Data Processing and Model Selection for Machine Learning-based Network Intrusion Detection

Abhijeet Sahu
ECEN Department
Texas A&M University
College Station, USA
abhijeet_ntpc@tamu.edu

Zeyu Mao
ECEN Department
Texas A&M University
College Station, USA
zeyumao2@tamu.edu

Katherine Davis
ECEN Department
Texas A&M University
College Station, USA
katedavis@tamu.edu

Ana E. Goulart
ESET Department
Texas A&M University
College Station, USA
goulart@tamu.edu

***Abstract*—Signature-based Intrusion Detection Systems (IDSes) such as Snort, BRO or Suricata depend on specific patterns and byte sequences in network traffic to detect intrusions; hence, they cannot prevent intrusions for unknown zero-day attacks. Various anomaly-based IDSes that have been proposed based on machine learning (ML) techniques incur high false positives. To overcome this, we explore different types of data processing, i.e. data balancing, feature correlation, normalization, and feature reduction, and whether they are necessary for datasets with different feature dimensions: Coburg Intrusion Detection Data Sets (CIDDS) with five features and Knowledge Discovery and Data Mining (KDD) with 41 features. Further, we perform model selection by comparing the performance of various linear and non-linear classifiers. Generally, our results show that non-linear classifiers outperformed linear ones and that using data balancing and normalization improves the overall accuracy for most classifiers.**

***Index Terms*—Data processing, model selection, data balancing, feature correlation, normalization, feature reduction, linear and non-linear classifier**

I. Introduction

Due to the proliferation of IT infrastructure in diverse sectors that include banking, health care, academics, and industrial networks, organizational groups can communicate through the internet or their secured network. To ensure secured operations, a robust Network Intrusion Detection System (NIDS) is desirable. Though there have been significant developments in NIDS techniques, most firms still rely on conventional signature-based intrusion detection systems such as Snort [1], BRO or Suricata instead of anomaly detection methods. Network security professionals are apprehensive to adopt machine-learning-based anomaly-based IDSes due to the high costs of mis-classification caused by the diverse nature of network traffic. Moreover, existence of a semantic gap in machine-learning-based anomaly intrusion detectors, such as transferring results from anomaly-based detectors to actionable reports for the network operator, makes these techniques less viable [2]. To reduce false positives and negatives in the machine learning (ML) techniques, the authors in [2] recommend to define the threat model, narrow the scope of the detection, and to make use of simplistic techniques that thoroughly explain the behavior of some unique attacks. It has also been observed that training an ML technique with a different dataset and using a real operational system for testing, may result in profusely degraded performance. Moreover, in a normal training dataset, the distribution of normal and intrusion traffic is non-uniform which may affect the classifiers' accuracy [3]. Additionally, there exist a high number features in network traffic, and these need to be pruned to make the classifier fast and accurate. Handling these features is an important practical challenge. Feature filtering, data balancing, and other transformations such as normalization or standardization can improve intrusion detection accuracy. Hence, our current work considers ML techniques for network intrusion detection and the impact of incorporating various data pre-processing methods before applying the classifiers for intrusion detection.

The major contributions of this paper are: a) Evaluate various data processing methods such as data balancing, feature correlation, normalization, and feature reduction for NIDS datasets. b) Compare various linear and non-linear classifiers with data processing for NIDS datasets. c) Analyse the cause of classifier's performance variation and explain using NIDS datasets characteristic.

The paper proceeds as follows. Section II provides a brief introduction on different types of intrusion detection systems and, if built using ML techniques, the performance metric used. In Section III, we discuss the four types of data processing techniques we adopt to improve the accuracy of the classifiers. In Section IV, we discuss various linear and non-linear classifiers with their pros and cons. Finally, in Section VI, we evaluate the performance of the ML techniques with the four types of data transformation, and we conclude the paper in Section VII.

II. Background

Intrusion detection systems are primarily classified into two types, *Network-based IDS* and *Host-based IDS (HIDS)*. Based on their functionality, they are classified into three categories: signature-based, anomaly-based, and hybrid [4]. A HIDS monitors traffic and suspicious activity on a local computer and alerts network admins and Security Information and Event Management Systems (SIEMS). A NIDS monitors network-based traffic and activity. Usually they are deployed just inside the external firewall. *Signature-based detection* looks for patterns of malicious behavior using records of predefined attack signatures and rules. *Anomaly-based detection* identifies deviations from normal network behavior using statistical measures [1]. Hybrid based methods use both pattern matching and statistical measures for detection purpose [4].

Machine learning is relevant when we are considering anomaly-based or hybrid-based IDS. The studies in this paper explore supervised learning techniques such as linear, ensemble, and neural network based classifiers. The classification techniques implemented in this work are evaluated based on predictive accuracy. A classifier's accuracy is computed using

Accepted for publication in IEEE workshop on Communication, Quality and Reliability (IEEE CQR 2020), May 2020

metrics such as *Recall*, *Precision*, and *F1-score*. Recall is the percentage of data rows that belong to an attack type, say A, that are correctly classified as belonging to type A. Precision is the percentage of those instances that truly belong to type A, among all those classified as type A. High precision relates to a low false positive rate. High recall relates to a low false negative rate. False negatives are costly since an undetected attack may escalate more privileges and result in large-scale network breach. False positives cost time and money for security professionals to investigate, and they can erode an organization's trust in the system's results. Hence, a weighted combination of recall and precision, called F1-score, is a preferable metric to provide a more balanced evaluation.

III. Data Processing

A. Data Balancing

Balanced data is generally good for classification [5], however imbalanced datasets are common from an actual production environment and can degrade a classifier's performance [6]. In a given network traffic, the intrusion traffic density will be dependent on the type of attack. For example, a Denial of Service (DoS) based TCP SYN Flood attack will have more SYN traffic compared to normal traffic in a given time range. But generally, the normal traffic would be more than intrusion traffic. Based on the imbalanced distribution of the dataset, the classifier may have the tendency to always predict the majority class and ignore the under-represented class. The impact of imbalanced data on classifiers has been discussed in [7]. There exist different techniques for addressing the issue of class imbalance, including re-sampling the dataset to offset this imbalance. Based on the distribution of the dataset, under-sampling the majority class(es), oversampling the minority class(es), or combining under-sampling and oversampling, can be used.

B. Feature Correlation

A common hypothesis in the machine learning area is that the features are independent. Thus, when datasets have too many correlated features, the classifier performance may vary, and it will not be improved by adding more correlated features (though this depends on the specifics of the problem like the number of variables and the degree of correlation). Generally, this can be viewed as a case of Occam's razor [8], and a simple model with the minimum features is preferable.

The Pearson correlation [9] is widely used to indicate whether two features are linearly correlated, and the Shapiro-Wilk algorithm [10] is useful to assess the normality of the distribution of instances with respect to the feature. These two methods can be utilized before the classifier model selection. Recursive Feature Elimination with Cross-Validation (RFECV) [11] is generally used to remove the unuseful or redundant features for specific classifiers.

C. Effect of Normalization

Feature normalization or scaling matters in ML techniques such as Principal Component Analysis (PCA), Support Vector Machines (SVMs), Multi-Layer Perceptrons (MLPs), etc. Very few techniques, such as Decision Trees, are scale-invariant. Before performing normalization, it is also essential to do some form of log transformation for the features with high variance. Hence, we evaluate both log transformation as well as scaling. We deploy *Min-Max scaling* that scales the numerical features using Eq. 1, where F is the list of continuous features.

$$f_{norm} = \frac{f - f_{min}}{f_{max} - f_{min}} \forall f \in F \qquad (1)$$

Min-Max scaling is only performed on continuous variables. For discrete variables such as source and destination ports, label encoders are used to encode the categorical features into integers. Additionally, we evaluate other types of normalization such as *Z-score normalization*, also called as standardisation, that scales the features to a standard normal distribution with $\mu = 0$ and $\sigma = 1$

D. Feature Reduction

There are two well-known feature dimensionality reduction techniques, Latent Dirichlet Allocation (LDA) and PCA. LDA uses class information to find new features, but since the original datasets have an imbalanced class distribution, we consider PCA for dimension reduction. PCA is a feature dimension reduction technique that transforms data records from higher dimension to a set of successive orthogonal components that accounts for majority of the variance in the original data set. The original data need to be scaled before employing PCA, because scaling affects the covariance, and PCA computes the covariance in one of its steps. PCA has been considered for feature selection and noise removal for network anomaly detection [12], [13].

IV. Classifiers for IDS

A. Linear Classifiers

1) Support Vector Classifier (SVC): The Support Vector classifier is a supervised learning technique that is effective for high dimensional feature space, even for cases with higher feature size than sample size. The classifier performance depends on the type of decision function, or support vectors, defined using the kernel type such as linear or radial basis function. SVCs or their enhanced forms have been predominantly proposed in intrusion detection solutions [14], [15].

2) Stochastic Gradient Descent (SGD) Classifier: SGD uses stochastic gradient descent for learning linear models such as linear SVM, logistic regression and multiple linear regression. SGD is an incremental anytime algorithm, so it can be applied to large-scale and sparse machine learning problems. The major disadvantage of this method is that it is computationally expensive. Additionally, it is sensitive to feature scaling. Many works on NIDS make use of these techniques, such as [16], [17] for intrusions and zero-day attacks, respectively.

3) Logistic Regression (LR) Classifier: Logistic regression is a probability-based classification algorithm which minimizes the error cost using the logistic sigmoid function. It is widely used in the industry because it is very efficient and highly interpretable [18].

B. Non-Linear Classifiers

1) Naive Bayes (NB) Classifier: Naive Bayes Classifier is a supervised learning technique making use of Bayes Theorem, with the assumption of independent features, given the class. There are different forms of the feature likelihood distribution, like Gaussian, Bernoulli etc. The method is computationally efficient, but the selection of feature likelihood may alter result. It is widely used in the area of spam filtering, text classification, and also proposed for NIDS [19].

2) Decision Tree (DT) Classifier: Unlike other classifiers, decision tree functions requires least data transformation such as normalization. It internally creates a model that predicts the target class by learning simple decision rules inferred from the features. To avoid the issue of over-fitting (i.e. learning complex tree that lack generalization), pruning techniques are adopted such as reducing the tree max-depth. The classifier creates biased trees if some classes are dominated. So the performance may be degraded if we have unbalanced data (i.e. different class records are non-uniformly distributed). Due to its simple inferred rule-based modeling and non-parametric learning approach, it has been proposed in [20]–[22]. It performs statistical analysis on specific communication protocols [21], [22] for inferring anomalous behavior.

3) AdaBoost (AB) Classifier: AdaBoost classifier is an iterative ensemble supervised learning technique, that considers the predictions of several weak classifiers on a repeatedly modified versions of data and combine them based on a weighted majority vote. It works by putting more weight on difficult to classify instances and less on those already handled well. This boosting based classifier is used in network security areas and especially for intrusion detection [23], [24].

4) Random Forest (RF) Classifier : Random Forest classifier is an ensemble based classifier in which a diverse collection of classifiers (decision trees) are constructed by incorporating randomness in tree construction. The use of randomness is to decrease the variance i.e. the high variance and overfit issues in DT. While comparing with SVMs, RF is fast and works well with a mixture of numerical and categorical features. Due to its variance reduction feature it is also used in security domain [25], [26].

5) Neural Network (NN) Classifier: Neural Network works well for highly complex non-linear models. In the intrusion detection classification problem, we make use of multi-layer perceptron as the supervised learning algorithm. It basically learns a non-linear function approximator whose inputs are the features for a record and outputs the class. Unlike a simple logistic regressor, it can have multiple hidden layers. The major issue with neural network models is there are huge set of hyper tuning parameter such as number of hidden neurons, layers and iterations, dropouts, etc that can alter the validation accuracy. Moreover, it is quite sensitive to feature scaling. Following Occam's razor, most of the security professional would prevent use of neural network for intrusion detection unless the machine learning techniques fails.

V. Dataset

A. KDD

The Knowledge Discovery in Databases (KDD) cup was an International Knowledge Discovery and Data Mining Tools competition that was held in 1999, with the purpose of collecting network traffic records to build a network intrusion detector, that can classify *bad* traffic from *good* traffic. The steps involved in the competition were data preparation, feature selection, data cleaning, incorporation of suitable prior knowledge from domain expert and elucidation of the results from data mining [4]. The dataset consists of 43 features per record with 41 features related to the traffic input such as source and destination IP address and port numbers. The last two feature being the *label* and *score*. The dataset broadly labels the attack types into four categories. Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L). Hence there are 5 classes including normal traffic. *DoS* attacks block the resources of the target system by creating congestion in the target network. *Probe* being the attack type were the attacker tries to collect information such as scanning open ports. *U2R* are privilege escalation types attack to gain root access from a user access. *R2L* attack involves unauthorized access to from a remote machine such as FTP write, where the attacks exploits a mis-configuration affecting write privileges of anonymous records on a FTP server. This dataset is widely utilized by many ML/DL researchers to validate the accuracy of their detection algorithm. For example, authors make use of KDD to design IDS based on neural networks, Support Vector Machines (SVM) and Ada-Boost [27], [28].

B. CIDDS

The CIDDS was created for anomaly based network IDS. The CIDDS contains two datasets. For the first dataset they created labelled flow-based data sets in an emulated small business environment using OpenStack. The emulated environment included e-mail and web servers, and all the traffic generated by the clients, managers and employees following a specific pattern. It contains four weeks of unidirectional flow-based network traffic. The data set encompasses an external server which was attacked in the internet [29]. For generating malicious traffic, Denial of Service (DoS), Brute Force attacks and Port Scans were executed within the network. The second dataset is a port scan dataset containing two weeks of unidirectional flow-based network traffic with more normal user behavior and different port scan attacks [29]. We have used the first dataset for our simulations. The dataset originally contains 12 features i.e. duration, protocol, source and destination IP address and ports, packets and bytes count in that duration, flow count, Type of Service and class. For our simulation study we considered duration, source and destination port numbers, packet count, flow count features. There are three categories in the class field: normal, suspicious and unknown.

VI. Results and Analysis

A. Imbalanced Data

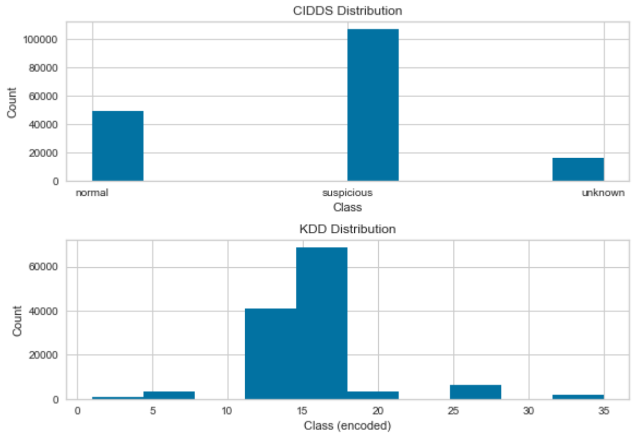


Fig. 1. Imbalanced distribution of CIDDS and KDD

The distribution of the CIDDS and the KDD dataset is shown in Fig.1. The two datasets are imbalanced in classes, and the distribution of the KDD is even less uniform: the two most dominant classes both have more than 40,000 instances while the least dominant 16 classes have less than 1,000 instances. Two types of data balancing methods are used: weight method is used for most classifiers, while the NearMiss algorithm [30] is used to down-sample the datasets for the NN.

For the CIDDS dataset, DT and RF achieve 100% accuracy in the imbalanced and balanced dataset. SVC performs similar after data balancing, while the LR's performance downgrades in the balanced dataset since the weight method largely increases the weight of the class "unknown", and the less-uniform features of this class affect this distance-based regression's accuracy. The balanced dataset dramatically improves the predictive accuracy of NN, as shown in Table I. This probability-based classifier is sensitive to the distribution of classes, thus in the result for the imbalanced dataset, the recall for the class "unknown" is very low due to its low percentage in all classes. After the data balancing, the recall for class "unknown" increases from 9% to 98%.

Class	*Imbalanced*			*Balanced*		
	Prec	Rec	F1	Prec	Rec	F1
Normal	0.83	0.50	0.62	0.98	0.98	0.98
Suspicious	0.72	0.95	0.82	0.95	0.67	0.78
Unknown	0.94	0.09	0.16	0.76	0.98	0.86

TABLE I
NN PERFORMANCE IN CIDDS

For the KDD dataset, DT, RF and SVC showed slight improvement (less than 1%) after the data balancing, while LR's performance downgrades heavily from 62% to 1%. This is due to the same reason mentioned above - the less-uniform KDD dataset makes LR using these weighted sparse classes even less accurate. Since there are 9 classes that have less than 20 instances, the down-sampling cannot be done for the dominant classes, thus there is no comparison for NN.

B. Feature correlation

To verify the impact of strongly correlated features on the classifiers, we first check the Pearson's correlation, shown in Fig. 2. We picked 4 pairs of strongly correlated features from the Pearson result (from dark color bins in the figure), with 2 relatively independent features ("flag" and "is host _login") to test the performance of classifiers, compared to the result by removing one of the feature from pairs. DT, RF, SVC, LR and NN all performs at the same level (difference less than 2%), which means removing part of these strongly correlated features will not affect the performance, which can be computationally useful for high dimensional dataset.

In Fig. 3 we use the Shapiro ranking method to find the distribution of each features. The high value indicates the feature follows a Gaussian distribution. We also use the RFECV to find the minimum and the most important features that preserve the predictive accuracy. These results are used as a reference for further normalization and feature reduction.

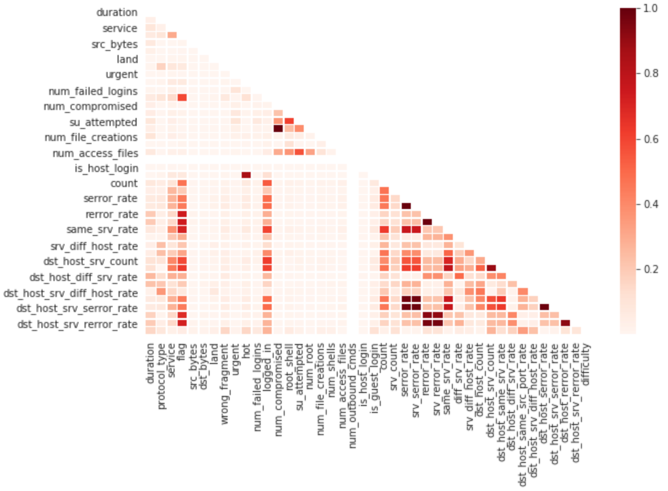


Fig. 2. Pearson correlation of KDD features

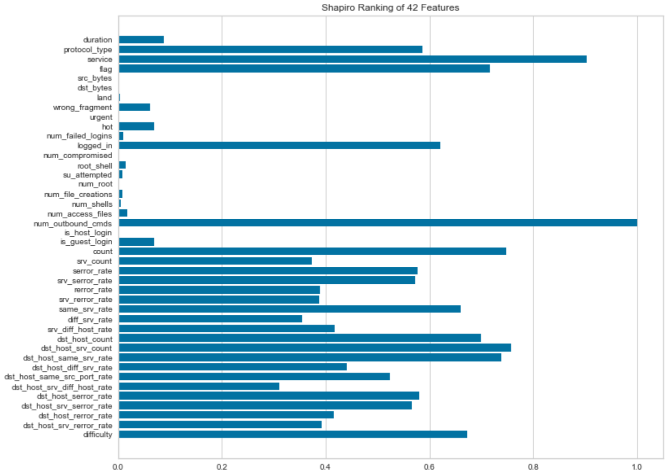


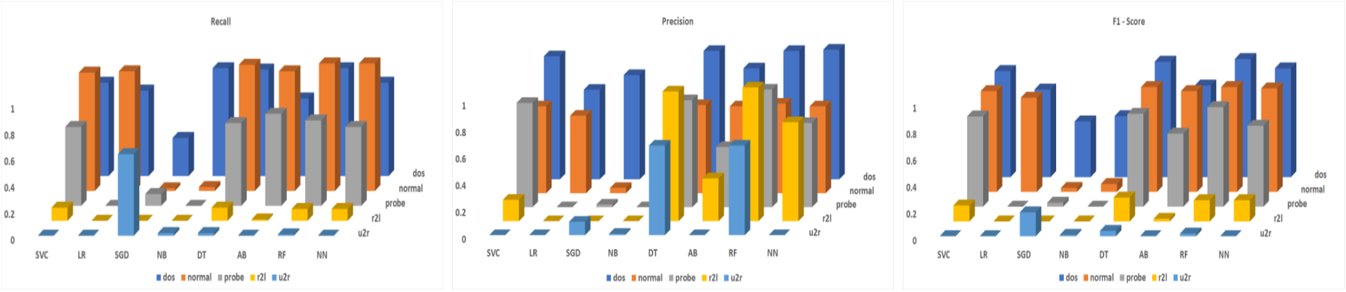
Fig. 3. Shapiro ranking of KDD features

C. Effect of normalization

With feature normalization for KDD dataset, the precision for the linear techniques such as SVC, LR and SGD increased for U2R and Probe class of attacks (Fig. 4(b) and Fig. 5(b)). For the ensemble techniques, such as DT, AB and RF, the performance were not effected much by normalization as they are not distance-based. With normalization, the Recall for LR, SGD and NB classifier improved (Fig. 4(a) and Fig. 5(a)). Normalization, still could not improve the recall for U2R and L2R attacks since they are scarce in the dataset. Normalization didn't improve NN classifier because ReLU [31] is used as activation function in the hidden layers. Data imbalance in attack types seems to be one of the probable reason for low recall even after normalization. We also tested the Z-score normalization, which is better for outlier detection, but such normalization degraded the performance of the classifier. Since, the features does not follow a normal distribution hence, z-score normalization is not a viable option for scaling. Using Shapiro ranking, we can select features with normal distribution and utilize z-score normalization. Normalization for CIDDS dataset, we did not observe much improvement since the number of continuous features were less. Classification using normalized dataset were observed to be quite faster especially in distance-based linear classifier.

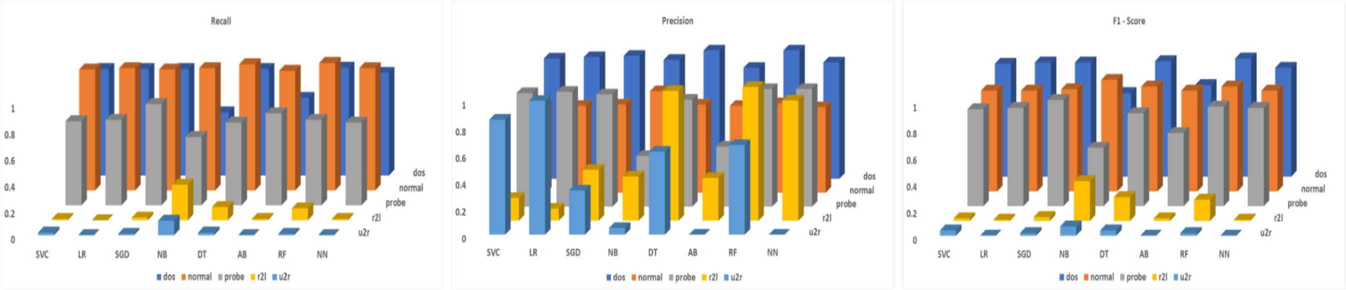
D. Impact of feature Reduction

In the CIDDS dataset the five features which we considered were reduced to two dimension using PCA. Similarly for the KDD dataset we reduced from 41 to ten dimensions. Fig. 6



(a) (b) (c)

Fig. 4. (a) Recall; (b) Precision; (c) F1 score for KDD Dataset **without normalization**



(a) (b) (c)

Fig. 5. (a) Recall; (b) Precision; (c) F1 score for KDD Dataset with **normalization**

shows the projection of features to the dominant orthogonal space. In most cases, the performance of the classifier reduced when non-normalized features were used as observed from Fig. 7. With normalization we improved the performance of the classifiers with the PCA reduced features (Fig. 8). Further, we utilized the results from Shapiro ranking, to select 20 top-ranked features and normalize them. Additionally, we reduced the dimension of these 20 normalized features with PCA, to further improve the classifier's performance as observed from Fig. 9. It can be observed that *DoS*, *Normal* and *Probe* classes had their recall values increased for both linear and non-linear classifiers. Only the performance of the Naive Bayes classifier degraded for the *R2L* type intrusions.

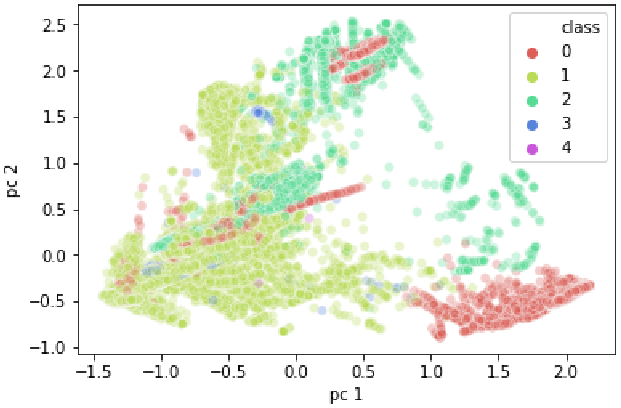


Fig. 6. PCA transformation on KDD dataset with classes 0 (DoS), 1 (Normal), 2(Probe), 3 (R2L) and 4 (U2R)

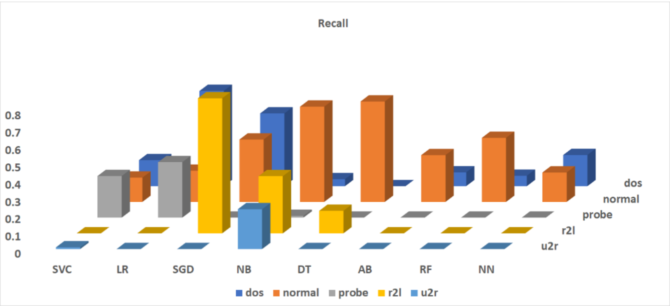


Fig. 7. Recall **without normalization** and **PCA feature reduction**

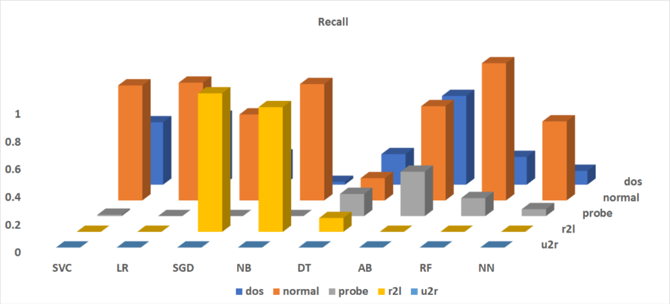


Fig. 8. Recall with **normalization** and **PCA feature reduction**

VII. Conclusion

Logistic regression is one of the most common classifier used in the industry for real-time operational application, however from the experiment we can conclude that the data

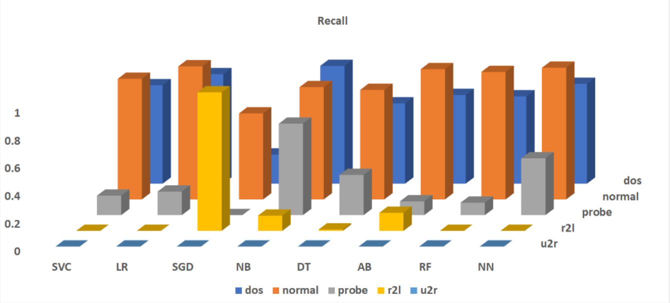


Fig. 9. Recall after **feature selection** with **normalization** and **PCA**

balancing cannot improve its performance if the training dataset is less uniform and the least dominant features are sparse. Data balancing will benefit the neural network if improving the predictive accuracy of less dominant classes is desired. However, neural network is not the must go for the best classifier for NIDS. Feature correlation helps to remove some correlations among features which will not affect the performance, but can be computationally beneficial for large scale networks with high dimensional feature. From the experiments, most of the linear classifier were outperformed by non-linear classifier, but with the use of normalization, we observed improvement in linear case. This is due to existence of many discrete features in the KDD dataset, making the linear models hard to estimate. Additionally, feature ranking assisted us in selecting features, that will benefit more by normalization, followed by feature reduction to improve the accuracy even in linear case.

Acknowledgment

This research is supported by the US Department of Energy's (DoE) Cybersecurity for Energy Delivery Systems program under award DE-OE0000895.

References

[1] R. Berthier, W. H. Sanders and H. Khurana, "Intrusion Detection for Advanced Metering Infrastructures: Requirements and Architectural Directions," 2010 First IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, 2010, pp. 350-355.

[2] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," 2010 IEEE Symposium on Security and Privacy, Berkeley, CA, 2010, pp. 305-316.

[3] H. Yin, K. Gai and Z. Wang, "A Classification Algorithm Based on Ensemble Feature Selections for Imbalanced-Class Dataset," 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), New York, NY, 2016, pp. 245-249.

[4] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," in IEEE Communications Surveys Tutorials, vol. 18, no. 2, pp. 1153-1176, Secondquarter 2016.

[5] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. 2004. A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. 6, 1 (June 2004), 20–29.

[6] Sun, Y., Wong, A.K. and Kamel, M.S., 2009. Classification of imbalanced data: A review. International journal of pattern recognition and artificial intelligence, 23(04), pp.687-719.

[7] Chawla, N.V., Japkowicz, N. and Kotcz, A., 2004. Special issue on learning from imbalanced data sets. ACM SIGKDD explorations newsletter, 6(1), pp.1-6.

[8] Blumer, A., Ehrenfeucht, A., Haussler, D. and Warmuth, M.K., 1990. Occam's razor. Readings in machine learning, pp.201-204

[9] Karl Pearson (20 June 1895) "Notes on regression and inheritance in the case of two parents," Proceedings of the Royal Society of London, 58 : 240–242.

[10] SHAPIRO, S. S., WILK, M. B. (1965). An analysis of variance test for normality (complete samples). Biometrika, 52(3–4), 591–611. https://doi.org/10.1093/biomet/52.3-4.591

[11] Guyon, I., Weston, J., Barnhill, S., Vapnik, V., "Gene selection for cancer classification using support vector machines", Mach. Learn., 46(1-3), 389–422, 2002.

[12] Eduardo De la Hoz, Emiro De La Hoz, Andrés Ortiz, Julio Ortega, Beatriz Prieto, "PCA filtering and probabilistic SOM for network intrusion detection", Neurocomputing, Volume 164, 2015, Pages 71-81

[13] Y. Lee, Y. Yeh and Y. F. Wang, "Anomaly Detection via Online Oversampling Principal Component Analysis," in IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 7, pp. 1460-1470, July 2013. doi: 10.1109/TKDE.2012.99

[14] Khan, Latifur Awad, M. Thuraisingham, Bhavani. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. VLDB J.. 16. 507-521. 10.1007/s00778-006-0002-5.

[15] Mulay, Snehal Devale, P.R. Garje, Goraksh. (2010). Intrusion Detection System Using Support Vector Machine and Decision Tree. International Journal of Computer Applications. 3. 10.5120/758-993.

[16] Kang MJ, Kang JW (2016) Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security. PLOS ONE 11(6): e0155781. https://doi.org/10.1371/journal.pone.0155781

[17] A. G. P. Lobato, M. A. Lopez, I. J. Sanz, A. A. Cardenas, O. C. M. B. Duarte and G. Pujolle, "An Adaptive Real-Time Architecture for Zero-Day Threat Detection," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-6.

[18] Yun Wang. 2005. A multinomial logistic regression modeling approach for anomaly intrusion detection. Comput. Secur. 24, 8 (November 2005), 662–674. DOI:https://doi.org/10.1016/j.cose.2005.05.003

[19] Saurabh Mukherjee, Neelam Sharma, "Intrusion Detection using Naive Bayes Classifier with Feature Reduction", Procedia Technology,Volume 4,2012,Pages 119-128

[20] Stein, Gary Chen, Bing Wu, Annie Hua, Kien. (2005). Decision tree classifier for network intrusion detection with GA-based feature selection. ACM Southeast Regional Conference Proceedings of the 43rd annual Southeast regional conference. 136-141.

[21] T. Abbes, A. Bouhoula and M. Rusinowitch, "Protocol analysis in intrusion detection using decision tree," International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004., Las Vegas, NV, USA, 2004, pp. 404-408 Vol.1. doi: 10.1109/ITCC.2004.1286488

[22] N. Moustafa, B. Turnbull and K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4815-4830, June 2019.

[23] Hu, Weiming Hu, Wei Maybank, Steve. (2008). AdaBoost-Based Algorithm for Network Intrusion Detection. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society.

[24] Mehrnaz Mazini, Babak Shirazi, Iraj Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms", Journal of King Saud University - Computer and Information Sciences, Volume 31, Issue 4, 2019, Pages 541-553

[25] Nabila Farnaaz, M.A. Jabbar, "Random Forest Modeling for Network Intrusion Detection System ", Procedia Computer Science, Volume 89, 2016, Pages 213-217, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2016.06.047.

[26] J. Zhang, M. Zulkernine and A. Haque, "Random-Forests-Based Network Intrusion Detection Systems," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 5, pp. 649-659, Sept. 2008. doi: 10.1109/TSMCC.2008.923876

[27] S. Mukkamala, G. Janoski and A. Sung, "Intrusion detection using neural networks and support vector machines," Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), Honolulu, HI, USA, 2002, pp. 1702-1707 vol.2.

[28] W. Hu, W. Hu and S. Maybank, "AdaBoost-Based Algorithm for Network Intrusion Detection," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 38, pp. 577-583, April 2008.

[29] Ring, Markus et al. "A Survey of Network-Based Intrusion Detection Data Sets." Computers Security 86 (2019): 147–167. Crossref. Web.

[30] Mani, I. and Zhang, I., 2003, August. kNN approach to unbalanced data distributions: a case study involving information extraction. In Proceedings of workshop on learning from imbalanced datasets.

[31] Glorot, X., Bordes, A., Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In Proceedings of the fourteenth international conference on artificial intelligence and statistics (pp. 315-323).