Birkbeck, University of London

Applied Machine Learning

Individual Practical Project

A machine learning based strategy for impersonation attacks detection using the Aegean Wi-Fi Intrusion Dataset

Student: Alexandra-Gabriela Gheorghe
S/N: 13162760

Word count: 2102

1. Introduction

The purpose of this project is to build a machine learning based intrusion detection system for recognition of impersonation attacks in wireless networks. Given the increasing reliance on edge computing, limited computational capability of IoT devices and the importance of real-time detection of cyber-security threats, the ideal strategy aims at minimizing both memory resources used and algorithm running times. To achieve this, current research proposes workflows based on feature abstraction, extraction and selection for dimensionality reduction, coupled with supervised classifiers capable of recognizing intrusive attacks[1][2][3].

2. Data and strategy selection

The training and testing data for the project were derived from a reduced portion of the AWID dataset built by Kolias *et al*[4]. The data is comprised of 152 features and a well-balanced binary class label, with a 50:50 distribution of the samples across the two labels (normal traffic and impersonation attack). The variable types were inspected via Wireshark. The experimental approach employed includes a data cleaning step, to reduce the number of constant, sparse and low variance features, followed by a feature abstraction step using a trained stacked autoencoder, a feature selection stage, and final classification. In parallel, dimensionality reduction by principle component analysis was performed for benchmarking the classification performance on subsets obtained after the feature selection stage.

3. Data cleaning and pre-processing

Parker *et al*[2] note that more than 100 features contained in (or derived from) the AWID dataset are statistically independent of the class outcome, as the amount of information gained about the class label by measuring each of these features separately (i.e. mutual information) is zero. In an attempt to remove features that are unlikely to contribute to the predictive power of the final model, variables with a single constant value across all the samples in the training set were filtered out. Histograms were built to better assess feature distributions, revealing continuous variables with non-normal distributions, some with low variance, as well as some binary variables with varying degrees of class imbalance (Figure 1). Binary features with severe class imbalance (less than 1000 observations falling under one of the classes) were removed. Duplicate variables were filtered, and samples were normalized to unit length in preparation for machine learning.

Univariate feature selection was performed using the scores obtained by a chi-square test of independence. The lowest scoring features (only scores of very low magnitude were chosen to limit the extent of feature selection) were removed.

The data pre-processing steps led to a ~74% reduction in the number of features. 40 features were retained for further analysis.

3. Feature abstraction and selection

A stacked autoencoder (SAE) was trained to reduce the dimensionality of the data to 10 latent vectors (Figure 2). Online guidance was employed in visualizing the autoencoder structure and learning curve[5]. A Lasso activity regularization parameter was added to the coding layer, to encourage sparse learning and increase the generalisation/de-noising power of the model. The autoencoder was used to compress both the training and the test data, and the 10 abstract features obtained were appended to the original datasets respectively.

Popular classifiers (logistic regression classifier, decision tree, k-nearest neighbour classifier and naïve Bayes classifier) were fit on the newly obtained training data and tested, both via cross-validation and directly on the test data. The results show a tendency to severely overfit the training data, as the model performs excellently on the cross-validated subsets (which are more likely to share common patterns to the training data used at each split), but performances are poor (close to no skill) on the testing set (Figure 3). The conclusions derived from this observation are two-fold:

1. The patterns of the training and testing data differ substantially. Subsequent feature-selection and model building steps need to introduce bias to balance the bias-variance ratio of the final model, in order to ensure more accuracy on the testing set. The final model needs to be optimized for high generalization capabilities.

2. Subsequent grid/random searches for the best subset of hyperparameters might prove difficult, given that optimizing performance on validation sets is not guaranteed to result in increased performance on the testing data, and might encourage overfitting.

Recent works by Parker *et al*[2] and Lee *et al*[1] demonstrate the effectiveness of coupling mutual information (MI) metrics (a measure of dependence between two features) with Decision Trees-based wrapper feature selection in producing optimal feature subsets.

To assess the feature selection capabilities of Decision Trees, recursive feature elimination (RFE) was performed, allowing the wrapper to choose the best subset of features via cross-validation. The results show that Decision Trees tend to overfit the training data even at very low depths, and may not be the most effective estimator for feature importances (Figure 4).

The alternative chosen was a Random Forest (RF)-based wrapper method, which uses the gini index (a measure of node impurity) to establish feature importance. The model fits a high number of bootstrapped trees of depth one. Essentially, the method assesses the ability of each predictor to partition the feature space in such a way that ensures maximal purity of the terminal nodes (i.e. best separation between the two classes). This ability is derived from the gini index and closely resembles mutual information metrics. In this case, overfitting is prevented due to bootstrapping and the limited depth of the trees (which introduces assumptions of both feature independence and linearity). To prevent the same strong predictor to be repeatedly chosen, and to decorrelate the trees, a limited number of randomly chosen variables is visible to the model at each split. James *et al*[6] recommend using the square root of the total number of predictors.

The feature subset selected by the RF Wrapper, as well as the time to build, were compared with the results generated by RFE method using a Logistic Regression (LR) Estimator. In both cases, a subset of 7 predictors was retained, after consulting the analysis provided by Lee *et al*[1] on historical model performances against number of features.

Interestingly, the methods selected different subsets of features (only two common features) and assigned importance to different features produced by SAE compression (Table 1). The RF wrapper proved to require more time to build comparatively to the LR-RFE approach.

Finally, principal component analysis was performed. The dimensionality was reduced to 5 principal components, capable of conserving 90% of the variance in the data set (Figure 5). The threshold was set at 90% with the aim of filtering out some of the noise and outliers that may cause overfitting in subsequent models. Guidance in coding the model was followed from Aurélien Géron's book. [7]

4. Final classifier choice

Given the good results obtained by Parker *et al*[2] and Lee *et al*[1] using linear classifiers, a logistic regression classifier and a support vector classifier with a linear kernel were considered as candidates (Table 2). A logistic regression classifier was trained separately on the autoencoder output (10 variables), the feature subsets obtained by RF and LR selection respectively (7 variables each), and the principal components analysis output (5 dimensions).

As expected, the highest accuracy was obtained on the LR-RFE generated subset (97.5%), with a drastic increase in performance (>40%) compared with using the whole feature set. This shows that the RFE approach was highly efficient at selecting the variables with high coefficient values and filtering out low-coefficient noisy variables that contributed to overfitting.

Interestingly, the performance of the RF-selected subset sits slightly below that of the LR-RFE subset in terms of accuracy, however, the detection rate is 0.6% higher, sitting very close to 100%, indicating a nearly perfect coverage of impersonation attacks. The accuracy of RF-selected subsets is very close to that of PCA-derived data. These results corroborate the linear nature of the relationship observed between predictors and class label in the AWID dataset in previous research efforts.

We tried to assess whether a support vector classifier (SVC) can bring further performance improvements. Linear SVC seeks to find the best separating (maximal margin) hyperplane between the two classes (exclusively dependent on a subset of support vectors), while still allowing few misclassifications (the number of admitted violations depending on the C parameter). As such, linear SVC takes advantage of the geometry of the data, and might produce better results than logistic regression when classes are well separated (as emphasized by James *et al* [6]). Adjusting the C parameter allows control over regularization of the method – a higher C parameter denotes decreased tolerance for observations violating the margin (and may cause overfitting), while a lower C value introduces bias by allowing a larger number of misclassifications. A grid search was performed across a range of C values to assess cross-validation performance. The values chosen for C were relatively small, as computational times increase with the constraints added by a higher C value. The best performing model assessed via grid search cross-validation (with C = 1.0) was used for predictions on the SAE output/test feature subsets/PCA output. The model improved prediction accuracy when the LR-RFE features subset was used, however, the detection rate remained similar to that of the LR model previously fitted. The prediction accuracy on the RF-selected subset decreased by 3%, similar to the accuracy on the PCA data. However, the detection rate on the RF-selected subset remains 3% higher compared to the PCA data. A decrease in the C parameter to 0.01 causes worsened accuracy performance on all the test subsets, with the exception of the autoencoder output. Interestingly, introducing additional bias improves the detection rate on both the PCA and autoencoder output, but not on the wrapper selected feature subsets. Considering all these observations, the model with C=1 was retained as the best performing model. Higher cost values were not trialled due to the subsequent increase in computational time.

Given the observation that the LR-RFE subset performs better than the RF-selected subset, the experiment assessed whether introducing non-linearity in the models fitted on the RF-selected subset leads to improved detection performance. We assessed the performance of a random forest classifier and a multi-layer perceptron (MLP) (Table 3).

Considering the tendency of decision trees to overfit the training data even at low depths, a random forest was built using 500 bootstrapped trees, and the optimal depth for these trees was assessed via grid search cross-validation. The grid search results show that a maximum depth of 3 should be optimal to stabilize the bias:variance ratio, as the model seems to shoot up in cross-validation performance at greater depths, indicating a potential tendency to overfit. As such, each tree was grown to a depth of 3, and decorrelation of the trees was achieved by considering the square root of the number of predictors at each split. Comparative to linear classifiers, this model achieves modest accuracy performances and relatively low detection rates.

Finally, a MLP was built, using the ReLU activation function and the 'Adam' stochastic gradient descent optimizer. Initially, the MLP showed severe overfitting of the training data even at a low epochs number. To counteract this, a tanh activation function was chosen to flatten the gradient and Lasso kernel regularization was added to the hidden layers. After optimization, the performance on the testing data improved to 93.0% accuracy, but a relatively modest detection rate of 89.0%, showing lower sensitivity to impersonation attacks than linear models. Subsequently increasing the strength of regularization, as well as trying to increase the epoch numbers and number of neurons did not lead to improved performance.

5. Comparison to baselines and conclusion

These findings reiterate the efficiency of feature abstraction and selection in minimizing the dimension of the AWID dataset and producing representative feature subsets (Table 4). In particular, stacked autoencoders were found to efficiently compress and de-noise high dimensional data, and wrapper-based feature selection using mutual information indices (i.e. gini index) and regression coefficients were both effective at delimitating relevant subsets. Linear classification methods perform well in separating normal traffic from impersonation attacks on the reduced subsets. These findings indicate that a combination of SAE compression, recursive feature elimination using LR and LR classification performs the best on the testing set, offering the advantage of both an interpretable feature selection strategy and a quick-to-train and easily interpretable classification method.

The LR-based selection is preferable to random forest-based feature elimination, and LR classifiers are preferred to linear SVC classifiers employed in the final step, due to their low resource requirements allowing rapid detection. RF-based selection coupled with LR is recommended for maximization of detection rates, but introduces additional time costs. With increasing sizes of the training set, the underlying patterns in the data might deviate from linearity, and the complementary non-linear classifiers proposed, i.e. random-forest detection, MLP detection should be considered for further analysis.

APPENDIX

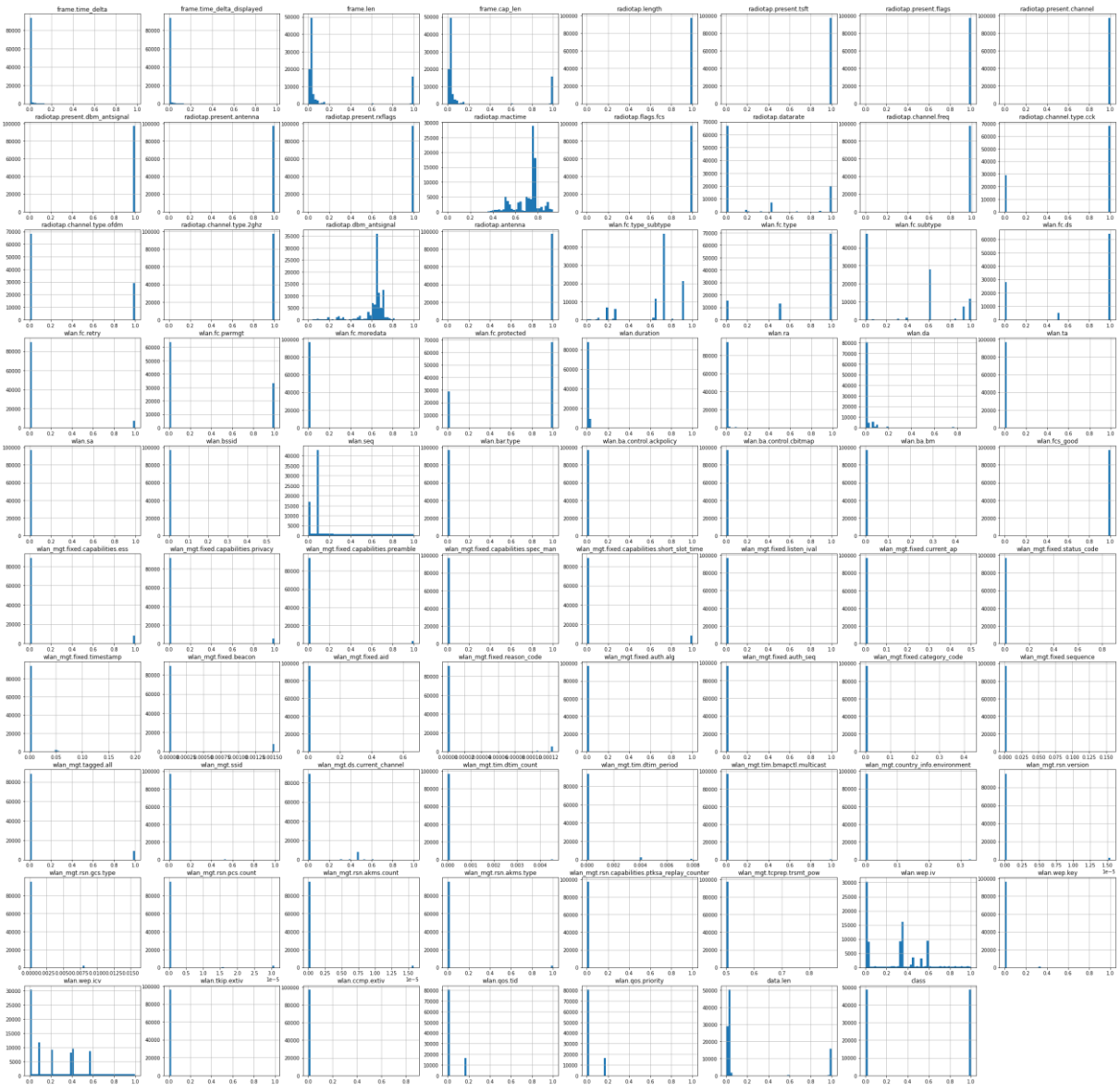

Figure 1. *Histograms of non-constant features in the reduced AWID dataset.*

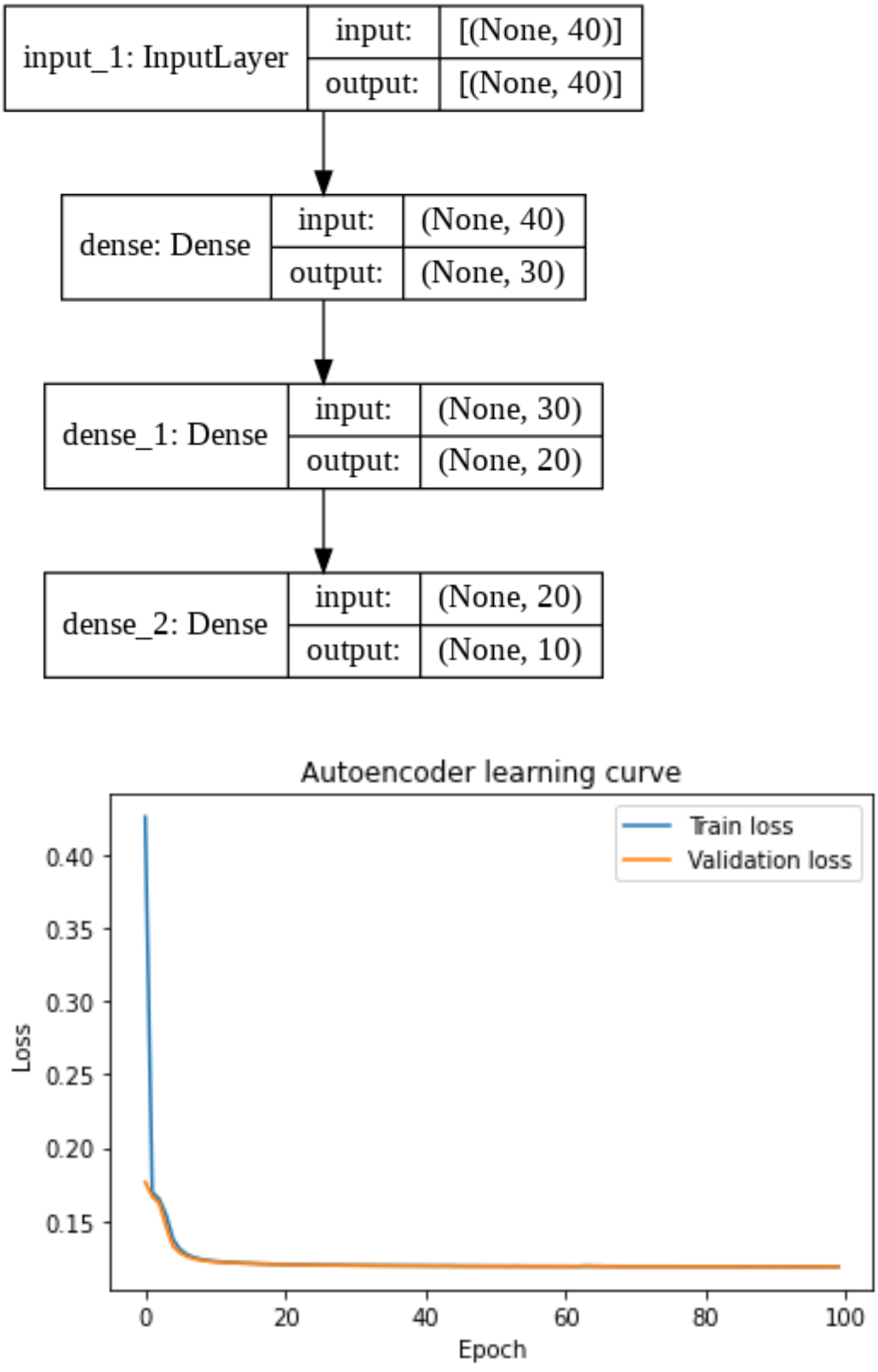

Figure 2. *Structure and learning curve of the stacked autoencoder. The two Dense hidden layers progressively decrease in the number of nodes, achieving feature compression. The autoencoder reconstructs the training and validation examples similarly, without overfitting the training data. The relatively high loss value may indicate efficiency in denoising/filtering during the encoding step, which prevents the decoder from fully reconstructing the original data.*

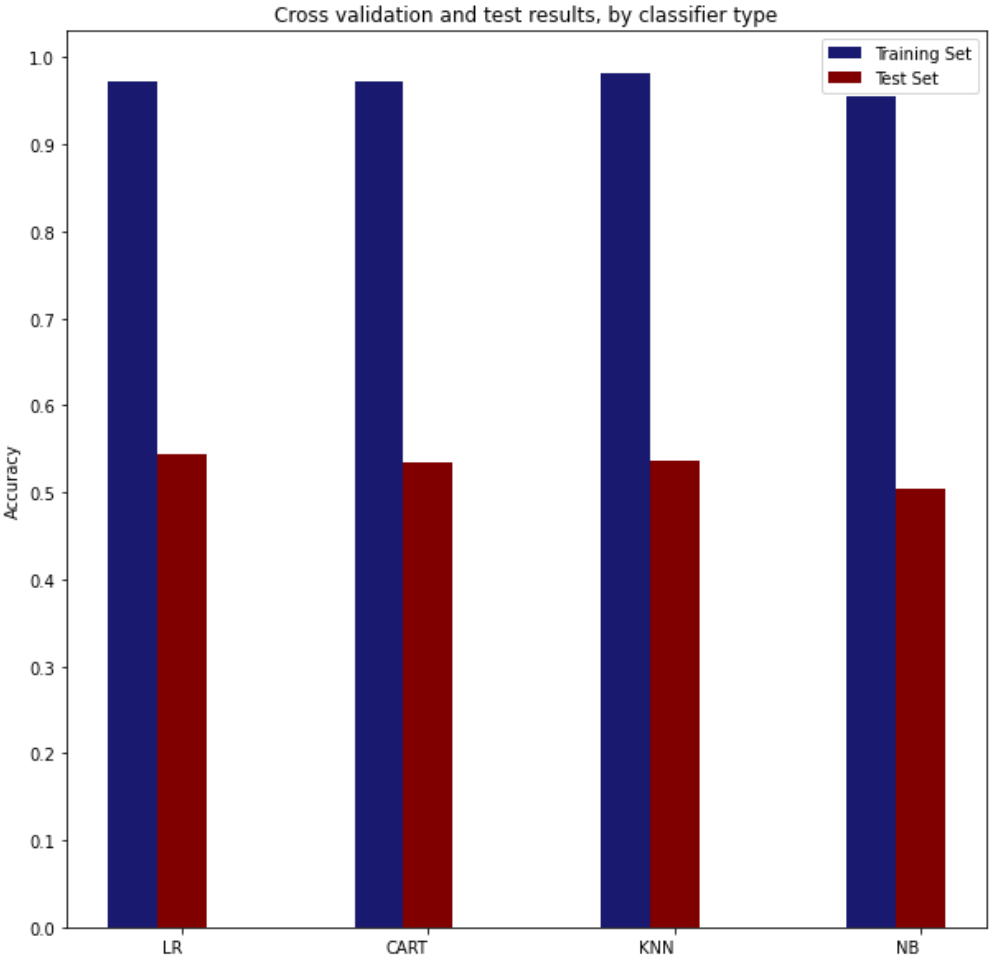

Figure 3. *Accuracy performance of different classifiers trained on a set comprised of original and encoded features. All classifiers assessed tend to overfit validation sets derived during cross-validation from the training data. Performance on the test data set is poor and close to no skill. We can conclude that the training and test data have different underlying patterns and introducing bias through feature elimination and optimization is recommended.*

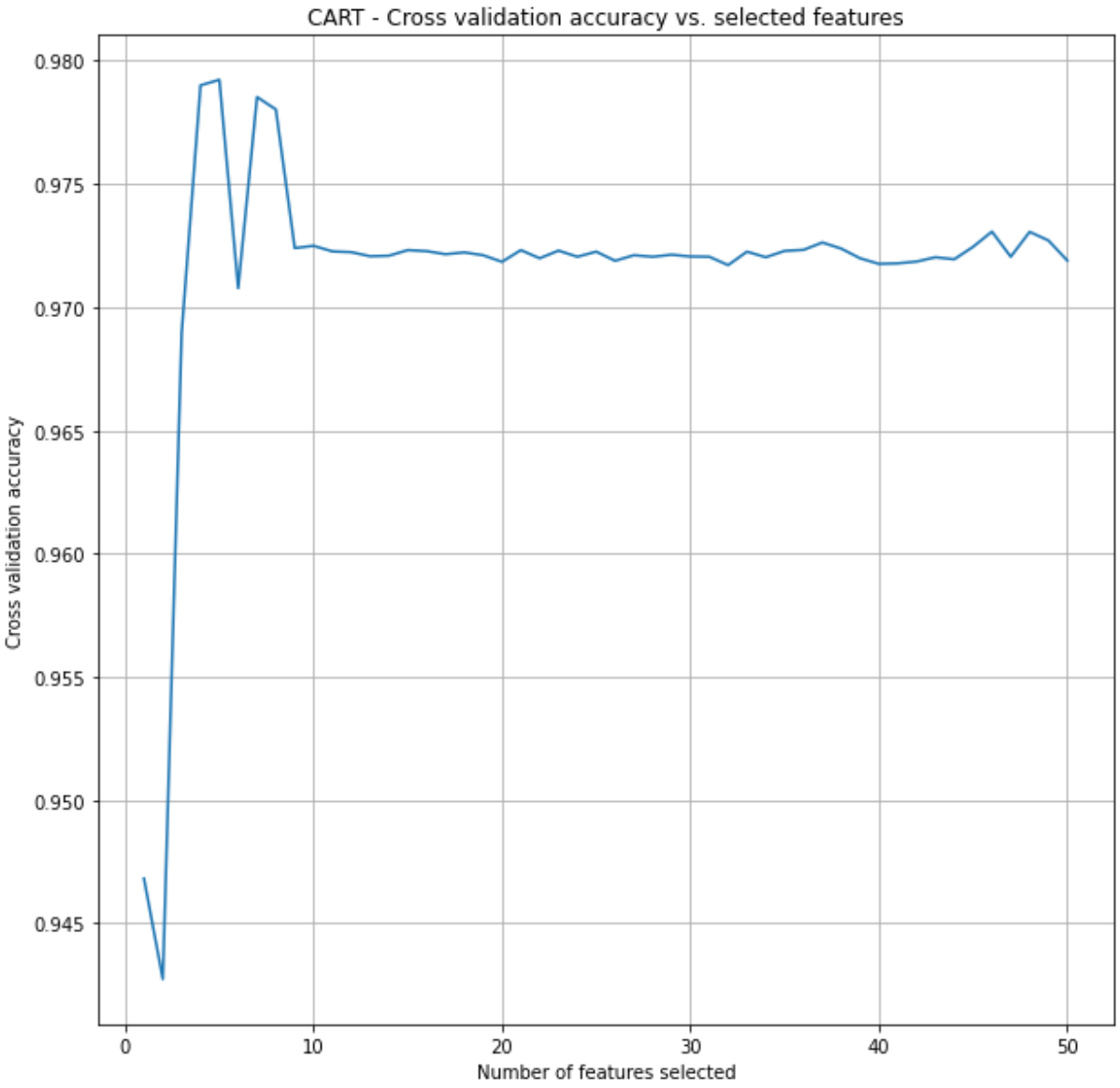

Figure 4. *Decision trees overfit the patterns in the training data even at very low depths. They perform well on subsets derived from the training data, but given previous observations, this might lead to poor performance on the test data. A random forest approach would be more suitable for feature selection, as bias can be introduced via optimizing tree depth and visible predictors, and predicting on bootstrapped data.*

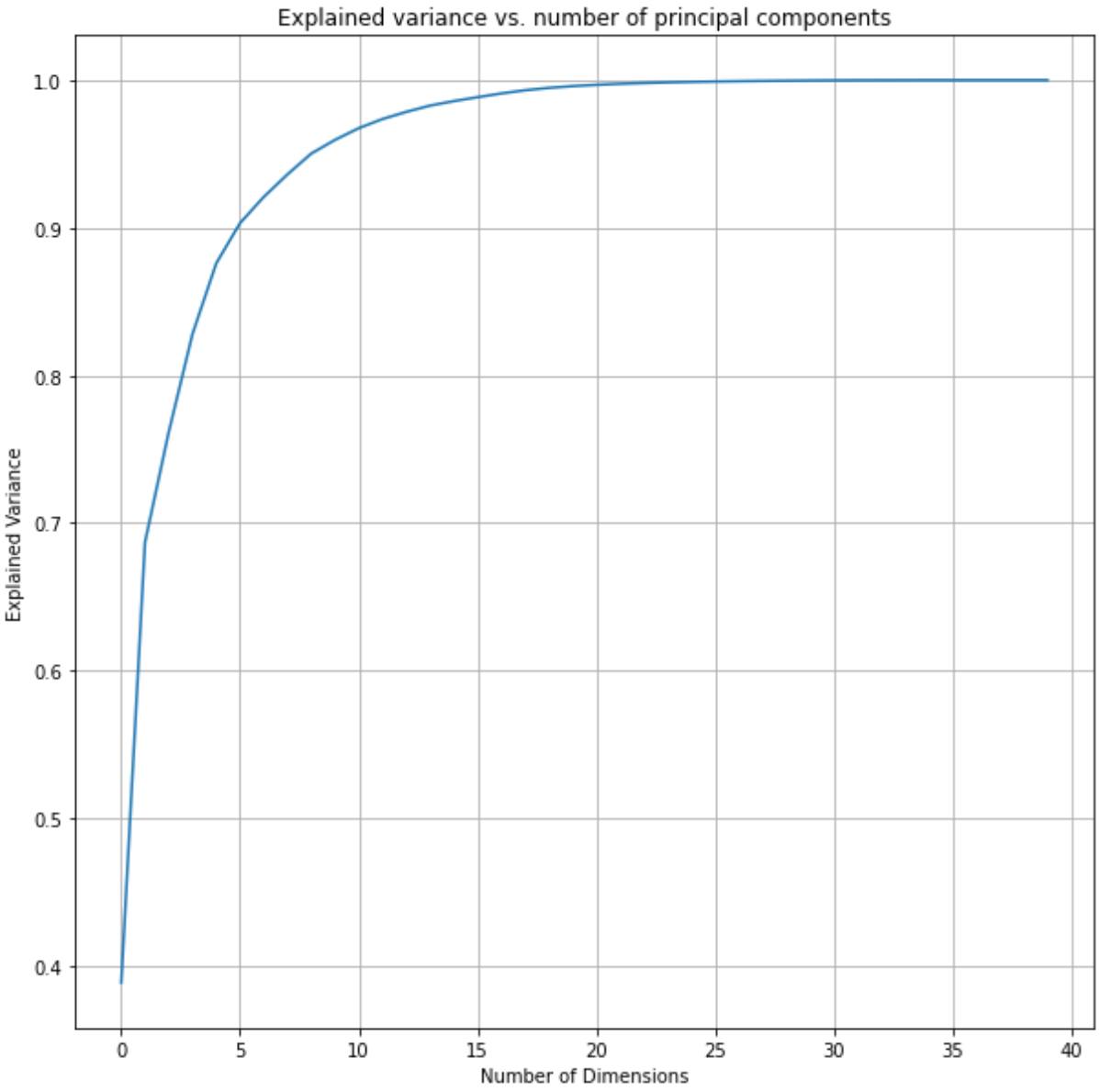

Figure 5. *Reducing the dimensionality of the data to 5 principal components conserves 90% of the variance observed in the dataset.*

	RF wrapper	LR-RFE wrapper
Features chosen	*wlan.fc.type* *wlan.fc.subtype* *wlan.fc.ds* *wlan.fc.pwrmgt* *wlan.fc.protected* *6* *7*	*radiotap.datarate* *wlan.fc.subtype* *wlan.seq* *wlan.fc.pwrmgt* *wlan_mgt.fixed.capabilities.preamble* *wlan_mgt.fixed.timestamp* *4*
Time to build	29.7 s	17.1 s

Table 1. *Feature subsets chosen by a random forest wrapper (feature importances based on gini index) versus feature subsets chosed by recursive feature elimination using a logistic regression estimator (features scored by regression coefficients). The subsets generated are different, and the two methods place weight on different vectors generated by encoding using the trained SAE.*

	LR				Linear SVC (C = 1.0)			
	On SAE output	On RF-subset	On LR-RFE subset	On PCA output	On SAE output	On RF-subset	On LR-RFE subset	On PCA output
Accuracy	93.03	95.17	97.52	95.98	83.56	92.12	98.28	92.72
F1	93.01	95.38	97.56	96.10	81.22	92.16	98.30	92.49
Detection Rate	92.68	99.81	99.29	99.02	71.10	92.66	99.26	89.63
MCC	86.07	90.72	95.10	92.13	69.30	84.24	96.58	85.60

Table 2. *Performance of linear classifiers on different feature subsets/compressed representations of the AWID data. Accuracy is chosen as first metric, as the predicted class is well-balanced. Detection rate (recall) is used to assess model sensitivity to impersonation attacks, while the F1 metric describes the balance between precision and recall. The Matthews correlation coefficient (MCC) takes into consideration performance on all the four positions of the confusion matrix, i.e. true and false positives, true and false negatives. The best detection rate can be observed after using a LR classifier on the RF-selected subset (99.8%), while the best accuracy performance is obtained using a linear SVC on the LR-RFE subset (98.28%). Given that LR requires less resources at the cost of a minimal accuracy loss (~0.7%) compared to SVC on the LR-RFE data, but offers superior detection rates on both the RF and LR-RFE subsets, it will be suggested as the most suitable linear separation method in this case. Both linear classifiers perform well on the PCA vectors, showing that PCA and feature abstraction and selection methods are both efficient in effectively reducing the dimensionality of the reduced AWID data, while conserving patterns relevant to the final prediction.*

	LR	Linear SVC	RF	MLP
Accuracy	95.17	92.12	87.17	93.09
F1	95.38	92.16	86.07	92.79
Detection Rate	99.81	92.66	79.33	89.00
MCC	90.72	84.24	75.26	86.46

Table 3. *Performance of linear and non-linear classifiers on the RF-selected feature subset. Linear classifiers show the best overall performance, with high accuracies and a good detection rate/accuracy balance. In particular, logistic regression shows the best detection rate performance on this subset, indicating that the subset of features reconstructs an impersonation attack signature efficiently. Interestingly, optimized non-linear classifiers such as the MLP built on this training set show good accuracies (higher than a linear SVC classifier in this instance), but lower detection rates for impersonation attacks, indicating a tendency to label impersonation attacks as normal traffic.*

	Best Results - Accuracy				
	No. of features	**LR**	**SVC**	**RF**	**MLP/ANN**
This report	7	97.52	98.28	87.17	93.09
Seo et al	5		98.22		
Parker et al	3/9/7/3	92.18	93.34	53.39	93.16
Aminanto et al	8		99.97		99.95

	Best Results - Detection Rate				
	No. of features	**LR**	**SVC**	**RF**	**MLP/ANN**
This report	7	99.81	99.26	79.33	89.00
Seo *et al*[1]	5		98.22		
Parker *et al*[2]	3/9/7/3	98.98	98.89	51.76	96.91
Aminanto *et al*[3]	8		99.92		99.88

Table 4. *Comparison to baselines. Best results for each classifier considered, irrespective of the feature subset it was obtained on. The results of this report reinforce the findings of current research that feature selection and linear classification is a highly effective machine learning workflow for the detection of intrusions (in particular impersonation attacks). The relatively low detection rate achieved by non-linear classifiers in this report might be corrected by using different feature subsets, and hyperparameters tuning, to achieve detection rates closer to the standard set by recent literature.*

DISCLAIMER:

Throughout the research work undertaken for this report, I observed variable performances of the Stacked Autoencoder when initiating the training process with different seeds. This is to be expected for neural network based algorithms, due to the weight initialization of the neural network, the random train/test splits and the stochastic nature of the adam optimizer. However, the performance of a logistic regression classifier trained and tested on encoded data (10 features) varies drastically with changing seeds, from 50% to 94% accuracy. Most autoencoders trained at varying seeds give an ulterior logistic regression performance on the encoded data of 80-94% accuracy, but occasionally this drops to 50-60%. After first assuming that this behaviour has to do with the randomness in the test/train splits (as perhaps the autoencoder trains better on some subsets than others), I trained an autoencoder on the whole training set supplied (that underwent the steps described in the data cleaning and pre-processing section). Same variable performance was observed, which indicates to me that this has to do with the neural network architecture. Further research is needed to establish whether increasing the strength of regularization results in more consistent autoencoder outputs, however, for the purpose of this report, I used one of the better performing autoencoders for the feature abstraction steps, to better show the power of this method in producing useful compressed representations of high dimensional data.

REFERENCES:

1. S. J. Lee, P. D. Yoo, A. T. Asyhari, Y. Jhi, L. Chermak, C. Y. Yeun and K. Taha, "IMPACT: Impersonation Attack Detection via Edge Computing Using Deep Autoencoder and Feature Abstraction" in *IEEE Access*, vol. 8, 2020, pp. 65520-65529.

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, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection," in *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 3, Mar. 2018, pp. 621-636.

4. C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset" in *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, 2016, pp. 184-208.

5. J. Brownlee, “Autoencoder Feature Extraction for Classification”, Machine Learning Mastery, https://machinelearningmastery.com/autoencoder-for-classification/, accessed January 15th, 2021.

6. G. James, D. Witten, T. Hastie, R. Tibshirani, “Tree-Based Methods”, in *An Introduction to Statistical Learning with Application in R*, New York, NY, USA: Springer, 2013, ch. 8, pp. 319-320.

7. A. Géron, “Dimensionality Reduction”, in *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, Sebastopol, CA, Canada: O’Reilly, 2019, ch. 8, pp. 222-223.