Towards Data-Driven Network Intrusion Detection Systems: Features Dimensionality Reduction and Machine Learning

https://doi.org/10.3991/ijim.v16i14.30197

Majdi Maabreh[1(✉)], Ibrahim Obeidat[1], Esraa Abu Elsoud[2], Asma Alnajjar[2], Rahaf Alzyoud[2], Omar Darwish[3]
[1]Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, P.O. Box 330127, Zarqa 13133, Jordan
[2]Department of Computer Information Systems, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, Zarqa, Jordan
[3]Information Security and Applied Computing Department, Eastern Michigan University, MI, USA
majdi@hu.edu.jo

Abstract—Cyber attacks have increased in tandem with the exponential expansion of computer networks and network applications throughout the world. Fortunately, various machine/deep learning models have demonstrated excellent accuracy in predicting network attacks in the literature; nonetheless, having simple and understandable models might be a big benefit in network monitoring systems. In this study, we evaluate four feature selection algorithms to find the minimal set of predictive features of network attacks, seven classical machine learning algorithms, and the deep learning algorithm on one million random instances of the CSE-CIC-IDS2018 big data set for network intrusions. The feature selection algorithms highlighted the importance of features related to forwarding direction (FWD) and two flow measures (FLOW) in predicting the binary traffic type; benign or attack. Furthermore, the results revealed that not all features are required to build efficient ML/DL in detecting network attacks, four features unanimously selected by the feature selection algorithms were enough to build comparable ML models to those trained on all features. This might lead to models that are more suitable for deployment in terms of complexity, explainability, and scalability. Moreover, by selecting four unanimity features instead of all traffic features, the training time may be decreased by 10% to 50%.

Keywords—intrusion detection, machine learning, feature selection, big data, deep learning, network security

1 Introduction

The reliance of countries on computer networks, particularly in emergencies like the Covid-19 epidemic, the requirement for mobile learning as a result of technological advancements [1, 2], and/or smart systems or campus applications [3] highlighted the significance of increased efforts in network security challenges, so that information system security becomes one of the most critical topics in industry and academics. One of the research areas is to develop a robust intrusion detection system (IDS) which is a

hardware or software program that watches for harmful activity or policy breaches on a network [4]. IDS systems can be Network-based Intrusion Detection Systems (NIDS) which detect violations using traffic data, and/or Host-based Intrusion Detection Systems (HIDS) which detect violations using host data. Signature-based intrusion detection (SIDS) and anomaly detection-based intrusion detection (AIDS) are two types of IDS detection. SIDS, for example, is built on the concept of establishing signatures for attack modes, and thus has high detection effectiveness for known attacks since signatures are available. This approach, on the other hand, could struggle in identifying new attacks. The concept of "AIDS" is based on the idea of precisely outlining the features of normal activity, and any deviation from that normal characteristic is deemed abnormal behavior. (AIDS) has the benefit of being able to identify new and undiscovered attacks. However, because it is difficult to distinguish between false and true alarms, it has a high False Alarm Rate (FAR) [5].

The researchers investigated the power of machine learning (ML) and deep learning (DL) approaches to have smart; learning meaningful knowledge from large amounts of data, and reliable IDS [6, 7]. However, AI-based solutions, in general, need to be built carefully since several factors should be taken into account when generalizing a model for IDS [8]. For example, but not limited to, the amount of data used for training a model, the performance metrics recorded on unseen testing data show no overfitting, training time, model complexity, the learning algorithms, dataset features, and/or the computation environments and the resources needed for deployment. Fortunately, several machine/deep learning models in the literature showed high accuracy in predicting network attacks [9], however, in this study, we focus on building efficient models with the same or higher accuracy while needing fewer data features. Feature selection algorithms could be used to identify irrelevant, unneeded, or even redundant feature set that either do not contribute to the model's accuracy or may reduce the model's accuracy. ML/DL model complexity and explainability are important factors to be taken into account while using these models to monitor the network traffic. Fewer features are preferable since they lower the model's complexity, and a simpler model is easier to comprehend and explain [10]. In addition, fewer features could be an advantage to mitigate the volume of data needed to be collected and consulting the ML/DL models about its nature; benign or attack.

Therefore, in this study, we randomly use one million of the recent dataset for IDS; CSE-CIC-IDS2018, and evaluate different learning algorithms along with several feature selection algorithms. We reported all the required metrics for performance evaluation; accuracy, precision, recall, AUC, and F1-score. We also recorded the training time for all combinations as a possible guide for more effort in this regard in both industry and academia.

The remainder of this work is organized as follows; related work is presented in Section 2. The details about the benchmark public datasets, learning algorithms, features selection algorithms, and evaluation metrics are illustrated in Section 3. Section 4 shows the results and a thorough discussion of the findings, followed by conclusions and future work in Section 5.

2 Related work

Anomaly detection is a technique to detect intrusions by learning the characteristics of normal activity, Then designing the systems to detect anything that deviates

from normal activity [11]. Many researchers have turned to use feature selection which is one of the effective ways that used to improve efficiency in the training models. Fitni and Ramli [12] applied ML algorithms on CSE-CIC-IDS2018 after reprocessing the datasets by removing the missing values they used the whole data 16,232,943 instances, the dataset was divided into 80% for training and 20% for testing. The models are built using a set of features selected by Spearman's rank correlation coefficient and Chi-squared test. The results showed that only 23 of the 80 features can be used to record accuracy of 98% using features of Spearman's rank correlation coefficient, and 90% using features selected by Chi-squared, while the accuracy was 93% when using all Features. In trying to study the effects of under-sampling and feature selection on ML models, the results in [13] showed that the accuracy has never affected when the number of benign samples was around 3 million, while the accuracy has dropped when the number of benign samples was around 1 million; from around 98.5% to around 97.7%. To evaluate the effect of feature selection, a random forest (RF) is used to evaluate all features in the raw dataset. Results showed only 54 features were effective, 10 features of them were selected as top N-features based on their information gain, and the accuracy is still 98.37% after selected features of datasets divided as 70% for training and 30% for testing. Using training data of 4,920,094 instances with top 10 features, the accuracy was 98% for RF and convolutional neural networks (CNN), 92% for support vector machine (SVM), and 84% for Naive Bayes (NB).

Kim et al. [14] used the CSE-CIC-IDS2018 dataset and selected DoS to train CNN; binary and multi-class classification. They used 11,000,000 benign and DoS instances divided into 70% training and 30% for testing. The CNN model showed 91.5% of accuracy on average. D'hooge et al. [15] have evaluated 12 learning algorithms and the results reported the tree-based classifiers performed better than others where the accuracy was about 99%.

In [16] the results for multiclass classification were quite similar for the dataset before and after reprocessing. The accuracy was 94% for logistic regression (LR) before reprocessing and 98% after reprocessing, and 99% for decision tree (DT) after and before processing the dataset. As for binary classification, the accuracy results show enhancement after reprocessing the datasets.

Joffrey et al. [17] used the CSE-CIC-IDS2018 dataset to attempt whether the feature selection affects the performance of classifiers in terms of the area under the curve (AUC) and F1-score. After they applied 7 ranking techniques to generate 7 feature lists using the Python libraries. The results show that the classifiers perform as well or better than they do when trained and tested with all available features. These results suggest that feature selection techniques should be used with classifiers to detect anomalies in CSE-CIC-IDS2018 data, as training the model with a reduced feature set uses less computational resources. Another recent study also studied the features selection on the performance of ML models, results showed that random forest and decision trees outperform other algorithms evaluated in [18]. In [19], CSE-CIC-IDS2018 Features were also selected by random-forest where the results showed high accuracy and a low false-positive rate. Feature selection, data balance techniques, and feature processing have improved the performance of classical machine learning algorithms on CSE-CIC-IDS2018 [20–22]. Several studies evaluated the ML algorithm on the CSE-CIC-IDS2018 dataset with features preprocessing and reported the decision tree, random forest, and KNN are more efficient in predicting the attacks [23, 24].

3 Methodology

3.1 Dataset

The most current intrusion detection dataset, CSE-CIC-IDS2018, is huge data, freely available, and covers a wide spectrum of attack types. The full dataset is available on the Amazon cloud by the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC). This dataset consists of 16,233,002 instances captured during 10 days of network traffic and the instances are distributed over ten downloadable CSV files. Besides the benign instances (normal traffic), there are six types of attacks; Denial of Service (DoS), Distributed Denial of Service (DDoS), brute force, infiltration, botnet, and web attack [25]. Fortuentlay, several experiments in the literature reported high accuracy for different ML/DL models using this dataset where the number of used features ranges from 10 to all available features [9]. As a result, the primary purpose of this research is to investigate the nature of key features and determine the smallest number of them required to produce ML/DL models that are equivalent to those trained on all features and stated to have high accuracy. To serve the purpose of this study, we randomly extract one million instances from the whole dataset where all types of attacks are represented in the sample.

3.2 Dataset preprocessing

There are 79 feature columns and one label column in the dataset. When analyzing feature selection methods and learning algorithm performance, we focus on network traffic characteristics, or Behavioural metrics [18]. As a result, the "Dst Port" "Protocol" and "Timestamp" columns are eliminated from the dataset. We also discovered eight columns with zero values in them. These columns, which have no value on the label, have been removed: "Fwd Byts/b Avg", "Fwd Pkts/b Avg", "Fwd Blk Rate Avg", "Bwd Byts/b Avg", "Bwd Pkts/b Avg", "Bwd Blk Rate Avg", "Bwd PSH Flags", "Bwd URG Flags". In the selected dataset, we also discovered 2512 missing values and 5676 infinite values. All instances of data with missing values or infinity have been eliminated as well [26]. Furthermore, we concentrate on the relevance of the features and the effectiveness of the ML algorithms on the dataset. All attacks' labels are combined into a single label, "Attack," converting the issue from multi-class classification to binary class classification. There are 500,000 benign or regular traffic instances and 500,000 attack cases in the label column. The data has not been transformed, and all features utilized in the subsequent procedure are in their original values. The dataset was divided into a training dataset (70%) and an isolated unseen testing dataset (30%).

3.3 Feature selection algorithms

The act of locating and selecting the most valuable features in a dataset, known as feature selection, is an important stage in the machine learning pipeline. Unnecessary features slow down training, reduce model interpretability, and, most critically, reduce test set generalization performance [27]. Four different feature selection algorithms have been used and evaluated to reduce the feature dimension of the above dataset.

First, Pearson correlation returns a result indicating the degree of correlation between any two variables by dividing the covariance of two variables by the product of their standard deviations.

Second, The two-way chi-square test is a statistical technique for determining how closely anticipated values match actual outcomes. Variables are assumed to be random and taken from a sufficient sample of independent variables in this technique. The chi-squared statistic results show how far the results differ from the predicted (random) outcome.

Third, Random Forest Built-in Mean Decrease Accuracy is a technique for calculating the relevance of features on permuted out-of-bag (OOB) samples based on the mean decrease of accuracy.

Fourth, Features importance in Deep Learning finds the relevance of features by identifying an ideal feature subset that maximizes the performance of a deep neural network and concurrently rating the importance of all features in this optimal subset.

3.4 Learning algorithms and evaluation metrics

In this work, the above feature selection strategies were used to compare the performance of eight common machine/deep learning algorithms for binary classification, thus every classifier performance was used to evaluate the goodness of the features produced by each selection method. Namely, Support Vector Machine (SVM) where the number of iterations is 10, lambda = 0.001, and single parameter trainer mode, Naïve Bays (NB) of both Gaussian and Bernoulli, Decision Trees (DT) where the quality of a split is measured by 'Gini' and max_depth = None, K-nearest neighbors (KNN) where k = 3, Logistic Regression (LR) where solver = 'lbfgs' and max_iter = 2000, Adaboost where n_estimators = 50 and learning_rate = 1, Random Forest (RF) where n_estimators = 100, criterion = 'gini', max_depth = None, and Deep Learning (DL) using 2 layers of 200 neurons each. Due to the observations on the performance issue, we evaluate the SVM using the implementation of Microsoft Azure; a two-class support vector machine. The deep learning models have been developed by the popular h2o.ai. The others were created using Scikit-learn on python. Any learning algorithm does not go through an optimization phase. For ML/DL model evaluation, five popular evaluation metrics have been used; accuracy, precision, recall, F1-score, and the area under the curve (AUC).

4 Results and discussion

There are sixty-eight features of network traffic available after dataset preprocessing. The results of four different feature selection algorithms revealed that not all features have a considerable impact on identifying the binary label (Benign, Attack). Figure 1 shows the top thirty features ranked by their importance; from the most important to the least important, in order of significance to the response variable. According to the Pearson correlation and chi-squared test, features ranked after thirty may have little impact on identifying the response label. Following that, The results show that not all of them are necessary to be used to build successful ML models.

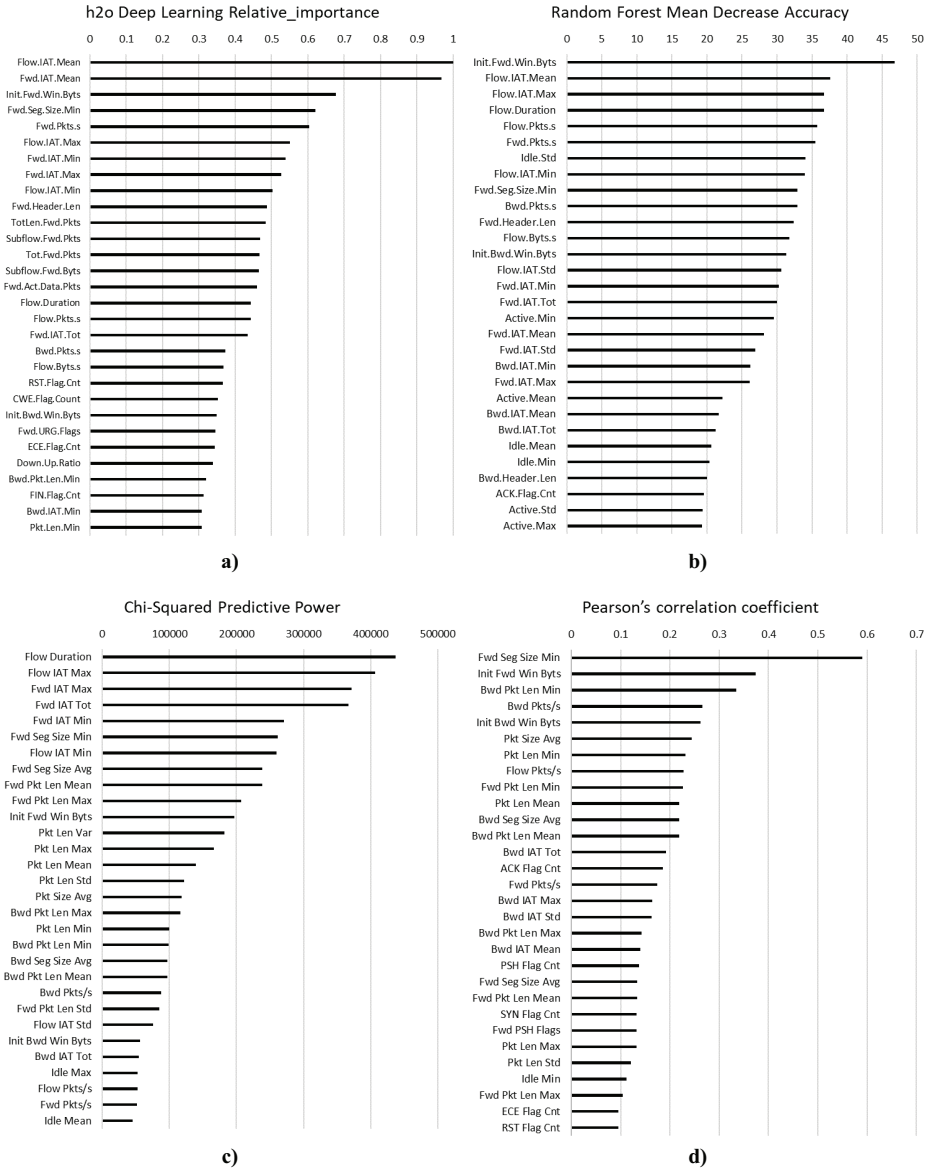


Fig. 1. Top 30 features selected by the feature selection algorithms; (a) Features importance in h2o Deep Learning algorithm, (b) Random Forest Built-in Mean Decrease Accuracy, (c) The two-way chi-square test, and (d) Pearson correlation

The feature sets ranked by deep learning, random forest, and chi-squared highlighted the importance of the statistics of two flows "FLOW IAT" and two flows in the forwarding direction "FWD IAT"; min, max, and avg, since they have been ranked in the top features. Pearson correlation reported the importance of some backward direction "BWD" and packets "Pkt" features in the top features. The chi-squared test prioritized the relevance of features based on their predictive power given the labels, where the predictive power drops to almost half of the ranked first feature (i.e. Flow Duration) starting with the tenth feature (i.e. Fwd Pkt Len Max). A similar scenario applies to h2o deep learning feature selection, where the relative value of features drops to about half that of the ranked first feature (i.e. Flow IAT Mean) after the tenth feature (i.e. Fwd Header Len). Using each feature selection algorithm's top ten features as a starting point, the union of them displays twenty-four network traffic features of certain measures; forward and backward directions, packet information, and flow features. The twenty-four features are {*Bwd Pkt Len Min, Flow Pkts/s, Fwd Pkt Len Max, Bwd Pkts/s, Fwd Header Len, Fwd Pkt Len Mean, Flow Duration, Flow IAT Max, Flow IAT Mean, Flow IAT Min, Fwd IAT Max, Fwd Pkt Len Min, Fwd IAT Mean, Fwd Pkts/s, Fwd IAT Min, Fwd IAT Tot, Fwd Seg Size Avg, Fwd Seg Size Min, Idle Std, Init Bwd Win Byts, Init Fwd Win Byts, Pkt Len Mean, Pkt Len Min, Pkt Size Avg*}. The list of twenty-four features shows seventeen features of them related to "forward" and "flow" measures. This finding could be of value and probably reduces the required information to be collected from the network for machine learning model development, which might also reduce the computational overhead of collectors. Furthermore, the result of the intersection of the top thirty features (a.k.a unanimous voting) shows only 6 features overlap among the four feature selectors; {*Flow Pkts/s, Init Fwd Win Byts, Fwd Seg Size Min, Bwd Pkts/s, Fwd Pkts/s, Init Bwd Win Byts*}. Interestingly, similar to the list of twenty-four features above, the intersection list (six features) shows four out of six features related to "forward" and "flow" measures, namely; {*Flow Pkts/s, Init Fwd Win Byts, Fwd Seg Size Min, Fwd Pkts/s*}, and two of them related to backward direction; namely {*Bwd Pkts/s, Init Bwd Win Byts*}. This observation encourages conducting more experiments to evaluate the learning algorithms not only on the top thirty features but also on the six and four unanimous features.

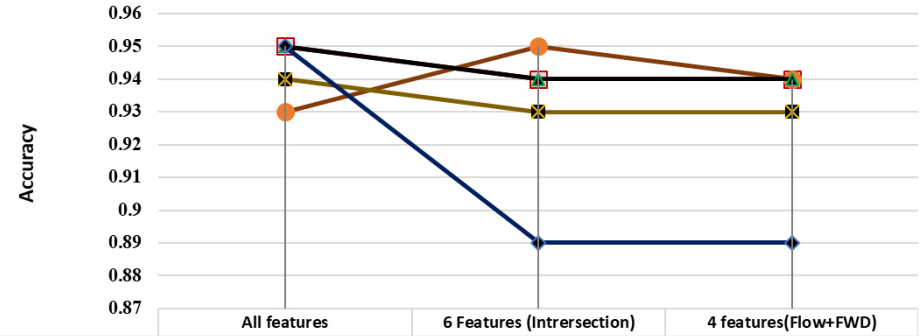
Table 1 shows the performance of the learning algorithms on all feature sets. Five performance metrics have been calculated for each learning algorithm and feature set on an unseen testing dataset; namely Accuracy (Acc), Precision (P), Recall (R), F1-score (F1), and Area Under the Curve (AUC). Learning algorithms may build models with the top thirty significant features that are equivalent in performance to models built with sixty-eight features, according to the performance criteria in Table 1. This observation backs with the feature selectors' claims about the quality of the features they choose.

Table 1. The performance of the learning algorithms on all feature sets

Feature Selection Method	Support Vector Machine					Random Forest				
	Acc.	**P**	**R**	**F1**	**AUC**	**Acc.**	**P**	**R**	**F1**	**AUC**
ALL Features	0.88	0.90	0.85	0.87	0.93	0.95	0.92	0.98	0.95	0.94
DLF	0.87	0.90	0.84	0.87	0.93	0.95	0.92	0.97	0.94	0.94
Chi-squared	0.87	0.89	0.86	0.87	0.93	0.95	0.92	0.98	0.95	0.95
Pearson Correlation	0.85	0.90	0.78	0.84	0.94	0.94	0.92	0.97	0.95	0.95
Random Forest	0.86	0.88	0.83	0.85	0.93	0.95	0.92	0.98	0.95	0.95
6 Features (Intersection)	0.83	0.77	0.94	0.85	0.89	0.94	0.92	0.97	0.94	0.95
4 Features (flow+fwd)	0.82	0.76	0.93	0.84	0.88	0.94	0.92	0.96	0.94	0.94
	NB – Bernoulli					**NB – Gaussian**				
ALL Features	0.68	0.66	0.73	0.69	0.68	0.57	0.92	0.15	0.27	0.57
DLF	0.69	0.67	0.75	0.71	0.70	0.56	0.93	0.14	0.25	0.56
Chi-squared	0.68	0.67	0.73	0.70	0.68	0.56	0.90	0.15	0.25	0.57
Pearson Correlation	0.68	0.67	0.73	0.70	0.68	0.63	0.92	0.29	0.44	0.63
Random Forest	0.67	0.66	0.70	0.68	0.67	0.57	0.92	0.16	0.26	0.57
6 Features (Intersection)	0.66	0.90	0.35	0.51	0.65	0.73	0.66	0.93	0.77	0.73
4 Features (flow+fwd)	0.65	0.90	0.35	0.50	0.65	0.71	0.64	0.92	0.76	0.71
	Decision Trees					**KNN**				
ALL Features	0.93	0.93	0.93	0.93	0.93	0.95	0.92	0.97	0.95	0.94
DLF	0.93	0.92	0.93	0.93	0.93	0.95	0.92	0.97	0.95	0.94
Chi-squared	0.93	0.93	0.93	0.93	0.93	0.94	0.92	0.97	0.94	0.94
Pearson Correlation	0.93	0.93	0.93	0.93	0.93	0.94	0.92	0.97	0.95	0.94
Random Forest	0.95	0.92	0.99	0.96	0.97	0.94	0.92	0.97	0.95	0.94
6 Features (Intersection)	0.95	0.92	0.99	0.96	0.97	0.94	0.92	0.97	0.94	0.94
4 Features (flow+fwd)	0.94	0.92	0.95	0.93	0.94	0.94	0.92	0.97	0.94	0.94
	Logistic Regression					**Adaboost**				
ALL Features	0.87	0.91	0.82	0.86	0.87	0.94	0.91	0.97	0.94	0.94
DLF	0.86	0.91	0.79	0.85	0.86	0.94	0.92	0.97	0.94	0.94
Chi-squared	0.81	0.87	0.72	0.79	0.81	0.93	0.90	0.97	0.94	0.93
Pearson Correlation	0.72	0.90	0.48	0.63	0.71	0.94	0.92	0.96	0.95	0.94
Random Forest	0.86	0.91	0.80	0.85	0.86	0.94	0.92	0.96	0.94	0.94
6 Features (Intersection)	0.62	0.88	0.28	0.43	0.62	0.93	0.91	0.96	0.93	0.93
4 Features (flow+fwd)	0.55	0.81	0.13	0.23	0.55	0.93	0.90	0.95	0.93	0.93
	Deep Learning									
ALL Features	0.95	0.94	0.94	0.94	0.97					
DLF	0.94	0.95	0.94	0.94	0.97					
Chi-squared	0.94	0.94	0.95	0.94	0.96					
Pearson Correlation	0.94	0.94	0.95	0.94	0.95					
Random Forest	0.94	0.93	0.94	0.94	0.96					
6 Features (Intersection)	0.89	0.89	0.90	0.89	0.94					
4 Features (flow+fwd)	0.89	0.89	0.89	0.89	0.91					

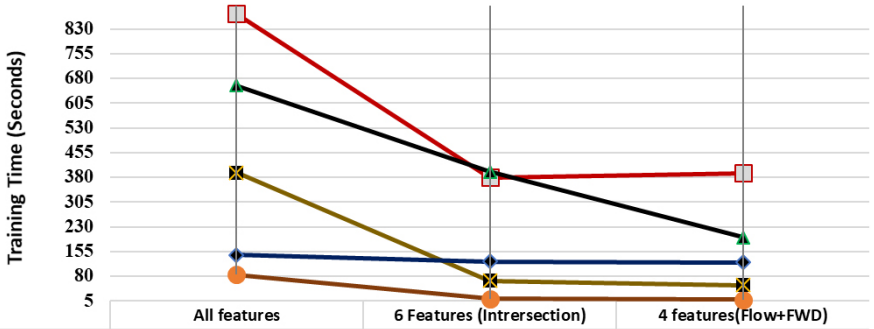
The performance metrics also show that support vector machine, logistic regression, and naïve bays can be replaced by others due to their not good enough performance compared to others in this experiment. Despite the slight differences in performance, Random forest, KNN, and Deep Learning show the best performance followed by Decision trees and Adaboost. The learning algorithms were also evaluated on the unanimous six features; the overlap among the top thirty features ranked by all feature selection methods. The performance of Gaussian naïve bays classifiers is still incomparable to others, yet significantly improved when trained on only unanimous six features; the accuracy and AUC have improved from 57% to 73%. The decision tree classifier has also improved by 2% accuracy and 4% on AUC. While random forest, KNN, and Adaboost show almost the same performance with only six features, the deep learning model performance has significantly dropped by 6%. This observation may not be surprising given that the deep learning algorithm tries to extract as much information as possible from the datasets [28].

The findings of the top features intersection and results in Table 1 encourage the evaluation of the learning algorithms on only those four features related to forwarding direction and traffic flow information. Random forest, KNN, and decision trees can produce models of comparable performance using only four features; namely "Flow Pkts/s", "Init Fwd Win Byts", "Fwd Seg Size Min", and "Fwd Pkts/s", where, on average, the accuracy, F1 score, and AUC are about 94%, Table 1. The models on four features could be easier to be explained and their complexity could significantly be improved. Figure 2 shows the performance comparison of five models built using all features (68 features), unanimous six features, and only four unanimous features. Figure 3 shows the training time in seconds for only five learning algorithms that perform well in this experiment. Besides, the possible benefits of model complexity and explainability when the models have been trained on only four features, the training time is also an important factor to take into account, especially if the models need to be updated frequently. Figure 3 shows that random forest, decision tree, KNN, and Adaboost can produce models on 4 features using almost 50%, 10%, 30%, and 13% respectively, of the training time on all features; 68 features.



	All features	6 Features (Intrersection)	4 features(Flow+FWD)
RF	0.95	0.94	0.94
DT	0.93	0.95	0.94
KNN	0.95	0.94	0.94
AdaBoost	0.94	0.93	0.93
Deep Learning	0.95	0.89	0.89

Fig. 2. The performance of five models built using all features, six unanimous, and only four unanimous features



	All features	6 Features (Intrersection)	4 features(Flow+FWD)
RF	876	379.47	393
DT	85	12.1	8.7
KNN	658.84	397	198.2
AdaBoost	394.66	65.843	53.04
Deep Learning	145	124	122

Fig. 3. Training times in seconds using different sets of features; all features, six unanimous, and only four unanimous features

5 Conclusion

The main objective of this study is to support the network traffic monitoring systems by injecting simple, lightweight, and explainable ML/DL models. These models could need a few data features to be collected from the traffics and consume less training or retraining time. For this purpose, we randomly sampled one million instances, that cover all possible types of network intrusions in the CSE-CIC-IDS2018 big data set, which have been divided into 70% for feature selection and model training, and 30% used as an unseen testing dataset. The dataset consists of 68 traffic features and one binary label; "Benign" or "Attack". The results showed that there is no significant difference in model performance when trained on all features or the top 30 features ranked by any of the four feature selection methods in this experiment. We also evaluated the models trained on only six and four unanimous features. Interestingly, we could have ML models, trained on only four features, comparable in their performance to the same models trained on all features. That being said, ML developers might have better models in terms of complexity, explainability, and size for deployment options. In addition, the training time can be dropped from ~10% to ~50% using four unanimous features instead of all traffic features. Furthermore, the feature selection methods highlighted the importance of the measures of forwarding (features of FWD) and traffic flow (features of FLOW) in this dataset.

The experiment findings also show that random forest, decision tree, KNN, Adaboost, and deep learning would be better choices for ML developers for this dataset. As for deep learning algorithms, using the set of all features could be better since the difference in training times is not significant. In our future experiments, we would extend this work to cover multi classes with some learning optimization algorithms with a focus on the imbalance problem in this dataset.

6 Acknowledgment

This publication was made possible by the research grant from the Deanship of Scientific Research at The Hashemite University, Jordan. The statements made herein are solely the responsibility of the authors.

7 References

[1] Pebriantika, Leni, Basuki Wibawa, and Maria Paristiowati. "Adoption of mobile learning: the influence and opportunities for learning during the covid-19 pandemic." International Journal of Interactive Mobile Technologies 15, no. 5 (2021): 222–230. https://doi.org/10.3991/ijim.v15i05.21067

[2] Nguyen, Nga Thuy, and Huong Thi Thu Tran. "Factors affecting students' desire to take upcoming online courses after e-learning experience during covid-19." International Journal of Interactive Mobile Technologies 16, no. 1 (2022): 22–37. https://doi.org/10.3991/ijim.v16i01.26777

[3] Ahmad, Kashif, Majdi Maabreh, Mohamed Ghaly, Khalil Khan, Junaid Qadir, and Ala Al-Fuqaha. "Developing future human-centered smart cities: critical analysis of smart city security, data management, and ethical challenges." Computer Science Review 43 (2022): 100452. https://doi.org/10.1016/j.cosrev.2021.100452

[4] Obeidat, Ibrahim, Nabhan Hamadneh, Mouhammd Alkasassbeh, Mohammad Almseidin, and Mazen AlZubi. "Intensive pre-processing of KDD cup 99 for network intrusion classification using machine learning techniques." International Journal of Interactive Mobile Technologies 13, no. 1 (2019): 70–84. https://doi.org/10.3991/ijim.v13i01.9679

[5] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches." Transactions on Emerging Telecommunications Technologies 32, no. 1 (2021): e4150. https://doi.org/10.1002/ett.4150

[6] Prasad, Ramjee, and Vandana Rohokale. Cyber security: the lifeline of information and communication technology. Cham, Switzerland: Springer International Publishing, 2020. https://doi.org/10.1007/978-3-030-31703-4

[7] Li, XuKui, Wei Chen, Qianru Zhang, and Lifa Wu. "Building auto-encoder intrusion detection system based on random forest feature selection." Computers & Security 95 (2020): 101851. https://doi.org/10.1016/j.cose.2020.101851

[8] Dwibedi, Smirti, Medha Pujari, and Weiqing Sun. "A comparative study on contemporary intrusion detection datasets for machine learning research." In 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 1–6. IEEE, 2020. https://doi.org/10.1109/ISI49825.2020.9280519

[9] Leevy, Joffrey L., and Taghi M. Khoshgoftaar. "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 big data." Journal of Big Data 7, no. 1 (2020): 1–19. https://doi.org/10.1186/s40537-020-00382-x

[10] Muhsen, Atheer R., Ghazwh G. Jumaa, Nadia F. AL Bakri, and Ahmed T. Sadiq. "Feature selection strategy for network intrusion detection system (NIDS) using meerkat clan algorithm." International Journal of Interactive Mobile Technologies 15, no. 16 (2021): 158–171. https://doi.org/10.3991/ijim.v15i16.24173

[11] Depren, Ozgur, Murat Topallar, Emin Anarim, and M. Kemal Ciliz. "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks." Expert systems with Applications 29, no. 4 (2005): 713–722. https://doi.org/10.1016/j.eswa.2005.05.002

[12] Fitni, Qusyairi Ridho Saeful, and Kalamullah Ramli. "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems." In 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), pp. 118–124. IEEE, 2020. https://doi.org/10.1109/IAICT50021.2020.9172014

[13] Hua, Yanpei. "An efficient traffic classification scheme using embedded feature selection and lightgbm." In 2020 Information Communication Technologies Conference (ICTC), pp. 125–130. IEEE, 2020. https://doi.org/10.1109/ICTC49638.2020.9123302

[14] Kim, Jiyeon, Jiwon Kim, Hyunjung Kim, Minsun Shim, and Eunjung Choi. "CNN-based network intrusion detection against denial-of-service attacks." Electronics 9, no. 6 (2020): 916. https://doi.org/10.3390/electronics9060916

[15] D'hooge, Laurens, Tim Wauters, Bruno Volckaert, and Filip De Turck. "Inter-dataset generalization strength of supervised machine learning methods for intrusion detection." Journal of Information Security and Applications 54 (2020): 102564. https://doi.org/10.1016/j.jisa.2020.102564

[16] ARSLAN, Recep Sinan. "FastTrafficAnalyzer: an efficient method for intrusion detection systems to analyze network traffic." Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi 12, no. 4 (2021): 565–572. https://doi.org/10.24012/dumf.1001881

[17] Leevy, Joffrey L., John Hancock, Richard Zuech, and Taghi M. Khoshgoftaar. "Detecting cybersecurity attacks using different network features with lightgbm and xgboost learners." In 2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI), pp. 190–197. IEEE, 2020. https://doi.org/10.1109/CogMI50398.2020.00032

[18] Huancayo Ramos, Katherinne Shirley, Marco Antonio Sotelo Monge, and Jorge Maestre Vidal. "Benchmark-based reference model for evaluating botnet detection tools driven by traffic-flow analytics." Sensors 20, no. 16 (2020): 4501. https://doi.org/10.3390/s20164501

[19] Lima Filho, Francisco Sales de, Frederico AF Silveira, Agostinho de Medeiros Brito Junior, Genoveva Vargas-Solar, and Luiz F. Silveira. "Smart detection: an online approach for DoS/DDoS attack detection using machine learning." Security and Communication Networks 2019 (2019). https://doi.org/10.1155/2019/1574749

[20] Karatas, Gozde, Onder Demir, and Ozgur Koray Sahingoz. "Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset." IEEE Access 8 (2020): 32150–32162. https://doi.org/10.1109/ACCESS.2020.2973219

[21] Ravikumar, Dharshini. Towards Enhancement of Machine Learning Techniques Using CSE-CIC-IDS2018 Cybersecurity Dataset. Rochester Institute of Technology, 2021.

[22] Alghayadh, Faisal, and Debatosh Debnath. "A hybrid intrusion detection system for smart home security." In 2020 IEEE International Conference on Electro Information Technology (EIT), pp. 319–323. IEEE, 2020. https://doi.org/10.1109/EIT48999.2020.9208296

[23] Lypa, Borys, Oleh Iver, and Viktor Kifer. "Application of machine learning methods for network intrusion detection system." In Proceeding of Processing, Transmission and Security of Information, pp. 233–240, Bielsko-Biała, Poland, Vol. 2, December 6, 2019.

[24] Sadiq, Ali. "Intrusion Detection Using the WEKA Machine Learning Tool." Department of Electrical and Computer Engineering, University of Victoria (2021).

[25] Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani. "Toward generating a new intrusion detection dataset and intrusion traffic characterization." ICISSp 1 (2018): 108–116. https://doi.org/10.5220/0006639801080116

[26] Zuech, Richard, John Hancock, and Taghi M. Khoshgoftaar. "Detecting SQL injection web attacks using ensemble learners and data sampling." In 2021 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 27–34. IEEE, 2021. https://doi.org/10.1109/CSR51186.2021.9527990

[27] Alsahaf, Ahmad, Nicolai Petkov, Vikram Shenoy, and George Azzopardi. "A framework for feature selection through boosting." Expert Systems with Applications 187 (2022): 115895. https://doi.org/10.1016/j.eswa.2021.115895

[28] Qolomany, Basheer, Majdi Maabreh, Ala Al-Fuqaha, Ajay Gupta, and Driss Benhaddou. "Parameters optimization of deep learning models using particle swarm optimization." In 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1285–1290. IEEE, 2017. https://doi.org/10.1109/IWCMC.2017.7986470

8 Authors

Majdi Maabreh, Ph.D., is an assistant professor in the Department of Information Technology, the program of Data Science and AI, at The Hashemite University, Jordan. His research interests include Big data science, machine learning, deep learning, smart services, and AI safety.

Ibrahim Obeidat, Ph.D., is a professor of networking and cybersecurity at Prince Al-Hussein Bin Abdullah II Faculty of Information Technology; The Hashemite University, Jordan. E-mail: imsobeidat@hu.edu.jo

Esraa Abu Elsoud, Asma Alnajjar, and Rahaf Alzyoud are graduate students pursuing a Master's degree in Cybersecurity at Prince Al-Hussein Bin Abdullah II Faculty of Information Technology; The Hashemite University, Jordan.

Omar Darwish, Ph.D., is an assistant professor in the Department of Information Security and Applied Computing at Eastern Michigan University, USA. His research interests include Cybersecurity, IoT, National Language Processing, Machine Learning, Public Health.

Article submitted 2022-02-14. Resubmitted 2022-04-14. Final acceptance 2022-04-14. Final version published as submitted by the authors.