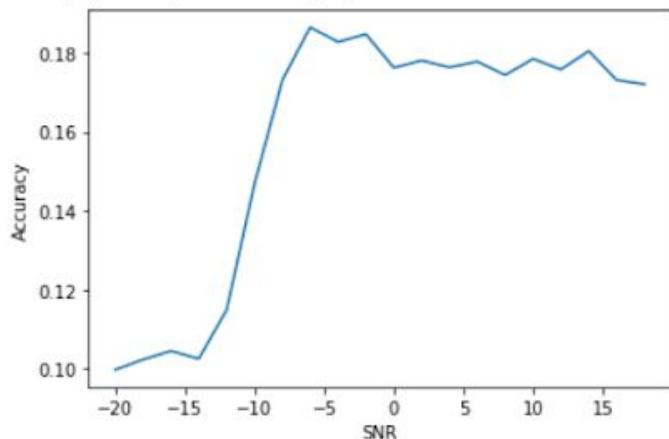
f) LSTM:

```
SNR ==> 0
561/561 [=====] - 3s 6ms/step - loss: 0.8901 - categorical_accuracy: 0.6736
[[ 932    2   11   66    4    6  301  151   323    0]
 [  0 1705    0    0    0    0    0    0    0 153]
 [ 14    1 1576    3    5 191    6    2   22    0]
 [ 90    0    0 1610    6    0   24   17   53    0]
 [  7    2    1    6 1693    0    2    0    3 35]
 [ 1    2   46    2    4 1714    1   16    1    1]
 [ 116    1    4    8    3   10 263 1401   22    1]
 [ 89    2    6    5    0   12 229 1408   15    1]
 [ 794    1   19   77    1    2 243 111 476    1]
 [  0 1065    0    0   34    0    0    0    0 702]]
Accuracy: 0.6735627056264987
```

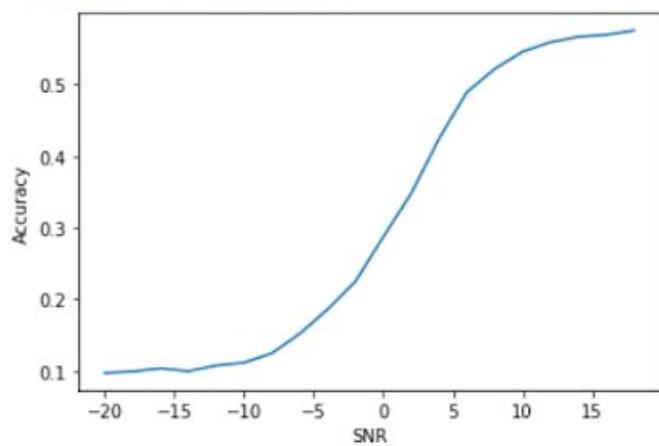
## Accuracy against SNR plots

### 1. Raw data

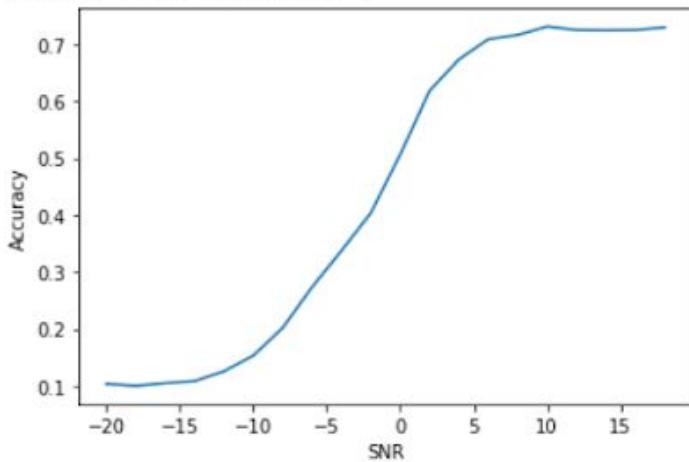
a) Logistic Regression:



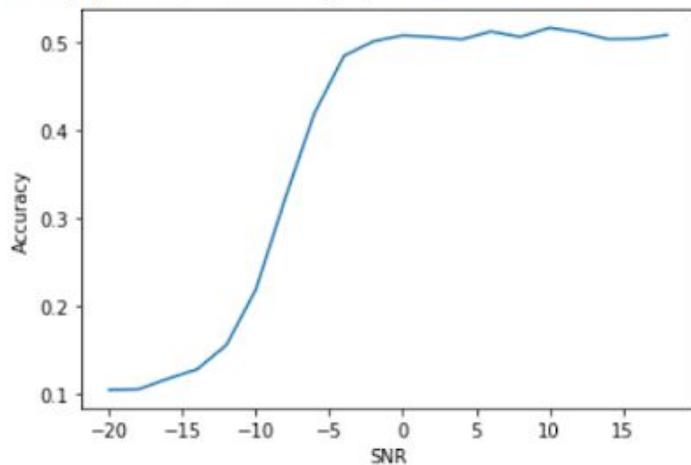
b) Decision Trees:



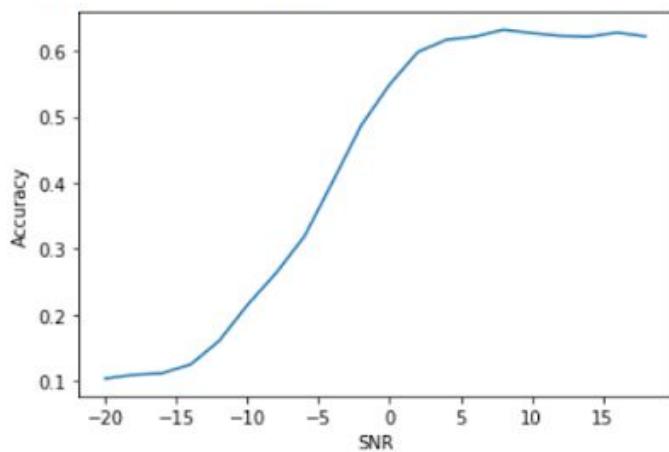
c) Random forest:



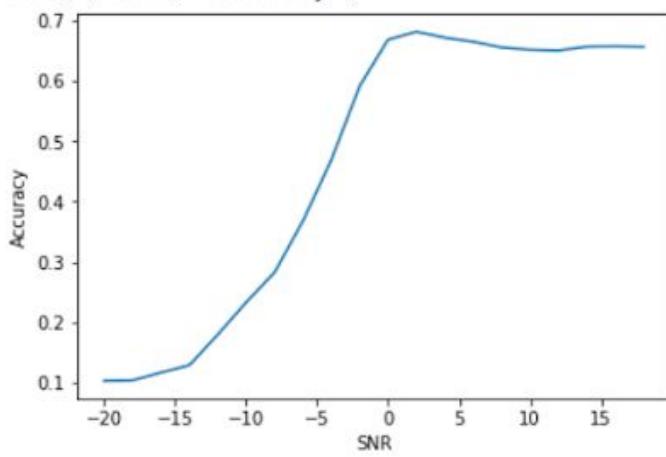
d) Fully connected layer:



e) CNN:

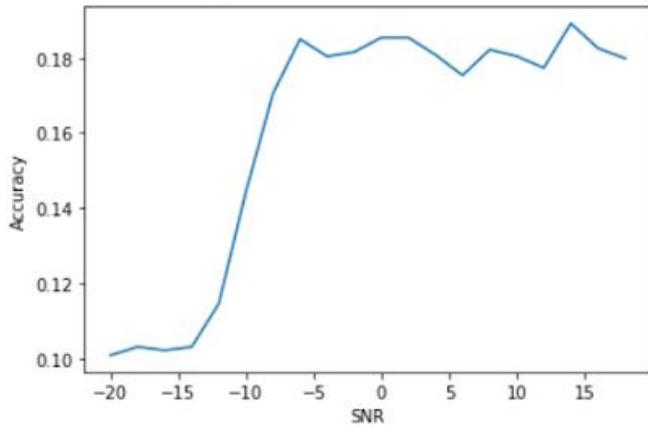


f) LSTM:

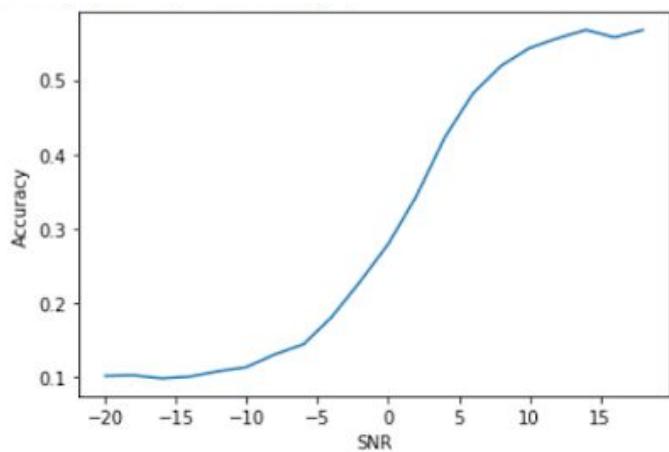


## 2. Derivatives of the data

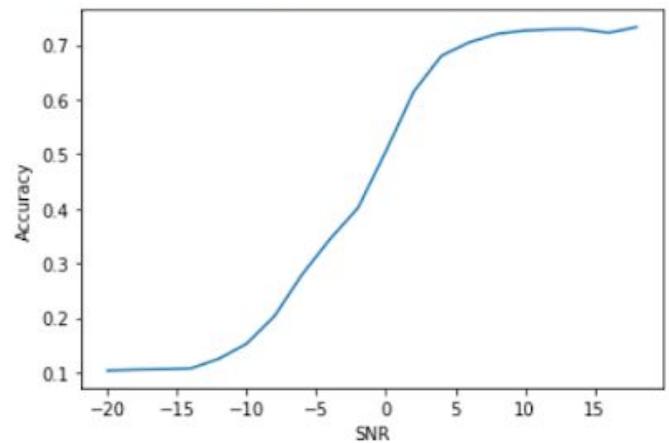
a) Logistic Regression:



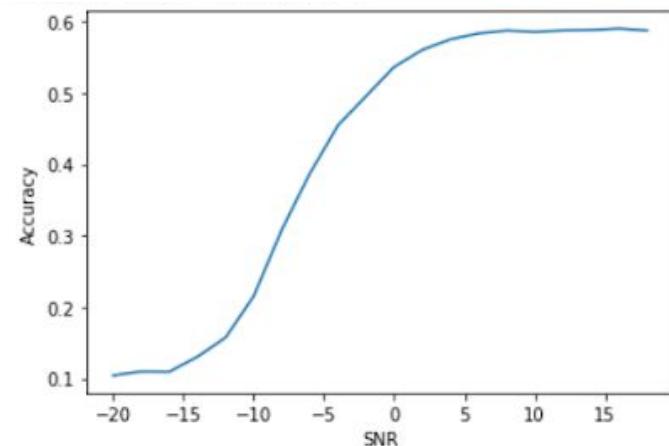
b) Decision Trees:



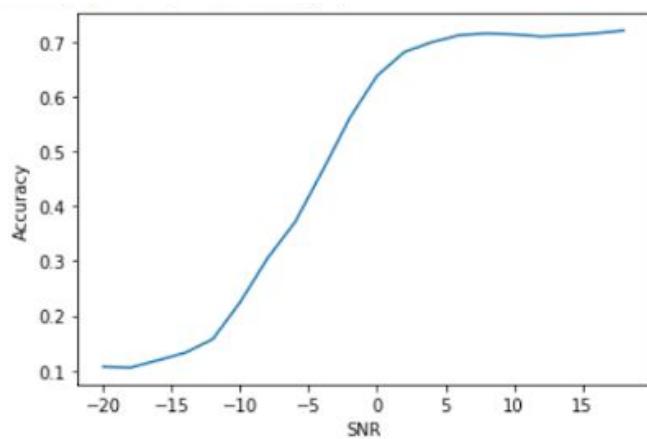
c) Random forest:



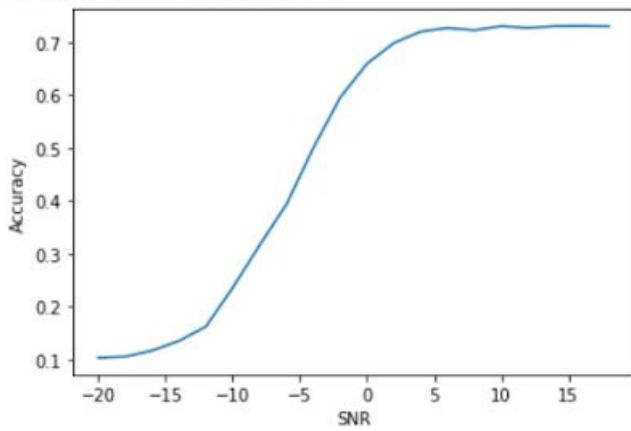
d) Fully connected layer:



e) CNN:

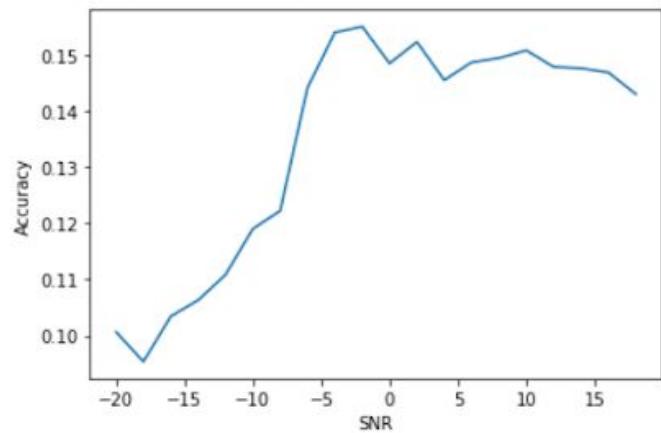


f) LSTM:

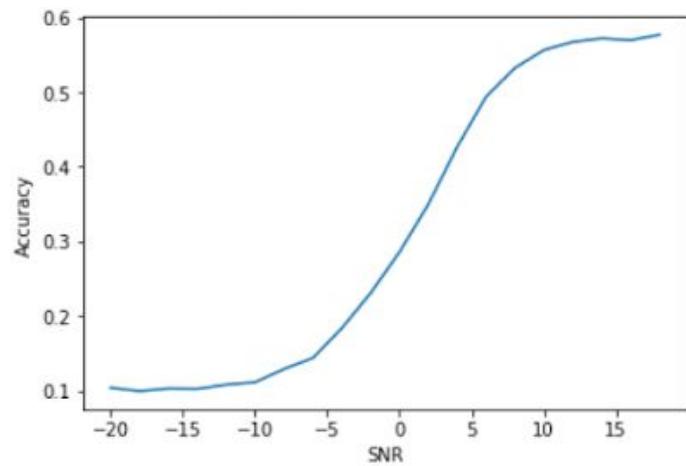


### 3. Integrals of the data

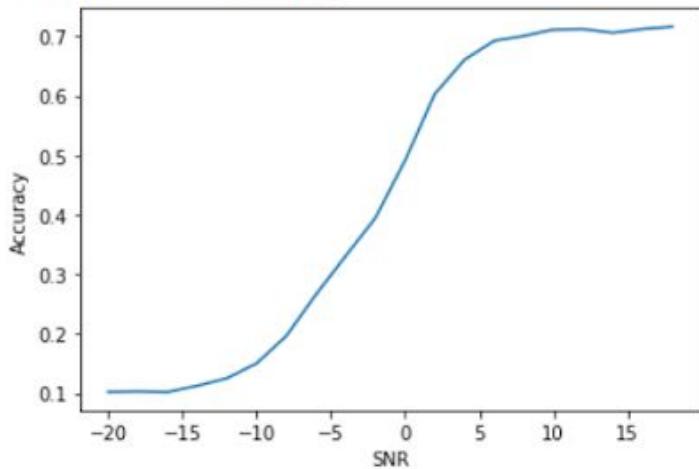
a) Logistic Regression:



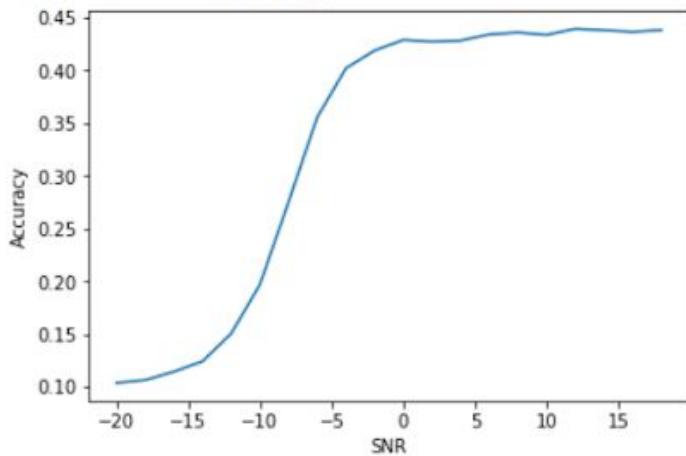
b) Decision Trees:



c) Random forest:

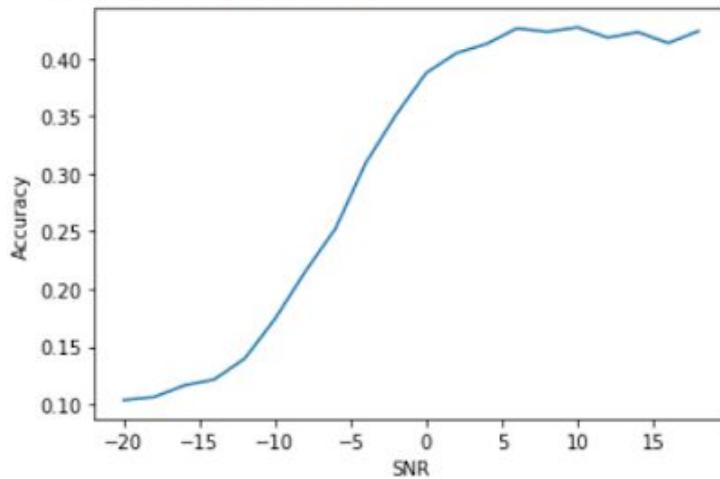


d) Fully connected layer:

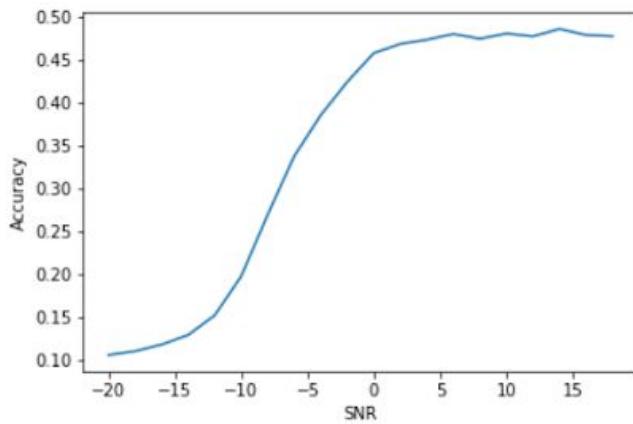


e) CNN:

Logistic Regression Decision Tree LSTM

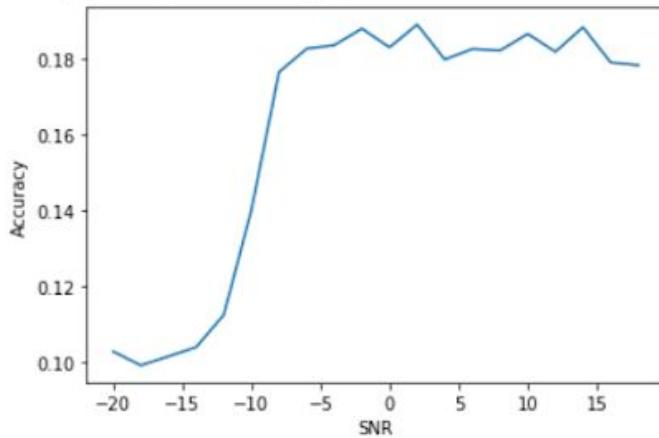


f) LSTM:

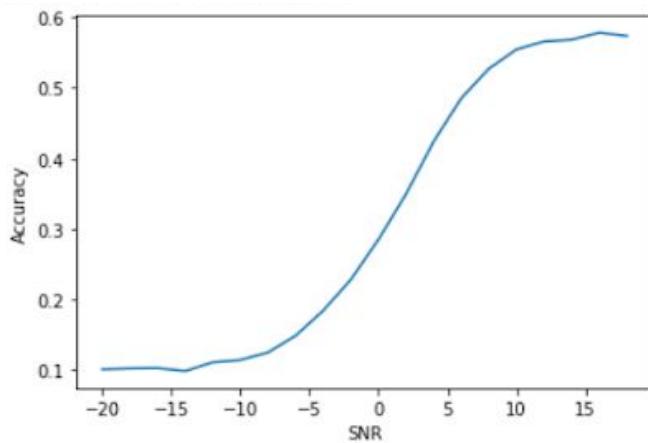


#### 4. Using a combination of raw data and the derivatives

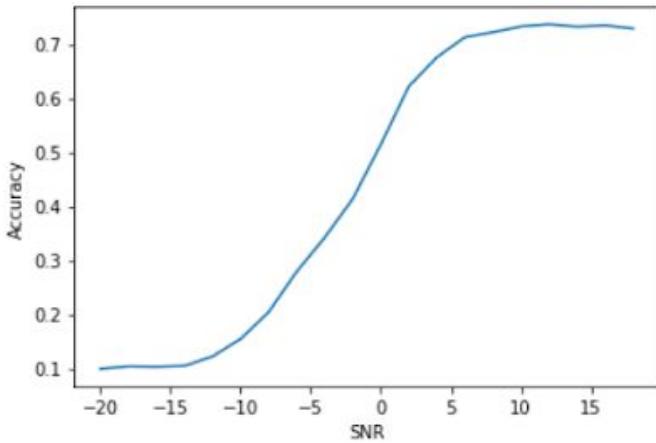
a) Logistic Regression:



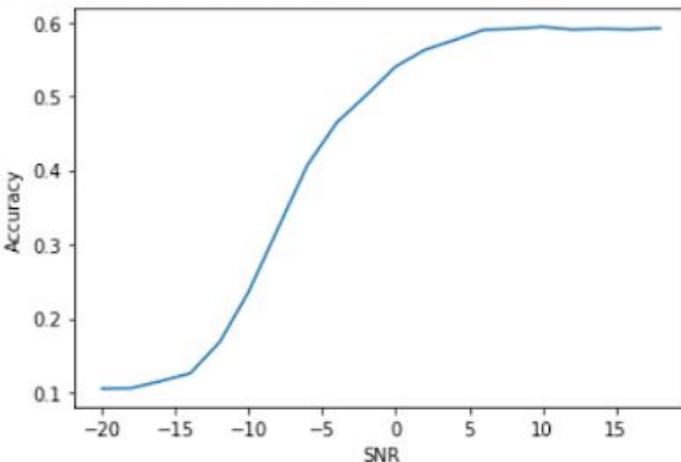
b) Decision Trees:



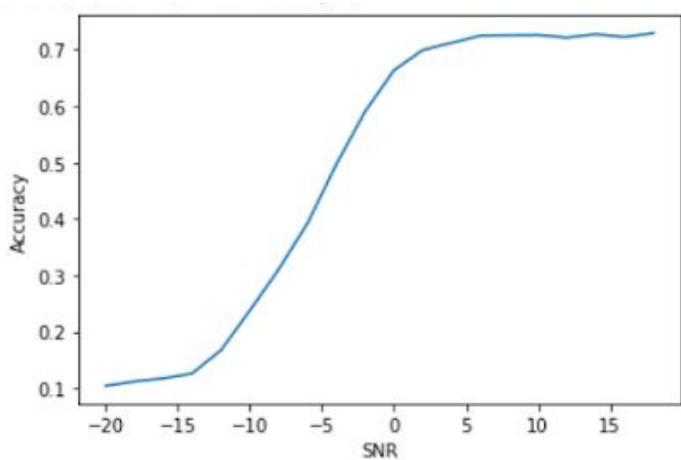
c) Random forest:



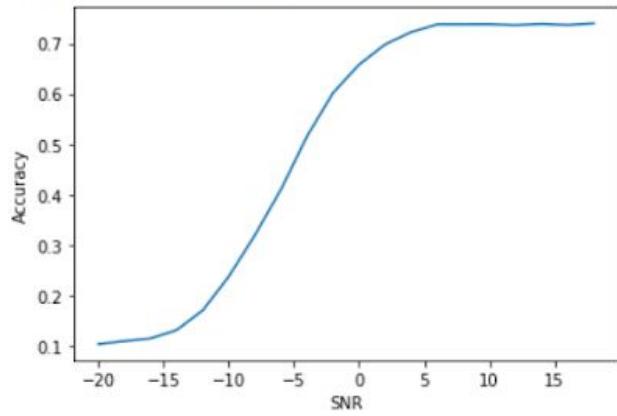
d) Fully connected layer:



e) CNN:

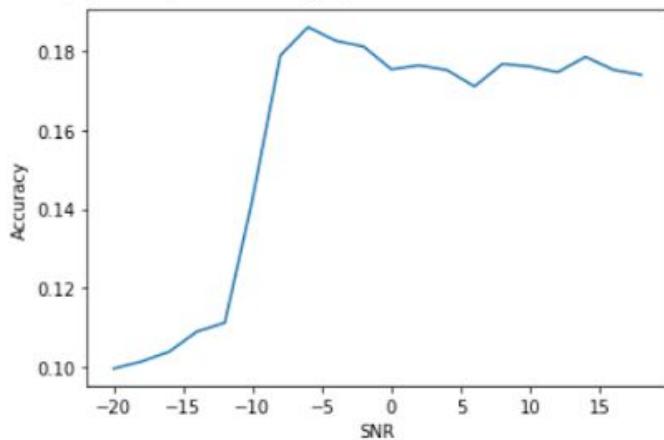


f) LSTM:

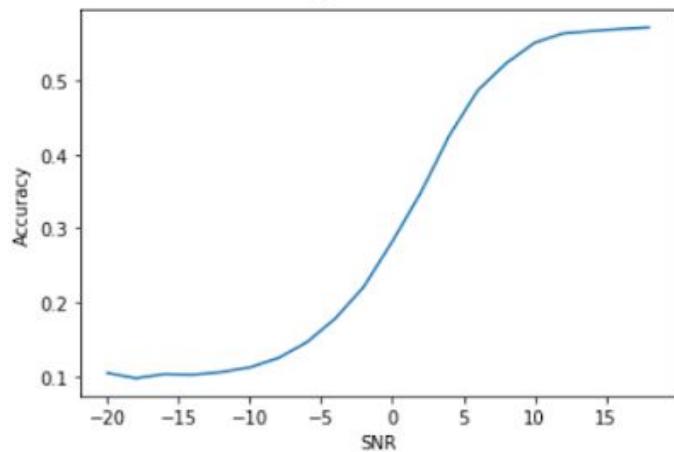


## 5. Using a combination of raw data and the integrals

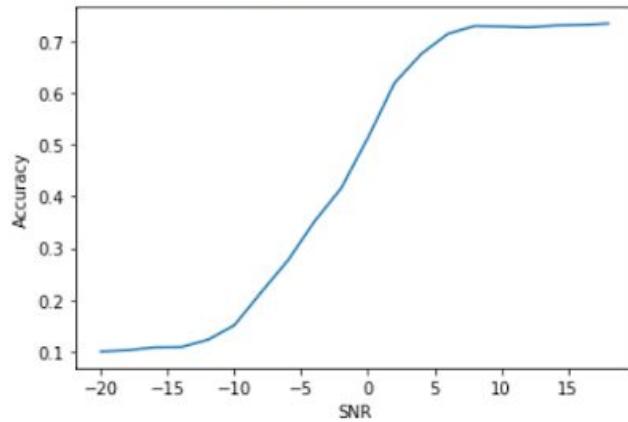
a) Logistic Regression:



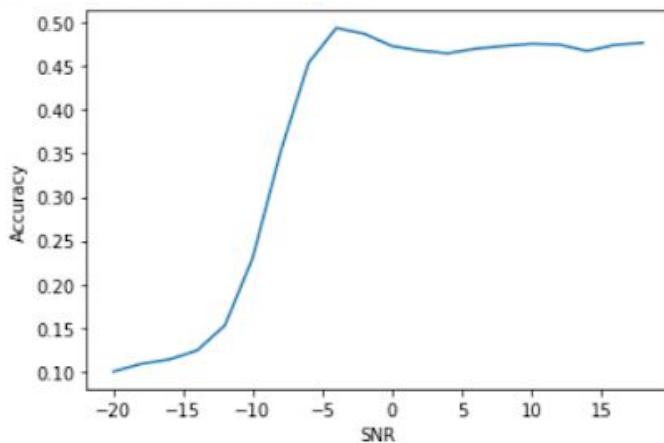
b) Decision Trees:



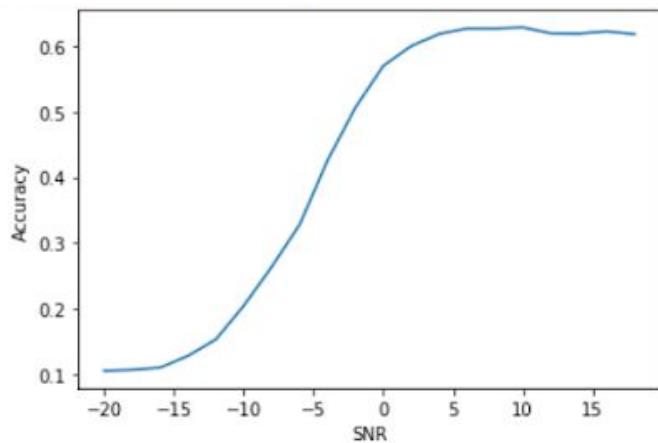
c) Random forest:



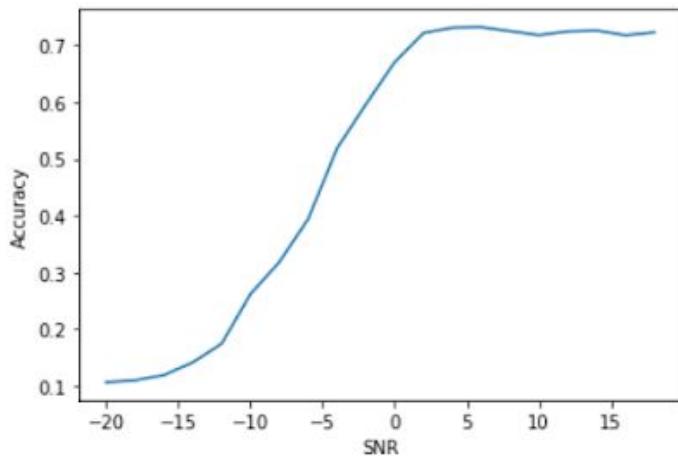
d) Fully connected layer:



e) CNN:

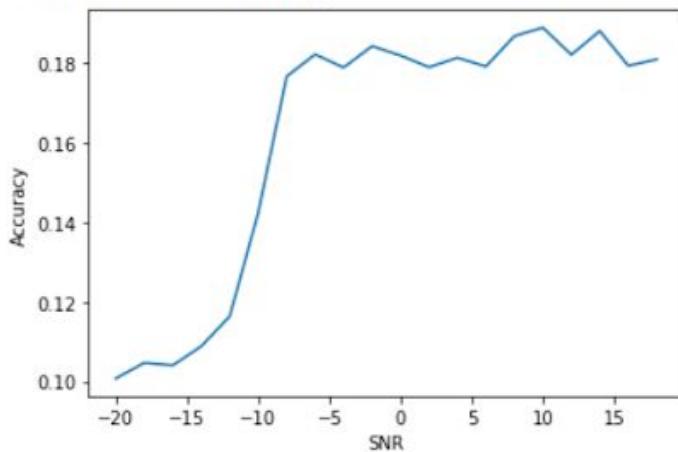


f) LSTM:

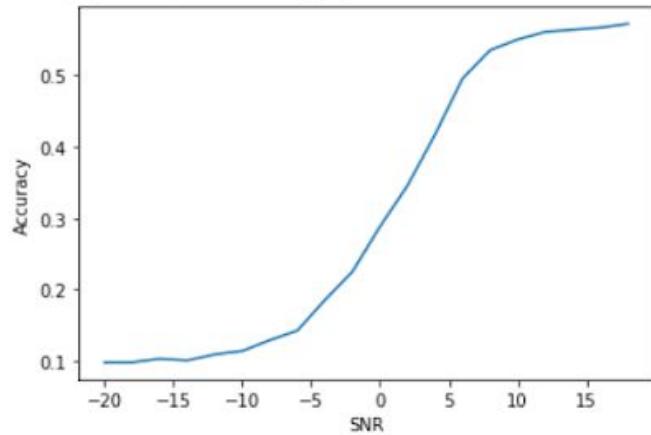


## 6. Using a combination of the derivatives and the integrals

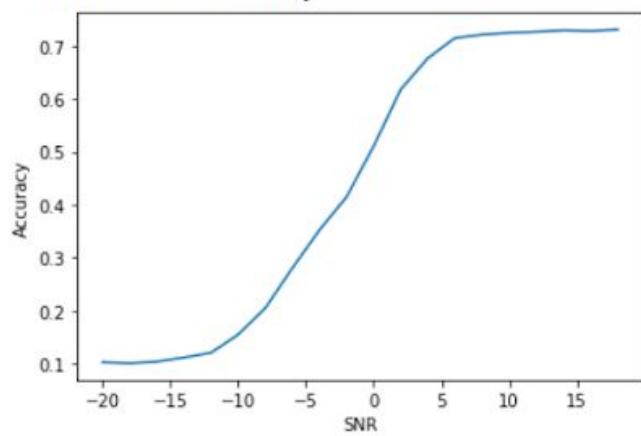
a) Logistic Regression:



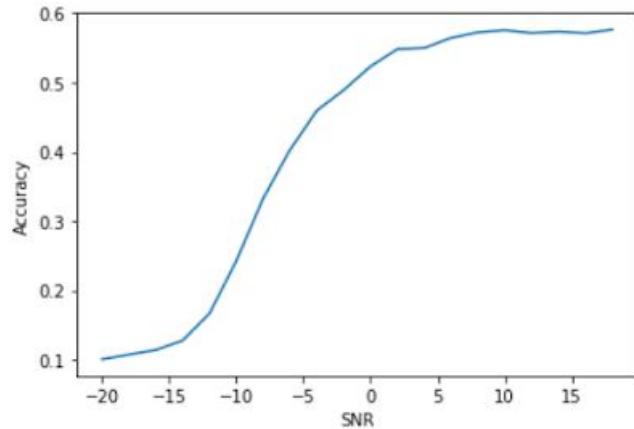
b) Decision Trees:



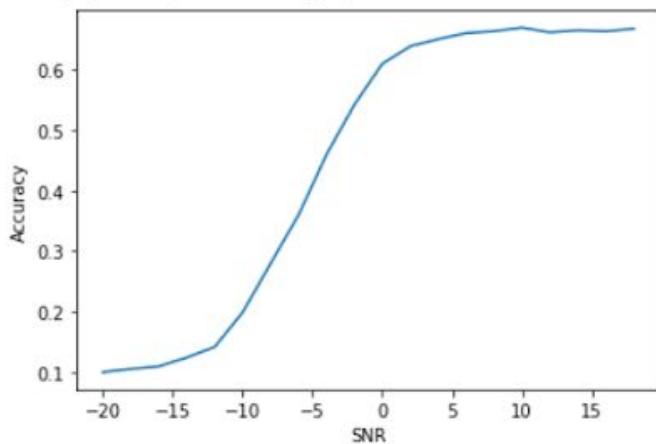
c) Random forest:



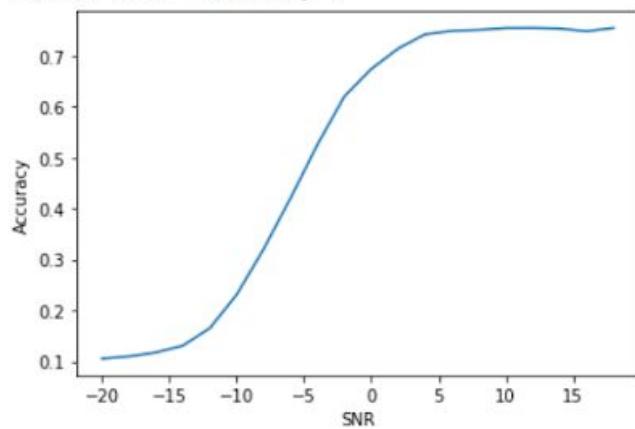
d) Fully connected layer:



e) CNN:

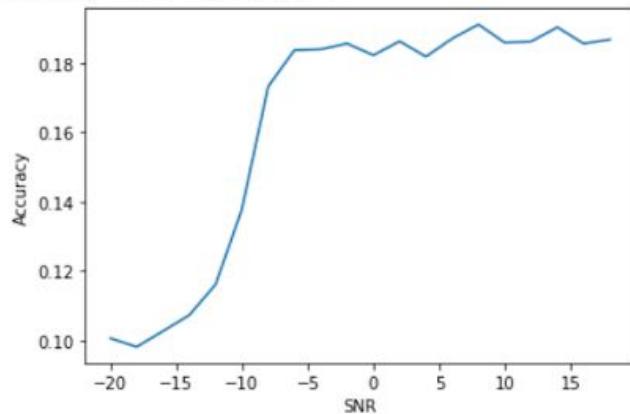


f) LSTM:

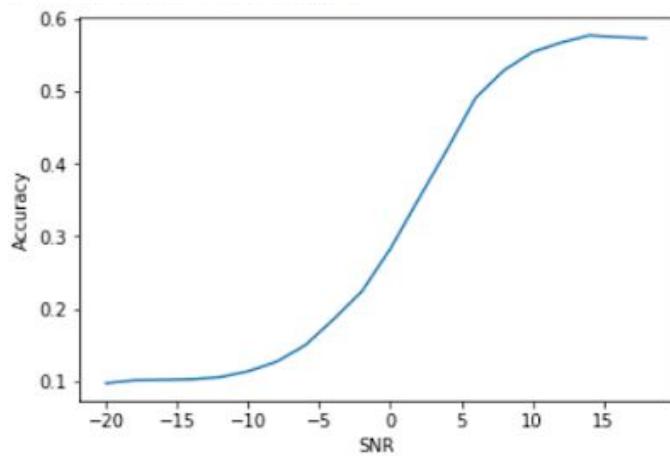


## 7. Using a combination of the raw data, the derivatives and the integrals

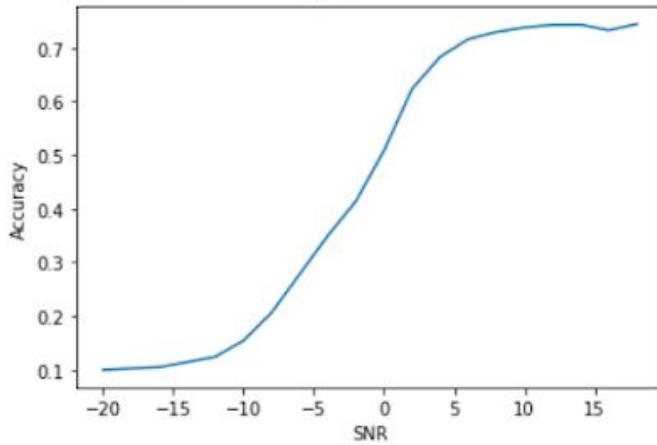
a) Logistic Regression:



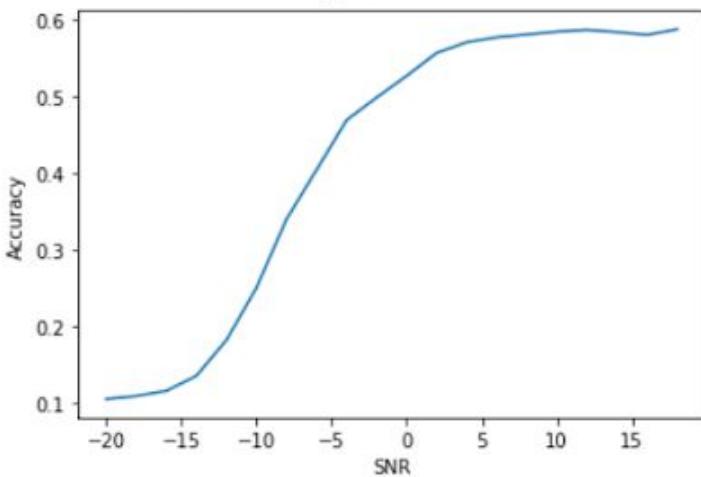
b) Decision Trees:



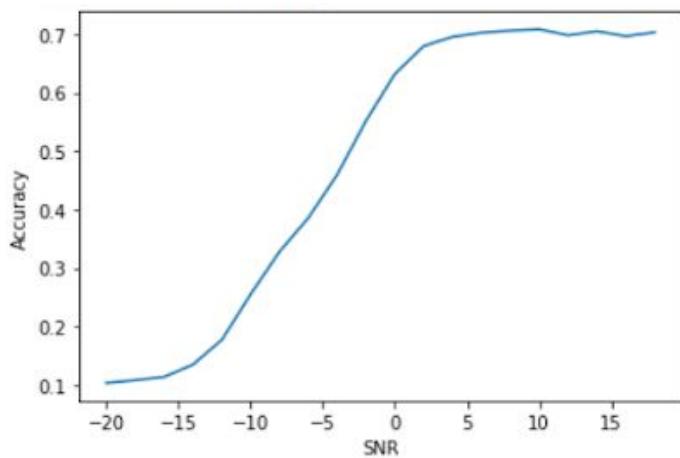
c) Random forest:



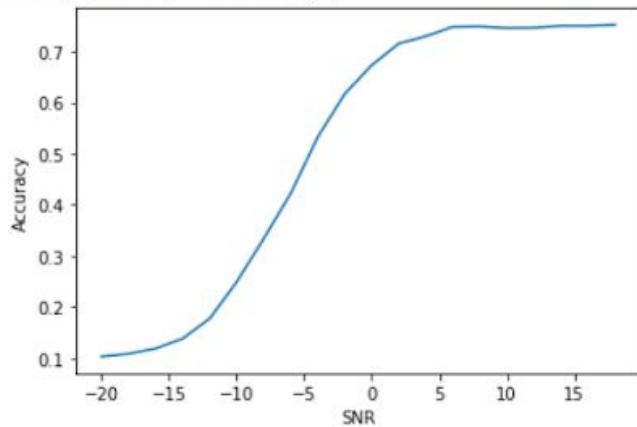
d) Fully connected layer:



e) CNN:



f) LSTM:



### Conclusion:

Almost all models' plots show that the classes with negative SNR are the most confusing.