Importance Estimate of Features via analysis of their Weight and Gradient profile

Ho Tung Jeremy Chan[1,*,+] **and Eduardo Veas**[2,+]

[1]Interactive System and Data Science, Technical University of Graz, Graz, 8010, Austria
[2]Human AI Interaction, Know Center, Graz, 8010, Austria
[*]jchan@know-center.at
[+]these authors contributed equally to this work

ABSTRACT

Understanding what is important and redundant within data can improve the modelling process of neural networks by reducing unnecessary model complexity, training time and memory storage. This information is however not always priorly available nor trivial to obtain from neural networks. There are existing feature selection methods which utilise the internal working of a neural network for selection, however further analysis and interpretation of the input features' significance is often limiting. We propose an approach that offers an extension that estimates the significance of features by analysing the gradient descent of a pairwise layer within a model. The changes that occur with the weights and gradients throughout training provide a profile that can be used to better understand the importance hierarchy between the features for ranking and feature selection. Additionally, this method is transferable to existing fully or partially trained models, which is beneficial for understanding existing or active models. The proposed approach is demonstrated empirically with a study which uses benchmark datasets from libraries such as MNIST and scikit-feat, as well as a simulated dataset and an applied real world dataset. This is verified with the ground truth where available, and if not, via a comparison of existing statistical based and deep learning based feature selection methods through the methodology of Reduce and Retrain.

1 Supplementary Material: Appendix

1.1 Model Architecture

The baseline architecture used in **E1: Simulated Dataset** is as follows,

- Linear(10, 160)
- ReLU
- Linear(160, 44)
- ReLU
- Linear(44,25)
- Sigmoid
- Linear(25, 166)
- ReLU
- Linear(166, 2)
- Softmax

The baseline architecture used in **E2: scikit-feature** is as follows,
For the *BASEHOCK* dataset:

- Linear(4862, 151)
- Linear(151, 184)
- Sigmoid

- Linear(184,168)
- Linear(168, 2)
- Softmax

For the *PCMAC* dataset:

- Linear(3289, 58)
- ReLU
- Linear(58,97)
- ReLU
- Linear(97,101)
- Linear(101,4)
- linear(4, 2)
- Softmax

For the *RELATHE* dataset:

- Linear(4322, 99)
- Tanh
- Linear(99,113)
- ReLU
- Linear(113,66)
- ReLU
- linear(66,2)
- Softmax

The baseline architecture used in **E3: MNIST** is as follows,

- Linear(784,800)
- Linear(800, 10)
- Softmax

The baseline architecture used in **E4: Dry Bean** is as follows,

- Linear(16,49)
- Dropout(0.00682)
- Linear(49,39)
- Dropout(0.208)
- Linear(39,7)
- LogSoftmax

BER	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.1193	0.1202	0.1687	0.1851	0.2041	0.205	0.2222	0.2423	0.2938	0.3506
Fisher	0.1193	0.1199	0.1213	0.1412	0.1487	0.1536	0.1541	0.2315	0.2361	**0.2991**
FScore	0.1193	0.1208	0.1208	0.1406	0.147	0.1552	0.1535	0.2331	**0.2334**	0.2994
Weight-Naive	0.1199	0.1203	0.1231	0.1416	0.227	0.2667	0.2654	0.389	0.359	0.4455
DF	0.1667	0.1228	0.135	0.1413	0.1408	**0.1412**	**0.1414**	0.1753	0.2548	0.4075
NFS	0.1455	0.1829	0.204	0.2084	0.2264	0.2259	0.2513	0.3786	0.4143	0.509
Grad-AUC	0.1173	0.121	0.1216	0.1212	0.1205	0.1776	0.224	0.2846	0.3466	0.3535
Grad-ROC	0.1173	0.119	0.121	0.1217	**0.12**	0.1897	0.2071	0.2853	0.3424	0.3759
Grad-STD	0.1173	**0.1176**	**0.118**	**0.1199**	0.124	0.1415	0.1418	**0.1706**	0.239	0.2992
F1	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.8776	0.8769	0.828	0.812	0.7923	0.7914	0.7734	0.7491	0.6841	0.6226
Fisher	0.8776	0.8774	0.8761	0.8541	0.8473	0.8418	0.8412	0.7619	0.7583	0.6967
FScore	0.8776	0.8766	0.876	0.8548	0.8486	0.8403	0.8425	0.7606	**0.7607**	**0.6968**
Weight-Naive	0.8772	0.877	0.8743	0.8546	0.767	0.6957	0.7072	0.5668	0.5304	0.4472
DF	0.8173	0.8748	0.8617	0.8541	0.855	**0.8546**	**0.8543**	0.8209	0.7372	0.5839
NFS	0.8508	0.8143	0.7932	0.789	0.7717	0.7721	0.7348	0.6163	0.5555	0.4819
Grad-AUC	0.8794	0.8756	0.8757	0.8761	0.8772	0.8173	0.7682	0.7084	0.6134	0.564
Grad-ROC	0.8794	0.8781	0.8763	0.8757	**0.8775**	0.8042	0.7871	0.7058	0.6289	0.5569
Grad-STD	0.8794	**0.8792**	**0.8787**	**0.8774**	0.8734	0.8533	0.8531	**0.8246**	0.7559	**0.6968**

Table 1. Reduce and retrain results of the *simul_study* dataset. The results show that *Fisher*, *FScore* and *Grad-STD* can all consistently outperform the 2 baselines, *Weight-Naive* and *RS*.

1.2 Results: Tables and Figures

BER and F1 results of reduce and retrain are available in tables 1, 2, 3, 4, 5, and 6. Figures of reduce and retrain are available in figures 1, 2, 3 and 4.

For E1: Simulated Dataset, the results are available in table 1 and figure 1. The results show that *Fisher*, *FScore* and *Grad-STD* can all consistently outperform the 2 baselines, *Weight-Naive* and *RS*. And *Grad-STD* is amongst the best at retaining performance during *Reduce and Retrain*.

For E2: scikit-feature, the results are available in table 2, 3, 4, and figure 2. The results of *BASEHOCK* show that *Grad* methods can consistently outperform the baseline, *RS*, and perform on par with the baseline, *Weight-Naive*. On the other hand, the results of both *PCMAC* and *RELATHE* show that *Grad* methods can consistently outperform both baselines, *RS* and *Weight-Naive*. And all 3 *Grad* methods are amongst the overall best at retaining performance during *Reduce and Retrain* for the datasets.

For E3: MNIST, the results are available in table 5 and figure 3. The results show that *Grad* methods can consistently outperform both baselines, *Weight-Naive* and *RS*. And, again, all 3 *Grad* methods are amongst the best at retaining consistent performance during *Reduce and Retrain*.

For E4: Dry Bean Dataset, the results are available in table 6 and figure 4. The results show that *DF* and *Grad* methods can consistently outperform both baselines, *Weight-Naive* and *RS*. And, again, all 3 *Grad* methods are amongst the best at retaining consistent performance during *Reduce and Retrain*.

Figures of importance estimate visualisation are available in figures 5 and 6.

Figures demonstrating the minimum difference in the results of partial training are available in figures 7 and 8.

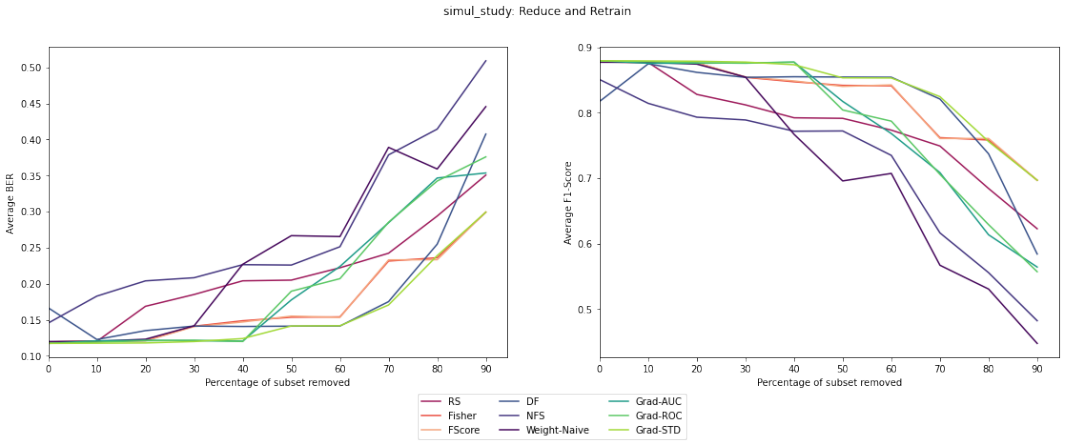

Figure 1. Reduce and retrain plot of the simulated dataset from E1: simul_study. The results show that *Fisher*, *FScore* and *Grad-STD* consistently outperform or perform on par with both *RS* and *Weight-Naive* as the baseline measures.

BER	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.0123	0.0149	0.0173	0.0207	0.0252	0.032	0.0431	0.0606	0.0921	0.1542
Fisher	0.0123	0.012	0.0135	0.0135	0.0155	0.0174	0.0215	0.0205	0.0246	0.0248
FScore	0.0123	0.0128	0.0125	0.0132	0.0152	0.0182	0.0214	0.0205	0.0248	0.0246
Weight-Naive	0.0123	0.013	**0.0113**	0.013	**0.0115**	**0.011**	**0.0115**	**0.014**	0.0178	0.041
DF	0.0433	0.0128	0.017	0.0235	0.0262	0.0359	0.0452	0.0642	0.1	0.1663
NFS	0.0145	0.0128	0.0157	0.0193	0.0233	0.0288	0.0432	0.0601	0.1003	0.1662
Grad-AUC	0.012	**0.0116**	0.0123	0.0126	0.013	0.0128	0.0158	0.0152	**0.0158**	**0.0178**
Grad-ROC	0.012	0.0123	0.0125	0.0125	0.014	0.0128	0.014	0.0148	0.0175	0.0198
Grad-STD	0.012	0.0121	0.0116	**0.0118**	0.0141	0.0163	0.0155	0.018	0.0195	0.0224
F1	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.9877	0.985	0.9826	0.9792	0.9747	0.9679	0.9567	0.9392	0.9076	0.8452
Fisher	0.9877	0.9879	0.9864	0.9864	0.9844	0.9824	0.9784	0.9794	0.9754	0.9751
FScore	0.9877	0.9872	0.9874	0.9867	0.9847	0.9817	0.9784	0.9794	0.9751	0.9754
Weight-Naive	0.9877	0.9869	**0.9887**	0.9869	**0.9884**	**0.9889**	**0.9884**	**0.9859**	0.9822	0.9585
DF	0.9565	0.9872	0.9829	0.9764	0.9736	0.9641	0.9548	0.9357	0.8997	0.8331
NFS	0.9854	0.9872	0.9842	0.9807	0.9766	0.9708	0.9565	0.9397	0.8992	0.8334
Grad-AUC	0.9879	**0.9884**	0.9877	0.9874	0.9869	0.9872	0.9842	0.9847	**0.9842**	**0.9822**
Grad-ROC	0.9879	0.9877	0.9874	0.9874	0.9859	0.9872	0.9859	0.9852	0.9824	0.9802
Grad-STD	0.9879	0.9879	0.9884	**0.9882**	0.9859	0.9837	0.9844	0.9819	0.9804	0.9774

Table 2. Reduce and retrain results of the *BASEHOCK* dataset. The results show that *Grad* methods can consistently outperform the baseline, *RS*, and perform on par with the baseline, *Weight-Naive*.

BER	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.0788	0.0915	0.1015	0.1073	0.1195	0.1281	0.1472	0.1753	0.2047	0.2606
Fisher	0.0788	0.0796	0.0791	**0.0752**	0.0793	0.0819	0.0816	0.0879	0.0969	0.1031
FScore	0.0788	**0.0764**	0.0793	0.0755	0.0783	0.0828	**0.081**	0.0873	0.0971	0.1008
Weight-Naive	0.0789	0.0802	0.0815	0.0772	**0.0734**	0.079	0.0814	0.0898	0.1021	0.1216
DF	0.1309	0.0832	0.0887	0.0927	0.0976	0.114	0.1259	0.1631	0.1919	0.2403
NFS	0.0949	0.0835	0.0919	0.1035	0.1195	0.1317	0.1491	0.1835	0.2266	0.2848
Grad-AUC	0.0802	0.0796	0.0798	0.0781	0.0835	0.0815	0.0829	**0.0851**	0.0852	**0.0952**
Grad-ROC	0.0802	0.0792	**0.0765**	0.0794	0.079	**0.0811**	0.0826	0.0866	**0.0836**	0.1057
Grad-STD	0.0802	0.0816	0.0814	0.081	0.083	0.0835	0.0891	0.094	0.0973	0.1086
F1	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.9209	0.9081	0.8981	0.8922	0.8801	0.8714	0.8523	0.8244	0.7946	0.7379
Fisher	0.9209	0.9204	0.9209	**0.9247**	0.9206	0.9169	0.9183	0.9115	0.9028	0.8966
FScore	0.9209	**0.9235**	0.9206	0.9245	0.9216	0.917	**0.9188**	0.9123	0.9026	0.899
Weight-Naive	0.9209	0.9193	0.918	0.9224	**0.9263**	0.9206	0.9183	0.91	0.8977	0.8778
DF	0.8688	0.9165	0.9108	0.9069	0.9018	0.8855	0.8734	0.8366	0.8077	0.7589
NFS	0.9046	0.9159	0.9074	0.8961	0.8798	0.868	0.8499	0.8156	0.7726	0.7143
Grad-AUC	0.9196	0.9201	0.9198	0.9216	0.9162	0.9183	0.9167	**0.9147**	0.9147	**0.9046**
Grad-ROC	0.9196	0.9204	**0.9232**	0.9204	0.9206	**0.9188**	0.9173	0.9131	**0.9162**	0.8941
Grad-STD	0.9196	0.918	0.9183	0.9188	0.9167	0.9162	0.9108	0.9059	0.9023	0.8909

Table 3. Reduce and retrain BER results o the *PCMAC* dataset. The results show that *Grad* methods can consistently outperform both baselines, *RS*, and *Weight-Naive*.

BER	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.0862	0.0889	0.0914	0.0999	0.1064	0.117	0.1309	0.1521	0.1893	0.2499
Fisher	0.0862	0.087	0.0838	0.0871	**0.0833**	0.0868	0.0917	0.1019	0.1169	0.1404
FScore	0.0862	0.0854	0.0831	0.0891	0.0845	**0.0853**	0.0925	0.1018	0.1152	0.1474
Weight-Naive	0.0837	0.0863	0.083	0.087	0.0846	0.0913	0.1009	0.1098	0.1201	0.1623
DF	0.1146	0.0878	0.0866	0.0989	0.1045	0.1104	0.1235	0.1486	0.1673	0.2272
NFS	0.0861	0.0902	0.0908	0.0947	0.1052	0.1137	0.1201	0.1351	0.1733	0.2379
Grad-AUC	0.087	0.0866	0.0853	0.0858	0.0878	0.0888	**0.0885**	0.0896	0.1014	0.1124
Grad-ROC	0.087	**0.0831**	0.0867	**0.0842**	0.0865	0.0875	0.0892	**0.0894**	**0.0954**	**0.1108**
Grad-STD	0.087	0.0848	**0.0826**	0.0868	0.0904	0.0924	0.0924	0.1	0.1045	0.1225
F1	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.9134	0.9111	0.908	0.8995	0.8933	0.8819	0.8683	0.8467	0.8098	0.7496
Fisher	0.9134	0.913	0.9159	0.9126	**0.9166**	0.9126	0.9077	0.8978	0.8831	0.8595
FScore	0.9134	0.9144	0.9161	0.9109	0.9151	**0.914**	0.907	0.8978	0.8845	0.8524
Weight-Naive	0.9158	0.9134	0.9166	0.9118	0.9143	0.9081	0.8982	0.8888	0.8781	0.838
DF	0.885	0.9124	0.9133	0.9005	0.8949	0.8892	0.8752	0.8505	0.8328	0.7728
NFS	0.9137	0.9094	0.9091	0.9057	0.8951	0.8861	0.8802	0.8649	0.8262	0.7617
Grad-AUC	0.9123	0.9127	0.914	0.9141	0.9112	0.9113	**0.9118**	0.9107	0.8991	0.8875
Grad-ROC	0.9123	**0.9169**	0.913	**0.9147**	0.9134	0.9119	0.9109	**0.911**	**0.905**	**0.8898**
Grad-STD	0.9123	0.9148	**0.9176**	0.9131	0.9099	0.9066	0.9069	0.8991	0.8954	0.8772

Table 4. Reduce and retrain results of the *RELATHE* dataset. The results show that *Grad* methods can consistently outperform both baselines, *RS* and *Weight-Naive*.

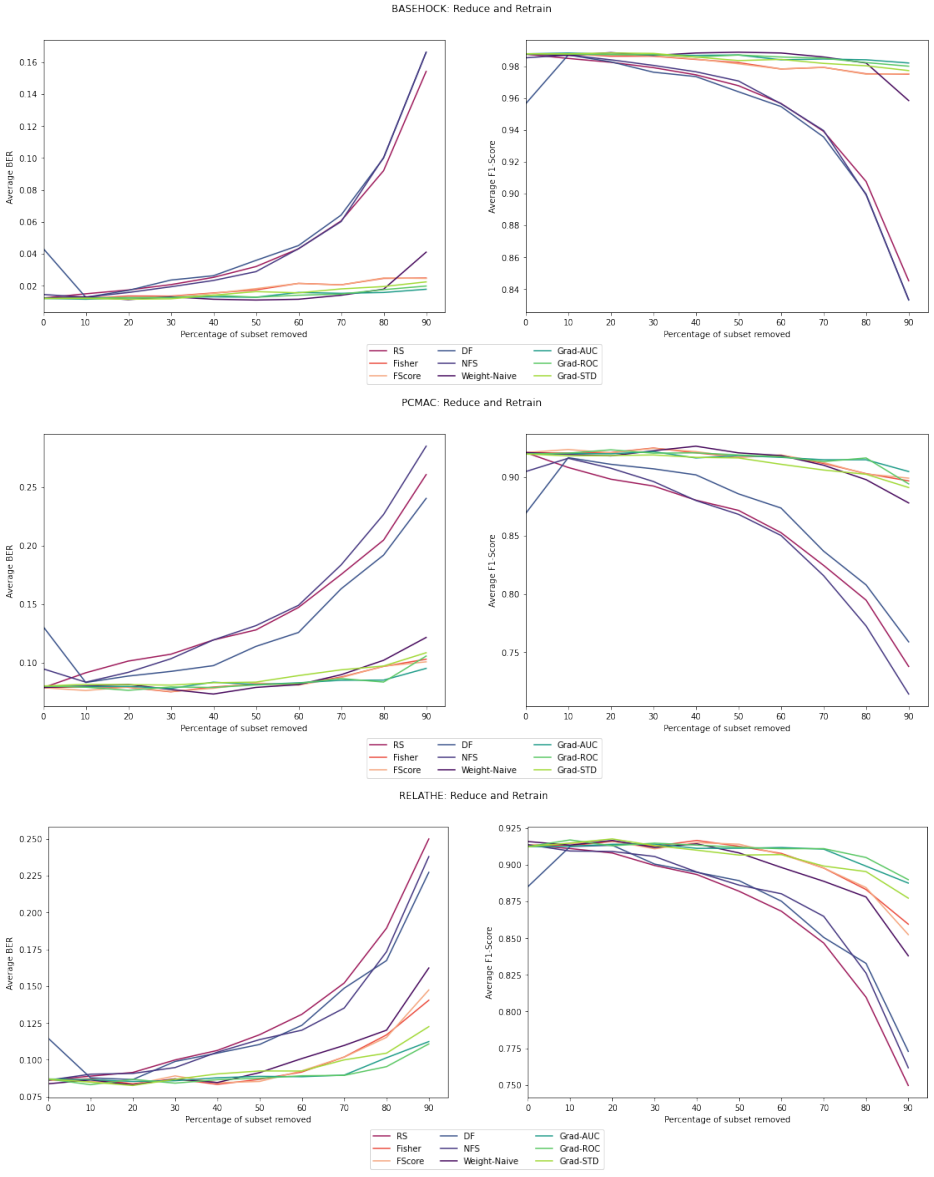

Figure 2. Reduce and retrain of the datasets, *BASEHOCK*, *PCMAC*, and *RELATHE* from E2: scikit-feature. The results of the binary datasets show that *Grad* methods can consistently outperform or perform on par with all other methods.

BER	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.0228	0.0239	0.0251	0.0259	0.028	0.0303	0.0362	0.0443	0.061	0.1084
Fisher	0.0228	**0.0225**	0.0237	0.0242	**0.0239**	0.0246	0.0259	0.0318	0.0511	0.1153
FScore	0.0228	0.0239	0.0241	**0.0238**	**0.0239**	0.0255	0.0292	0.0447	0.0899	0.3141
Weight-Naive	0.0238	0.027	0.0374	0.049	0.0572	0.0983	0.1538	0.2638	0.3829	0.4182
DF	0.1019	0.0247	0.0256	0.0269	0.0258	0.0272	0.0287	0.033	**0.0377**	**0.0544**
NFS	0.023	0.0248	0.025	0.0264	0.0281	0.0324	0.0352	0.0458	0.0644	0.12
Grad-AUC	0.0233	0.0236	**0.0231**	**0.0238**	0.0245	0.0258	0.0265	0.0312	0.0415	0.0839
Grad-ROC	0.0233	0.024	0.0243	0.0242	0.0242	0.0248	0.0259	**0.0297**	0.042	0.0766
Grad-STD	0.0233	0.0235	0.0244	0.0241	0.0248	**0.0241**	**0.0255**	0.03	0.0424	0.0851

F1	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.9772	0.9761	0.9749	0.974	0.972	0.9697	0.9638	0.9556	0.9389	0.8915
Fisher	0.9772	**0.9775**	0.9763	0.9757	**0.9761**	0.9753	0.9741	0.9681	0.9489	0.8846
FScore	0.9772	0.9761	0.9758	**0.9762**	**0.9761**	0.9744	0.9708	0.9552	0.91	0.6829
Weight-Naive	0.9762	0.9729	0.9625	0.9509	0.9426	0.9013	0.8452	0.7306	0.5475	0.3697
DF	0.8963	0.9753	0.9743	0.9731	0.9741	0.9728	0.9713	0.9669	**0.9622**	**0.9455**
NFS	0.9769	0.9751	0.9749	0.9736	0.9718	0.9676	0.9647	0.9542	0.9354	0.8796
Grad-AUC	0.9766	0.9763	**0.9768**	0.9761	0.9754	0.9742	0.9735	0.9688	0.9585	0.916
Grad-ROC	0.9766	0.9759	0.9756	0.9758	0.9758	0.9752	0.9741	**0.9703**	0.958	0.9232
Grad-STD	0.9766	0.9765	0.9755	0.9759	0.9751	**0.9759**	**0.9745**	0.9699	0.9576	0.9148

Table 5. Reduce and retrain results of the *MNIST* dataset. The results show that *DF*, *Grad-AUC*, *Grad-ROC* and *Grad-STD* are able to consistently outperform both *RS* and *Weight-Naive* as the baseline measures with only a slight difference in the 90% removal.

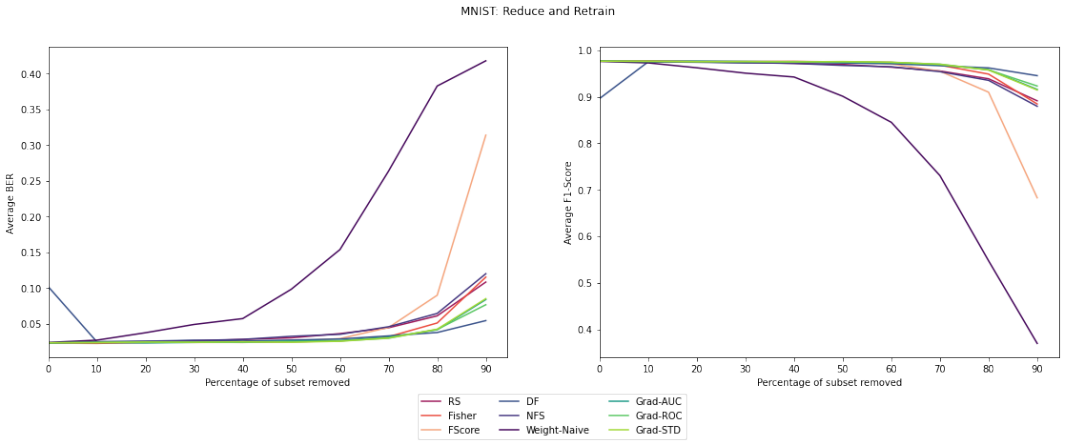

Figure 3. Reduce and retrain of the dataset from E3: MNIST. The results show that *Grad* methods can consistently outperform both baselines, *Weight-Naive* and *RS*.

BER	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.0922	0.0929	0.0937	0.0958	0.0982	0.1034	0.1048	0.1282	0.1759	0.4114
Fisher	0.0922	0.0912	0.0957	0.1099	0.111	0.1111	0.1129	0.1537	0.4113	0.5128
FScore	0.0922	**0.0898**	0.0961	0.1106	0.1102	0.1111	0.1123	0.1678	0.4201	0.5125
Weight-Naive	0.0881	0.0912	0.1013	0.1104	0.111	0.1119	0.1191	0.1283	0.3482	0.4004
DF	0.1382	0.0939	0.098	0.1047	0.1064	0.1095	0.1095	0.1094	0.1133	**0.1164**
NFS	0.0949	0.0941	0.0942	**0.0922**	**0.0977**	**0.0964**	**0.1007**	**0.104**	0.2049	0.3157
Grad-AUC	0.0912	0.0936	0.1102	0.1106	0.1095	0.1097	0.1107	0.1115	0.1171	0.1166
Grad-ROC	0.0912	0.0952	0.1022	0.1041	0.1103	0.1093	0.1096	0.1112	**0.1163**	0.1169
Grad-STD	0.0912	0.0908	**0.0891**	0.0932	0.1007	0.1034	0.1092	0.1119	0.1171	0.1166
F1	Removal									
Methods	Baseline	10%	20%	30%	40%	50%	60%	70%	80%	90%
RS	0.8954	0.8949	0.8937	0.891	0.8869	0.876	0.8722	0.8471	0.8071	0.5986
Fisher	0.8954	0.8964	0.8907	0.8751	0.8748	0.8703	0.8652	0.75	0.5446	0.5236
FScore	0.8954	**0.8988**	0.8897	0.8743	0.8755	0.8717	0.8675	0.7371	0.5406	0.524
Weight-Naive	0.9004	0.8966	0.8842	0.8754	0.8746	0.8729	0.8599	0.8191	0.5949	0.5851
DF	0.8258	0.8935	0.8898	0.8814	0.8799	0.8769	0.8774	0.8772	0.8707	0.8649
NFS	0.8918	0.893	0.8937	**0.8958**	**0.8887**	**0.8906**	**0.8835**	**0.8801**	0.7627	0.6889
Grad-AUC	0.8966	0.8931	0.8749	0.875	0.8763	0.8766	0.8755	0.8745	0.8653	0.866
Grad-ROC	0.8966	0.8919	0.8833	0.8817	0.8755	0.8766	0.8769	0.8746	**0.8668**	**0.8658**
Grad-STD	0.8966	0.8979	**0.9003**	0.8948	0.8864	0.8838	0.8761	0.8713	0.8648	0.8644

Table 6. Reduce and retrain results of the *DryBean* dataset. The results show that *DF* and *Grad* methods can consistently outperform both baselines, *Weight-Naive* and *RS*.

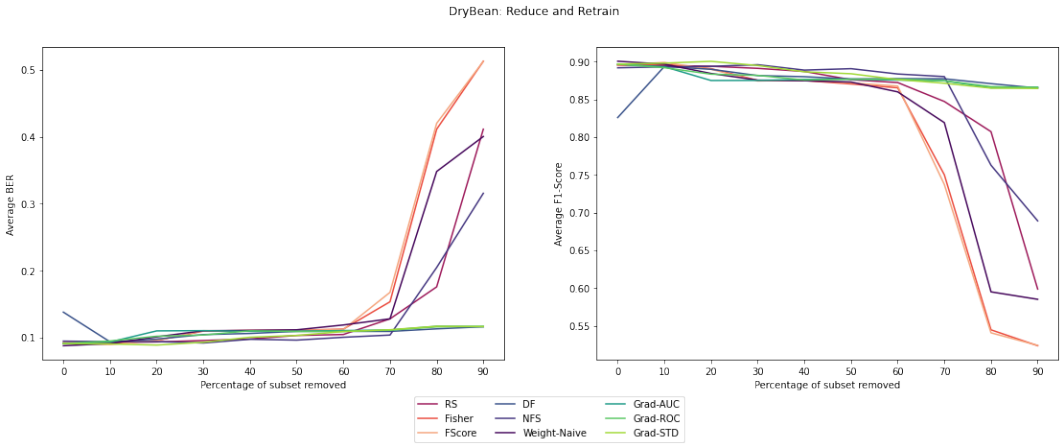

Figure 4. Reduce and retrain of the dataset from E4: Dry Bean Dataset. The results show that *DF*, *Grad-AUC*, *Grad-ROC*, and *Grad-STD*, are able to consistently outperform both *RS* and *Weight-Naive* as the baseline measures.

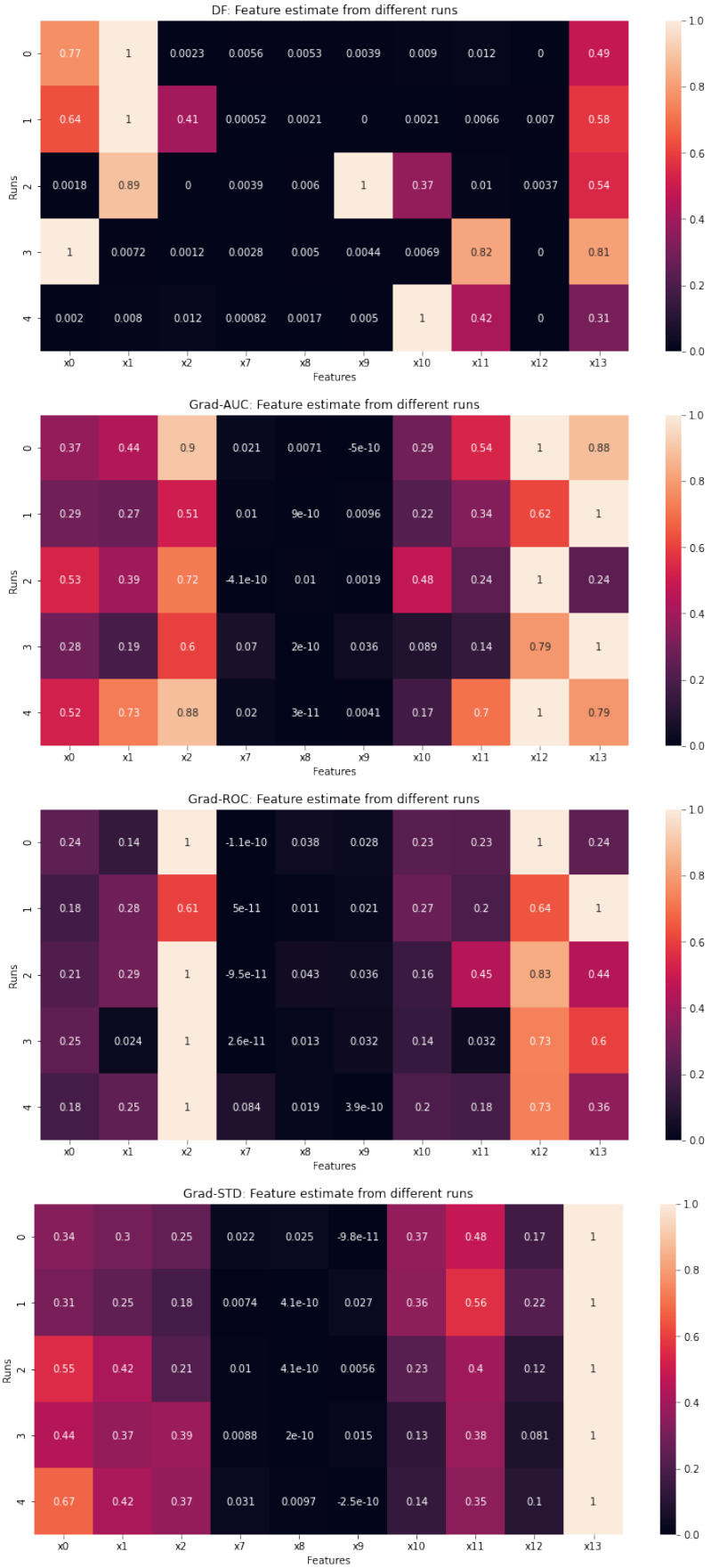

Figure 5. Visualisation of feature importance from E1: simul_study. Based on the setup of the simulation, variables, x_0, x_1, x_2, should have a higher importance, and variables, x_7, x_8, x_9, should have the least importance. This pattern is evident in all runs of *Grad-AUC*, *Grad-ROC* and *Grad-STD*, however it is only present in portions of the runs of *DF*.

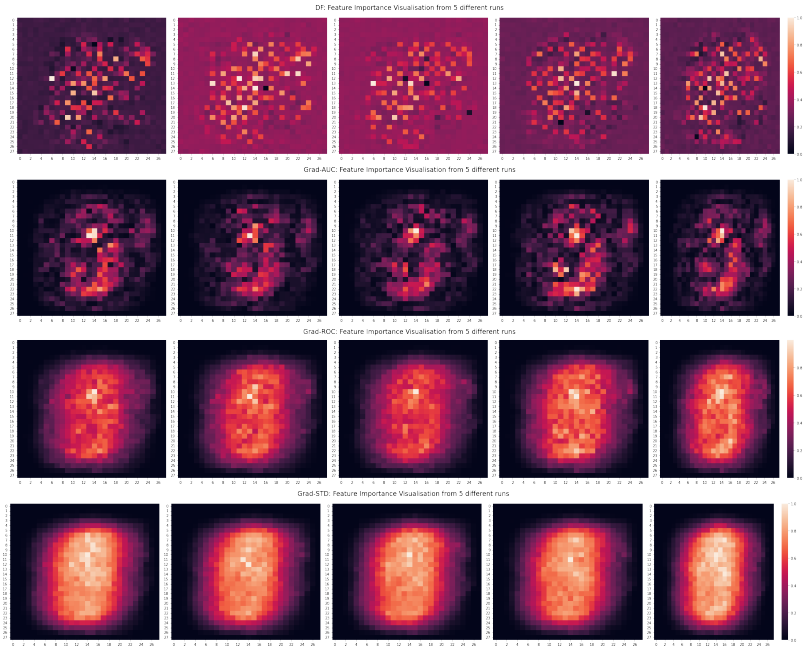

Figure 6. Visualisation of feature importance from E3: MNIST. The centre of the image where the digit lies is consistently highlighted for importance for *Grad-AUC*, *Grad-ROC* and *Grad-STD*. However, the centre is only sporadically highlighted for *DF*.

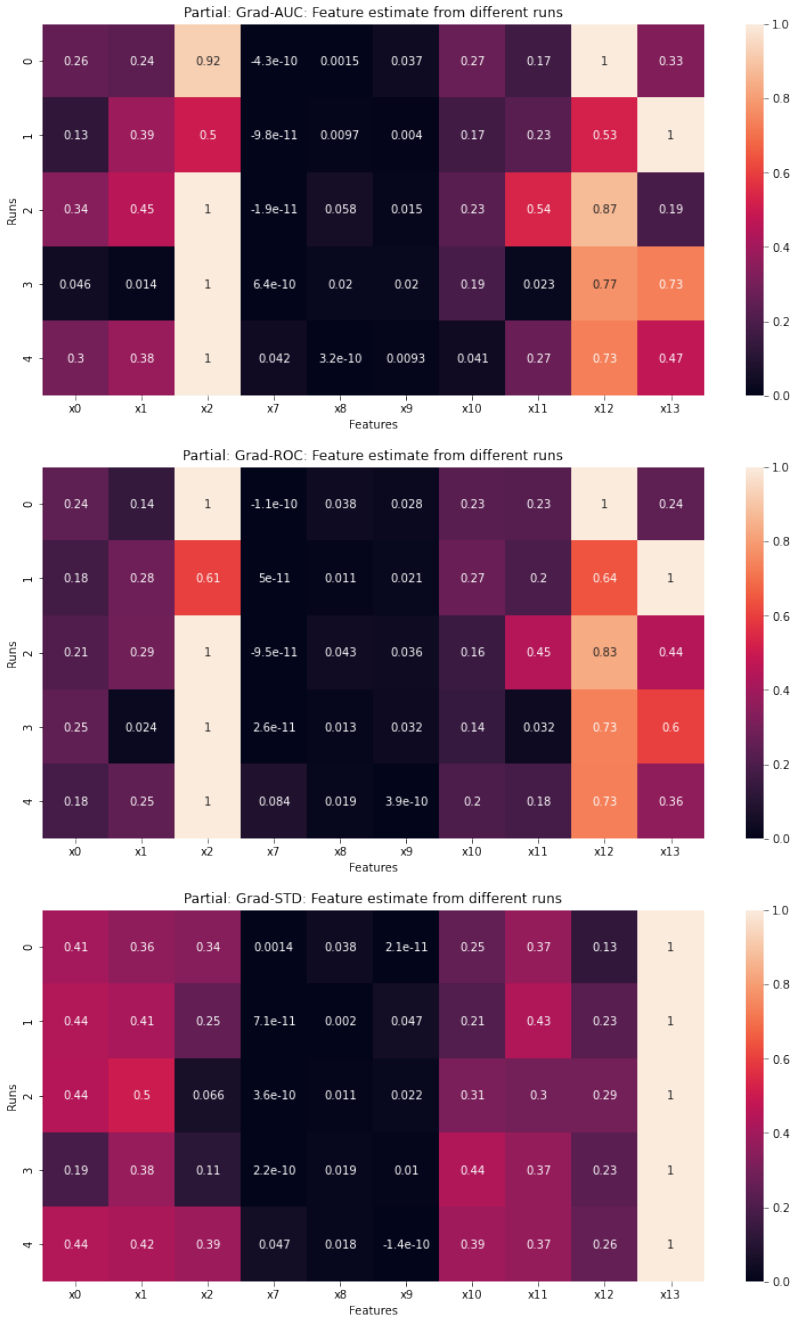

Figure 7. Visualisation of feature importance from E1: Simulated Dataset with the Partial Training paradigm. The pattern found in partially trained model resemble the pattern of normally trained model, this shows that there is a minimum difference in the estimate of importance between the two paradigms.

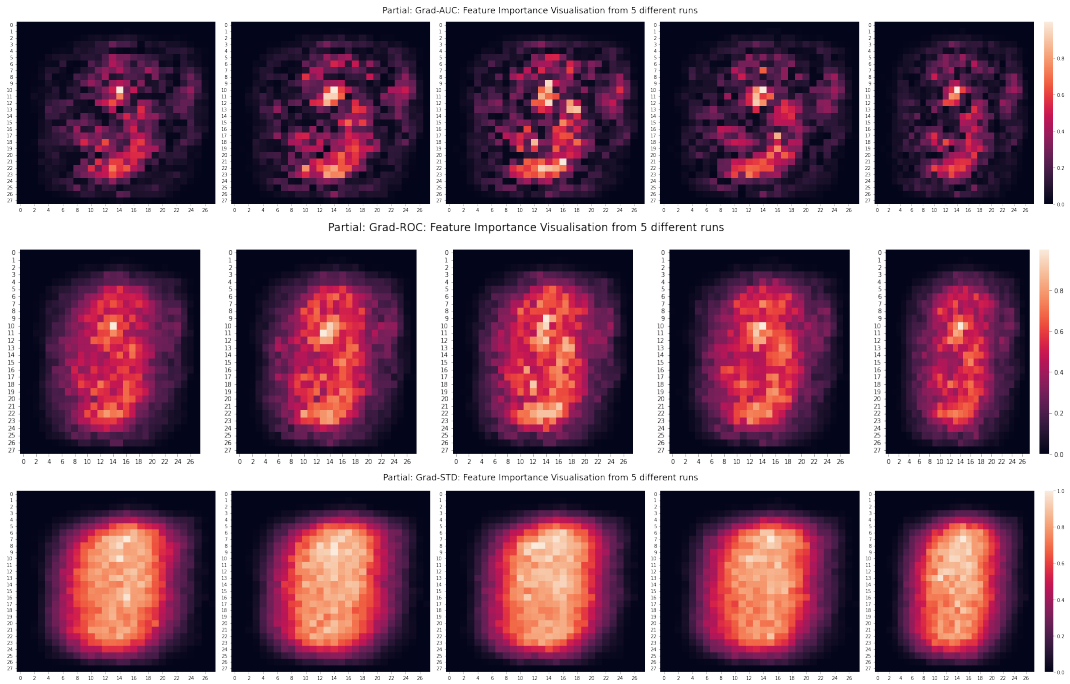

Figure 8. Visualisation of feature importance from E3: MNIST with the Partial Training paradigm. The pattern found in partially trained model resemble the pattern of normally trained model, this shows that there is a minimum difference in the estimate of importance between the two paradigms.