Module: Applied Machine Learning Coursework Report

Assignment type: Individual Practical Project (IPP)

Cyber Threat Detection Competition using Machine Learning

Eirini Zygoura

Student Number: 13177951

Wordcount: 2179

1. INTRODUCTION

The scope is to built a predictive model for cyber threat detection. To train, test and evaluate the proposed model, reduced CLS portion of the AWID dataset [1] is used. We were given two dataset files for training and testing the model (97044 and 40158 observations respectively). The datasets are balanced in terms of normal and attack instances. Variables 4 and 7 were removed as they provide temporal information. Raw, normalised and rescaled datasets are used for feature extraction. Feature selection techniques are applied. ML models determine the type of data to proceed with, the feature selection methodology to apply and the encoder models to use for feature extraction.

Logistic Regression applied to the dataset combined from the most powerful predictors of the original dataset and the extracted features from a Stacked Autoencoder led to Detection Rate of 99.8% and False Alarm Rate of 8.6%.

2. METHODOLOGY

2.1. DATA PREPARATION

Features that contain constant values (constant features) should be removed from the dataset (don't provide information, reduce performance). To reduce the training time, we filtered out all the columns of the dataset that have variance equal to zero and their observations were combined only to apply the filter. 82 input variables and the target variable were selected and the datasets were separated again.

Input variables have varying scales with unknown distribution of each feature (Gaussian or not) we applied separately normalisation and rescale methods to the input features.

2.2. CONSTRUCTING AND SELECTING FEATURES

2.2.1. FEATURE SELECTION

In order to reduce the overfitting effect, training time, and improve the accuracy of the model, three feature selection techniques were chosen. Parker et al. [2] proved that a "good enough" detection performance is by training their model by the rank-based MI theoretic feature selection (top 7). Similarly, 10 highly ranked features were selected from the set of 82 features, using:

- K-Best approach with chi-squared statistical test,
- Recursive Feature Elimination (RFE) with Logistic Regression algorithm
- Feature Importance obtained by Extra Trees Classifier (ETC).

2.2.2. FEATURE EXTRACTION

20 additional features were created using 3 different Autoencoder networks. After training the Autoencoder networks, only the encoders that produce features are retained:

- First model emanates from Stacked Autoencoder (2 hidden layers: 500/300 neurons, code layer: 20 neurons).
- Second model is constructed by a Denoising Autoencoder of the same layers and neurons and a noise factor of 0.4.

- Third encoder makes use of a Sparse Autoencoder with input and output layers and a code layer of 20 neurons. (activity regulariser = 0.00001).

All autoencoders were trained with batch size 200 and 20 epochs, binary cross entropy was defined as the loss parameter.

2.2.3. COMBINING FEATURES

New train and test datasets contain 20 extracted features from Stacked and Sparse encoder plus:

- Set 1: 82 original features (102 total).
- Set 2: 10 original features selected with ETC method (30 total).

Out of the above sets new datasets were created with ETC method:

- Set 3: 10 selected features from set 1
- Set 4: 10 selected features from set 2

2.3. SELECTION OF MACHINE LEARNING ALGORITHMS

ML algorithms considered for cyber thread detection are:

- Logistic Regression (LR),
- Linear Discriminant Analysis (LDA),
- K-Nearest Neighbours Classification (KNN),
- Support Vector Classification (SVC),
- Linear Support Vector Classification (LnSVC).

All the above algorithms are used in raw, normalised and rescaled data on:

- 82 selected features from constant column filtering
- 10 selected features (from 82 selected features) for each method described at 2.2.1
- 20 extracted features from each Autoencoder model described at 2.2.2

Applying all the above algorithms in all above datasets, using the classification accuracy as evaluation metric, selections were made on:

- Type of data to proceed with
- Feature selection methodology to apply,
- Encoder models for feature extraction
- ML models for further investigation.

For the next stage more evaluation metrics were added:

- Detection Rate (DR),
- Precision (Prec),
- False Alarm Rate (FAR),
- False Negative Rate (FNR),
- Errors Type I, II,
- Time to build model (TBM),
- Time to test model (TTM).

3. EVALUATION

3.1. FEATURE SELECTION EVALUATIONS

Performing K-Best methodology in raw and rescaled data had identical results. For the normalised data only 1 feature out of 10 was different, with different ranking score of each feature compared with other data types. The RFE method had more variations for each type of data. Comparing raw with normalised data, the results showed 5 out of 10 different selected features. Raw and rescaled data differed in 4 features and normalised and rescaled showed difference in 8 out of 10 features. ETC method presented more similar results for every dataset than RFE. For the raw data, only 1 feature differed from the normalised. The selection from the rescaled data was the same as for the raw, with the same ranking scores. The rank of the common features in each case of feature selection technique is similar. Results for each methodology and type of data are presented in Table 1 of the Appendix.

In the next stages of the process, ETC method was performed to Sets 1 and 2 (described at 2.2.3). The results are presented in Tables 2a and 2b of the Appendix. While selecting from Set 1, 5 features differed among the encoder models. 5-7 of the selected features came from the set of the original data and 5 of them appeared for every case of encoder. Regarding Set 2, among the encoders 7 of the selected features were in common and only 3 were the extracted features for every encoder.

Comparing the original features that were selected from these sets, most of them are repeated in both cases. It is an indication that these features are the most powerful predictors for cyber thread detection. There is also evidence for extracted features that are selected in both cases for each type of encoder. The majority of the selected features emanate from the original set.

Except features 8 and 9 all of our selected original features appeared in Aminanto's et al. [3] selected features considering all the methods he used. As they mention, features 4, 7 and 71 are essential for detecting impersonation attacks. Features 4 and 7 were removed from our dataset, but feature 71 appeared persistently in our selections (marked with green colour on Tables 1 and 2a, 2b). Lee et al. [4] selected 20 from the AWID set added 50 extracted features. The original features after the selection were 6 out of 20 (4, 7, 8, 9, 38 and 82). Parker et al. [2] also selected 20 features and found 4 of the original features included in the selection (4, 8, 38 and 82), all common with Lee's et al. [4] findings. All of our selected features also appeared in the above mentioned papers, which is a good indication to trust our selections.

3.2. FEATURE EXTRACTION EVALUATIONS

In order to evaluate the performance of the autoencoders Table 3 is created. Validation losses reveal terrible fit for the rescaled data in stacked and sparse autoencoders. The very low training loss (0.05) reveals overfitting that only the denoising autoencoder was able to overcome reporting validation losses equal to training (0.4). As for the raw data, for sparse autoencoder although a decrease is reported (comparing the first and the last epoch), there is a minimum of validation loss (0.4) in the epoch 15, still a high number comparing to the results for normalised data where validation loss is 0.17 with some small variations during the epochs. Stacked autoencoder showed rising validation loss over the epochs for raw and normalised data, with bigger increase for the raw. Normalised data had smaller variations in validation loss, keeping the score around 0.17 for every epoch.

All of the above reveal that normalised data fed into a stacked or sparse autoencoder could produce more representative features.

3.3. EVALUATION OF MACHINE LEARNING ALGORITHMS

3.3.1. PERFORMANCE ON ORIGINAL DATA

Table 4 presents the results of fitting ML models in Set 1. All the training accuracies are above 0.985. For KNN and SVC the testing accuracies are close to 0.5, revealing overfitting. LR and LDA scored testing accuracies above 0.93 in some cases. LDA performed very well for raw and rescaled data. LR had the best performance for normalised data and satisfactory for raw. LnSVC better scored for raw data, but only 0.65.

3.3.2. PERFORMANCE ON SELECTIONS OF THE ORIGINAL DATA

The effect of the overfitting for KNN and SVC persists after applying feature selection. K-Best's higher accuracy was only 0.675 for LDA on normalised data, in which RFE had the highest accuracy (0.958). RFE had bad results for the other models. LR, LDA and LnSVC scored 0.783-0.866 selecting from raw and rescaled features. Apart from KNN and SVC models ETC method scored testing accuracies above 0.90. (Table 5)

3.3.3. PERFORMANCE ON EXTRACTED DATA

Fitting ML models to the 20 extracted features of every encoder (Table 6) revealed also a lot of overfitting situations. In all cases training accuracy is very high. KNN and SVC showed better performance for denoising encoder for normalised data and for sparse encoder for the raw, with testing accuracies below 0.85. Rescaled data only with sparse encoder achieved a score above 0.5, which was 0.917 for LDA. Sparse encoder had best testing accuracy scores.

3.3.4. DECISIONS FOR THE PROPOSED MODEL

Concluding, Table 5 verifies the choice of ETC for feature selection method. Normalised data is preferred considering Tables 3, 6 and 4. Tables 3 and 6 indicate stacked and sparse encoders. ML models worthing further evaluation are LR, LDA and LnSVC (Tables 4, 5 and 6). From now on we proceed evaluations on the testing metrics, which is our target.

3.3.5. PERFORMANCE ON COMBINED DATA

Examining the Set 1 (Table 7), LDA and sparse encoder and LR and stacked encoder performed the best regarding their accuracies, DR and Precision. LDA with FNR=0 is able to detect all the intrusions but can categorise more cases of normal traffic as attack (FAR=0.105) comparing to other methods. LR and stacked encoder has lower FAR and very small FNR. Moderate performance had LR and sparse encoder. All the other methods showed unacceptable results.

Selecting 10 features from Set 1 (Set 3, Table 8) resulted in small increase in FNR and decrease in FAR for LDA and sparse encoder. Other metrics maintained in satisfactory levels. LR and stacked encoder can no longer be considered as a suggestion to detect intrusions as FNR got almost 0.5. Contrariwise, LDA acquired very good scores for every metric.

Set 2 applied on stacked encoder (Table 9) performed well on every model, detecting extremely high percentage of the attacks, especially LR and LDA, with LDA lacking a little in identifying normal traffic. Sparse encoder and LnSVC are not suggested for our purpose.

Compared to Set 2, for Set 4 (Table 10) the performance of the models was reduced for stacked encoder. For the sparse encoder LR showed little improvement, but still suffers from FNR>0.1. LDA improved a lot and is now an acceptable model to use with that encoder, along with LnSVC, which was the only case that produced acceptable results.

Regarding the TBM has maximum at 11.48 sec for LnSVC model and stacked autoencoder. Generally LnSVC demanded more time than LR and LDA that reported TBM below 2.62. The TBM time increases as the number of features increases .TTM varied from 0.11 to 0.15 sec.

3.4. SELECTION OF A PROPOSED MODEL

Best performance considering all the metrics and their importance on intrusion detection, was reported for LR model using the Set 2 for stacked encoder. FNR and Type II error are very small, FAR and error Type 1 are a little higher, still in acceptable values. This model can detect 98% of the intrusions (DR), but categorises incorrect as intrusions 8.6% of the instances.

3.4.1. EVALUATION OF THE PROPOSED MODEL

The accuracy of LR on the 82 original features was 96.5%, on the selected 10 of them 0.964% and on the extracted 79.1%, although the proposed model achieved 95.6%. Table 11 compares the metrics for each case. Besides the reduction of the accuracy and precision the model is more capable on detecting a thread since DR, FNR and Type II error scored better, in expense of increase of normal traffic instances categorised as attack.

3.5. COMPARISON AGAINST BASELINES

Table 12 provides a comparison of our model with previously tested models against the CLS portion of the AWID dataset. After Aminanto's et al. [3] our model is the second best in detecting impersonation attacks, but is the least capable on identifying correctly the normal traffic instances.

4. CONCLUSIONS

- Performing Feature Selection techniques to the combined data showed that the majority of the predictors emanate from the original set, which is unsatisfactory about the ability of the encoder models to produce strong predictors.
- Comparing the evaluation ML models on the extracted features and on the original data, the extraction has lowered the accuracy and increased the False Alarm Rate of the model. Although the Detection Rate was increased.
- Strong intrusion detection predictors 4 and 7 [3] are missing from your dataset, reducing the performance of our model, resulting to lacking in comparison with established models

5. REFERENCES

[1] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, ‘‘Intrusion detection in 802.11 networks: Empirical evaluation of threats and a pub- lic dataset,’’ IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 184–208, 1st Quart., 2016.

[2] L. R. Parker, P. D. Yoo, T. A. Asyhari, L. Chermak, Y. Jhi, and K. Taha, ‘‘DEMISe: Interpretable deep extraction and mutual information selection techniques for IoT intrusion detection,’’ in Proc. 14th Int. Conf. Availability, Rel. Secur., 2019, pp. 1–10.

[3] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo and K. Kim (2018) Deep abstraction and weighted feature selection for Wi-Fi impersonation detection, IEEE Transactions on Information Forensics and Security, 13(3), 621–636.

[4] S. J. Lee, P.D. Yoo, T. A. Asyhari, Y. Jhi, L. Chermak, C. Y. Yeun, and K. Taha. 2020. IMPACT: Impersonation attack detection via edge computing using deep autoencoder and feature abstraction. IEEE Access (2020)

APPENDIX

K-Best			RFE			ETC		
raw	norm	resc	raw	norm	resc	raw	norm	Rest
71	71	71	9	47	47	50	71	50
51	67	51	47	50	64	71	50	71
67	51	67	67	61	68	67	67	67
47	47	47	70	67	70	47	51	47
154	154	154	82	71	82	51	47	51
50	50	50	94	82	107	68	73	68
8	8	8	98	94	108	73	82	73
9	9	9	107	98	112	38	142	38
68	68	68	112	130	122	82	68	82
82	73	82	141	140	141	66	38	66

TABLE 1 : Selected features using K-Best, RFE and ETC methods for raw, normalised and rescaled original datasets (For K-Best and ETC methods variables are listed from top to bottom according to ranking scores)

Encoder type	Original features	Extracted features
Stacked	71, 67, 73, 50, 51	2, 15, 4, 14, 3
Sparce	71, 50, 73, 67, 47, 51, 68	4, 2, 11

TABLE 2a: Selected features from the full set of the original plus the extracted features

Encoder type	Original features	Extracted features
Stacked	67, 50, 68, 71, 51, 47, 73	2, 15, 5
Sparse	67, 50, 68, 71, 73, 47, 51	4, 2, 3

TABLE 2b: Selected features from the selected set of the original plus the extracted features

* The labels of the original features correspond to the labels of the original dataset which is compatible with the AWID dataset.

** Extracted features are labeled from 0 to 19, according to Python column enumeration

*** The label of all features presented in the tables have been transformed from the output of the algorithm, to match the AWID dataset enumeration and the arbitrary numbering of the extracted features

**** Lee's et al [4] selected best features are marked with orange colour, and Parker's et al. [2] and Lee's et al. [4] with blue

	Training Loss		Validation Loss	
	First epoch	Last epoch	First epoch	Last epoch
raw	0.1379	0.0588	0.5096	0.4517
norm	0.251	0.1468	0.1777	0.1713
resc	0.1324	0.0499	-4443283.500	-12473388.000

TABLE 3a: Performance of the Stacked Autoencoder on raw, normalised and rescaled data

	Training Loss		Validation Loss	
	First epoch	Last epoch	First epoch	Last epoch
raw	0.4764	0.4061	0.4554	0.4291
norm	0.4903	0.4165	0.4447	0.4174
resc	0.4682	0.3910	0.4168	0.3915

TABLE 3b: Performance of the Denoising Autoencoder on raw, normalised and rescaled data

	Training Loss		Validation Loss	
	First epoch	Last epoch	First epoch	Last epoch
raw	0.3317	0.0596	0.2602	0.3857
norm	0.3898	0.1474	0.1716	0.1757
resc	0.3138	0.0508	-873671.750	-11094170.000

TABLE 3c: Performance of Sparse Autoencoder on raw, normalised and rescaled data

	Raw		Normalised		Rescaled	
	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
LR	0.992	0.868	0.989	0.965	0.996	0.506
LDA	0.985	0.931	0.985	0.614	0.985	0.931
KNN	0.999	0.534	0.999	0.532	1.000	0.536
SVC	0.995	0.530	0.995	0.531	0.997	0.499
LnSVC	0.995	0.652	0.992	0.525	0.997	0.509

TABLE 4: Training and Testing Accuracies for raw, normalised and rescaled data for Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbours Classification (KNN), Support Vector Classification (SVC) and Linear Support Vector Classification (LnSVC)

* Accuracies close to 0.5 are marked with grey colour, training accuracies greater than 0.90 marked with red colour.

	K-Best		RFE		ETC	
	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
LR	0.961	0.500	0.985	0.861	0.970	0.959
LDA	0.929	0.277	0.969	0.820	0.938	0.916
KNN	0.999	0.500	1.000	0.531	0.998	0.521
SVC	0.992	0.500	0.997	0.521	0.988	0.502
LnSVC	0.960	0.500	0.993	0.866	0.971	0.958
NORMALISED DATA						
LR	0.929	0.500	0.987	0.520	0.954	0.964
LDA	0.928	0.675	0.974	0.958	0.934	0.907
KNN	0.998	0.500	0.999	0.532	0.999	0.528
SVC	0.982	0.500	0.992	0.527	0.989	0.507
LnSVC	0.941	0.499	0.988	0.519	0.962	0.967
RESCALED DATA						
LR	0.961	0.500	0.995	0.813	0.969	0.959
LDA	0.929	0.277	0.980	0.783	0.938	0.916
KNN	0.999	0.500	0.999	0.535	0.998	0.521
SVC	0.992	0.500	0.997	0.494	0.988	0.502
LnSVC	0.960	0.500	0.995	0.816	0.971	0.958

TABLE 5: Training and Testing Accuracies for the selected features from the original dataset, using K-Best approach (chi-squared), Recursive Feature Elimination (RFE) with Logistic Regression and feature importance obtained by Extra Trees Classifier (ETC) on raw, normalised and rescaled data, for Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbours Classification (KNN), Support Vector Classification (SVC) and Linear Support Vector Classification (LnSVC)

* Accuracies close to 0.5 are marked with grey colour, training accuracies greater than 0.90 marked with red colour.

	RAW DATA		NORMALISED DATA		RESCALED DATA	
	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
STACKED ENCODER						
LR	0.996	0.826	0.971	0.791	0.989	0.516
LDA	0.976	0.840	0.953	0.875	0.973	0.543
KNN	1.000	0.533	1.000	0.533	1.000	0.500
SVC	0.998	0.533	0.997	0.528	0.999	0.500
LnSVC	0.996	0.532	0.974	0.769	0.988	0.516
DENOISING ENCODER						
LR	0.989	0.495	0.986	0.864	0.995	0.510
LDA	0.977	0.940	0.974	0.500	0.980	0.468
KNN	1.000	0.529	0.999	0.841	1.000	0.500
SVC	0.997	0.529	0.987	0.690	0.998	0.500
LnSVC	0.990	0.496	0.990	0.956	0.995	0.506
SPARSE ENCODER						
LR	0.992	0.945	0.991	0.859	0.995	0.473
LDA	0.978	0.944	0.978	0.947	0.982	0.917
KNN	0.999	0.633	0.999	0.529	1.000	0.500
SVC	0.993	0.896	0.993	0.526	0.998	0.499
LnSVC	0.992	0.848	0.992	0.860	0.994	0.470

TABLE 6: Training and Testing Accuracies for the extracted features from the Stacked, the Denoising and the Sparse Encoder, on raw, normalised and rescaled data, for Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbours Classification (KNN), Support Vector Classification (SVC) and Linear Support Vector Classification (LnSVC)

* Accuracies close to 0.5 are marked with grey colour and training accuracies greater than 0.90 are marked with red colour.

	Stacked Encoder			Sparse Encoder		
	LR	**LDA**	**LnSVC**	**LR**	**LDA**	**LnSVC**
Accuracy	0.938	0.461	0.530	0.857	0.947	0.710
DR	0.955	0.793	0.072	0.761	1.000	0.455
Prec	0.924	0.476	0.859	0.942	0.905	0.929
FAR	0.078	0.872	0.012	0.047	0.105	0.035
FNR	0.045	0.207	0.928	0.239	0.000	0.545
Type I error	1574	17506	238	939	2113	702
Type II error	899	4158	18635	4806	1	10934
TBM (sec)	2.37	2.49	11.48	2.09	2.62	8.90
TTM (sec)	0.13	0.11	0.12	0.12	0.12	0.12

Table 7: Evaluation of Logistic Regression (LR), Linear Discriminant Analysis (LDA) and Linear Support Vector Classification (LnSVC) for Set 1 for Stacked and Sparse encoder

	Stacked Encoder			Sparse Encoder		
	LR	**LDA**	**LnSVC**	**LR**	**LDA**	**LnSVC**
Accuracy	0.709	0.924	0.671	0.809	0.924	0.815
DR	0.503	0.931	0.423	0.691	0.927	0.691
Prec	0.854	0.919	0.838	0.903	0.921	0.918
FAR	0.086	0.082	0.082	0.074	0.079	0.062
FNR	0.497	0.069	0.577	0.309	0.073	0.309
Type I error	1730	1653	1646	1492	1591	1244
Type II error	9976	1380	11586	6195	1465	6195
TBM (sec)	0.45	0.16	5.55	0.26	0.14	1.02
TTM (sec)	0.12	0.12	0.11	0.12	0.11	0.11

Table 8: Evaluation of Logistic Regression (LR), Linear Discriminant Analysis (LDA) and Linear Support Vector Classification (LnSVC) for Set 3 for Stacked and Sparse encoder

*Metrics with excellent scores are marked with red colour and metrics revealing poor performance with grey

	Stacked Encoder			Sparse Encoder		
	LR	**LDA**	**LnSVC**	**LR**	**LDA**	**LnSVC**
Accuracy	0.956	0.928	0.958	0.858	0.833	0.662
DR	0.998	0.999	0.937	0.757	0.763	0.359
Prec	0.921	0.875	0.979	0.949	0.887	0.911
FAR	0.086	0.143	0.020	0.041	0.098	0.035
FNR	0.002	0.001	0.063	0.243	0.237	0.641
Type I error	1719	2866	409	817	1958	702
Type II error	49	23	1258	4871	4751	12880
TBM (sec)	1.16	0.42	7.09	0.97	0.42	3.74
TTM (sec)	0.12	0.15	0.11	0.13	0.13	0.11

Table 9: Evaluation of Logistic Regression (LR), Linear Discriminant Analysis (LDA) and Linear Support Vector Classification (LnSVC) for the combination of Set 2 for Stacked and Sparse encoder

	Stacked Encoder			Sparse Encoder		
	LR	**LDA**	**LnSVC**	**LR**	**LDA**	**LnSVC**
Accuracy	0.904	0.912	0.927	0.898	0.932	0.940
DR	0.936	0.935	0.936	0.861	0.927	0.931
Prec	0.879	0.894	0.919	0.929	0.936	0.948
FAR	0.129	0.111	0.082	0.065	0.063	0.051
FNR	0.064	0.065	0.064	0.139	0.073	0.069
Type I error	2590	2229	1651	1311	1264	1017
Type II error	1284	1308	1284	2800	1466	1377
TBM (sec)	0.35	0.14	3.5	0.29	0.13	1.77
TTM (sec)	0.12	0.13	0.11	0.12	0.15	0.11

Table 10: Table 9: Evaluation of Logistic Regression (LR), Linear Discriminant Analysis (LDA) and Linear Support Vector Classification (LnSVC) for Set 4 for Stacked and Sparse encoder

* Metrics with excellent scores are marked with red colour and metrics revealing poor performance with grey

	A	B	C
Accuracy	0.965	0.964	0.956
DR	0.986	0.991	0.998
Prec	0.946	0.940	0.921
FAR	0.056	0.063	0.086
FNR	0.014	0.009	0.002
Type I error	1121	1273	1719
Type II error	281	180	49

Table 11: Evaluation of the LR model on the full set of the 82 original features (A), the selected set of the original features (B) and the proposed model (C)

* Metrics that scored better in the proposed model are marked with red colour

	Detection Rate (%)	Accuracy (%)	False Alarm Rate (%)
Aminanto et al. [3]	99.918	99.97	0.012
Parker et al. [2]	99.07	98.04	2.96
Lee et al.[4]	97.64	98.22	1.20
Our model	99.8	95.6	8.6

Table 12: Comparison of our model with previously tested models