2022 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

2022 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom) | 978-1-6654-6497-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISPA-BDCLOUD-SOCIALCOM-SUSTAINCOM57177.2022.00040

Curse of Feature Selection: a Comparison Experiment of DDoS Detection Using Classification Techniques

Wenjia Wang[1], Seyed Masoud Sadjadi[2], Naphtali Rishe[3]
Knight Foundation School of Computing and Information Sciences
Florida International University
Miami, USA
Email:wwang048@fiu.edu[1], sadjadi@cs.fiu.edu[2], rishen@cs.fiu.edu[3]

***Abstract*— Distributed denial-of-service (DDoS) attack is a malicious cybersecurity attack that has become a global threat. Machine learning (ML) as an advanced technology has been proven to be an effective way against DDoS attacks. Feature selection is a crucial step in ML, and researchers have put endless efforts to mitigate the "Curse of Dimensionality". Feature selection is also causing problems to ML models, such as a decrease in prediction accuracy. Four supervised classification techniques, namely, Decision Tree (DT), k-Nearest Neighbors (KNN), Logistic Regression (LR), and Random Forest (RF), are tested using mutual information score ranking to study the necessity of feature selection in DDoS detection.**

***Index Terms*— Cybersecurity, DDoS, Supervised Learning, Classification, Curse of Dimensionality, Feature Selection**

NOMENCLATURE

Distributed denial-of-service (DDoS), Machine Learning (ML), Supervised learning (SL), Unsupervised Learning (UL), Mutual Information (MI), Decision Tree (DT), k-Nearest Neighbors (KNN), Logistic Regression (LR), Random Forest (RF)

I. INTRODUCTION

Distributed denial-of-service (DDoS) attack is a cyber-attack that mainly targets websites and online services. By directing enormous traffic from multiple servers, attackers can flood a targeted service and make it inaccessible to legitimate users. DDoS attack has always been an immense threat that is causing significant financial losses all over the world [1]. Not only conventional web-based hosting services are DDoS attackers' targets [2], but also infrastructure clouds, cloud computing, and Internet of Things (IoT) are under their menacing strikes [3] [4]. DDoS attack has also been used as a weapon in wars which makes it a threat to human civilization: broadcast media companies in Ukraine have experienced the most DDoS attacks among all the industries during the ongoing Russo-Ukrainian War [5]; both the spread of information and the communication in Ukraine have been severely affected by the attacks [6] [7].

Techniques against DDoS attacks have been studied and presented by different researchers and organizations worldwide [8]. Detection, filtering, and traceback are the three mainstream DDoS defense mechanisms. Meanwhile, DDoS attacks are making fascinating developments over time; they are getting larger and more complex [9], and consistent efforts working on DDoS detection and mitigation are indispensable.

A. Hypothesis

Using machine learning (ML) models has proven to be effective for detecting DDoS attacks [10]. Researchers have shown a significant interest in this approach [11] [12] [13]. Supervised learning (SL) and unsupervised learning (UL) are two fundamental approaches in ML. UL is using ML algorithms to train a machine without human intervention; datasets used in UL are neither labeled nor classified. By using externally supplied labeled data, SL can produce general patterns and hypotheses to predict the fate of unlabeled new data [14] [15]. While processing and analyzing data, SL can be categorized into two types of problems, regression, and classification.

Classification techniques are used to divide data instances into specific categories [16], such as classifying spam into another separate folder from the user's inbox or separating dog pictures from cat pictures. In contrast, regression techniques are more suitable for comprehending the connection between dependent and independent variables. A job like predicting numerical values derived from different data points would need regression models. As for the issue in this study – using a whole lot of input variables to predict a target output (i.e., label), SL classification is the ideal way. The label that needs to be predicted in this work is ”Benign” or ”DDoS”.

As one of the important techniques for implementing ML models, feature selection is both needed in SL and UL. It is used to reduce overfitting and learning time as well as to improve the accuracy of prediction. Classifying and detecting DDoS attacks can be used to distinguish anomaly traffic from normal network traffic, and the network traffic usually has more than 80 network flow features [17].

"Curse of dimensionality" is a famous issue caused by numerous input variables (i.e., dimensionality): the more di-

978-1-6654-6497-0/22/$31.00 ©2022 IEEE
DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00040

mensionalities there are, the higher the number of calculation errors is [18]. In the field of ML, the volume of data that needs to be generalized accurately grows exponentially with the growth of the number of features or dimensions. This is where feature selection techniques come into play. A model that has applied feature selection to find the best possible set of features can get a higher prediction accuracy with fewer data [19]. Thus, it is always common sense that feature selection is an indispensable step for ML: feature selection techniques not only improve models' predictive performance, but also reduce data or computational needs. Precisely for this reason, many feature selection techniques for network traffic classification have been developed [20] [21].

In our study of using classification techniques to detect DDoS attacks, we found a case where a model fitted with all features performed better than a model applied with feature selection. Is this a special case that appears in the DDoS detection or is it possible that models applied feature selection techniques always perform worse than the models trained with all existed features? Could feature selection be a step to skip in DDoS detection or all binary classification tasks? If it is, would skipping feature selection a good choice for all classification cases, such as multi-label classification, multi-class classification, and imbalanced classification? Is there a "curse of feature selection" in ML, which means feature selection is overrated and can be harmful to model training? With such doubts, we conducted a series of experiments using four SL classification algorithms, including decision tree (DT), k-nearest neighbors (KNN), logistic regression (LR), and random forest (RF). Through a two-by-two combination of four models and three datasets with different selected features, a total of twelve models were trained and evaluated.

B. Contribution

1) We offer a comparison of the accuracy of four SL classification techniques predicting DDoS attacks.
2) We explore the relationships between the number of features and prediction accuracy of classification models in ML.
3) We demonstrate that fitting the same classification model with a different number of features may create noise for model training and lower the prediction power.
4) We offer a hypothesis on the potential negative impact of feature selection in the accuracy of the resulted predictive models and conduct experiments to verify it.

II. Related Work

ML methodologies have been used to help against cyber-crime and support data-driven decision-making in Cybersecurity for many years, and building a data-driven security model for every security problem has always been the ultimate goal [22]. As one of the more serious hazards amidst all cybersecurity issues, DDoS attacks are in the spotlight and related detection strategies have been studied extensively [23], and DDoS detection is a typical binary classification problem. This section has reviewed feature selection methodologies from two perspectives: feature selection techniques for DDoS detection and binary classification.

A. Feature Selection Techniques for DDoS Detection

As is common in feature selection approaches, applied DDoS feature selection techniques can be categorized into three kinds of methods: filter, wrapper, and embedded.

1) Filter: uses statistical methods to find the intrinsic properties of the features. Mutual information (MI) score used in this study is a typical filter-based feature selection method. A threshold is set to select the independent and relevant DDoS attack features in this work [24]. A confidence-based filtering technique has been tested that can satisfy the real-time filtering requirements in cloud environment [25]. DDoS classification achieved using a dynamical threshold has been proven to gain higher performance [26]. Filter feature selection technique has also been integrated with an ensemble algorithm to improve DDoS classification accuracy [27].

2) Wrapper: builds a subset of features for the input dataset and evaluates the model trained by the subset, then chooses a new subset and repeats the steps until an ideal performance is obtained [28]. Sequential feature selection was used for optimal feature choosing and combined with MLP-based detection methodology for DDoS detection [29]. Wrapper feature selection approach designed for imbalanced classification has been proven useful [30]. It is believed that a lot of DDoS detection systems have performed better with the enhancement of wrapper strategies [31].

3) Embedded: also called hybrid feature selection, is a combination of both wrapper and filter methods. A hybrid Filter-Wrapper feature selection is used to reduce the number of input variables [32].

Four filter techniques, including information gain, gain ratio, Chi-squared, and ReliefF, were tested in each of six classifiers for DDoS attack detection, and information gain has the best result among the four filter techniques and random forest is the most efficient classifier [33]. It is noteworthy that in this study [33], models trained without any feature selection (i.e., using all features) have better performances overall compared to models that used filter techniques. In another similar study, all three feature selection techniques were used with different ML models; it shows ML models applied feature selections can have better performance in the reductions of processing loads and time [34].

B. Feature Selection Techniques for Binary Classification

The stability of variable selection is very important for model building, which means the smaller the range of variation of feature selection using different training datasets is, the better the feature selection technique is. A method that can improve the stability of wrapper feature selection techniques that suits binary classification is proposed in [35]. A modified discrete particle swarm optimization (PSO) algorithm has been proposed to select the feature subsets for binary classification based on the relevance and dependence of the features [36]. Many preceding studies have shown the defectiveness of single feature selection results, which causes difficulties for professionals in a variety of fields (e.g., medical practitioners) to analyze and interpret the obtained feature subsets. Whereas each of these methods is highly biased, an ensemble feature selection has the advantage to alleviate and compensate for such biases. A novel ensemble feature selection is presented in [37] to alleviate and compensate biases in feature selection for binary classification problems. Some extensions for greedy forward selection and genetic algorithms used in binary classification have been proposed, the run-time is decreased, and the detection rate of relevant features is increased [38].

III. SUPERVISED LEARNING CLASSIFICATION MODELS

This section presents brief descriptions and explanations of the SL classification models used in this work: DT, KNN, LR, and RF.

A. *Decision Tree*

A DT is a hierarchical model that divides the fitting data into homogeneous subsets (nodes), and the subsets are formed based on decision rules, which is asking every feature straightforward yes-or-no question [39]. Every DT is a tree-structured flowchart. As shown in Fig. 1, it is a DT example of removing coordinate points that are belonging to the first and third quadrant. Every input data point will be asked if the value of X is greater than 0; if yes, then the value of y will be checked. Only data points that contain different signs X and Y will be kept. From the root to a certain leaf node, every data point is sorted by several yes-or-no questions, every instance is classified.

B. *K-Nearest Neighbors*

KNN is another essential classification algorithm in SL, and it is widely used in intrusion detection [40] and pattern recognition [41].

The intuition for KNN algorithm can be explained by the example shown in Fig. 2, and KNN computation process can be understood better by figuring its intuition out. The dataset of Fig. 2 contains two sets of points: X points and O points. Triangle c1 is the predicted target that needs to be decided whether it is an X point or an O point. The intuition of nearest-neighbor classification is to find the most similar example with c1. In Fig. 2, c1 is surrounded with O points, the most similar example to c1 is an O point; based on the relative distance measures, c1 is most similar to O points. Hence, c1 will be classified as an O point. Points a1 and b1 are called outliers in this situation. If b1 is the nearest neighbor of c1, c1 could be labeled as an X point. To avoid this situation where a single mislabeled example (i.e., an outlier) has a huge impact on the prediction, the nearest-neighbour algorithm uses more than one (k) nearest neighbor to make the decision, and k most similar training examples are used to label the predict target. That is where the name of this algorithm KNN comes from.

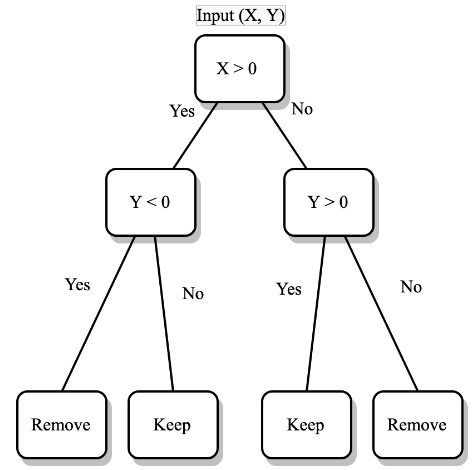

Fig. 1. An example of decision tree

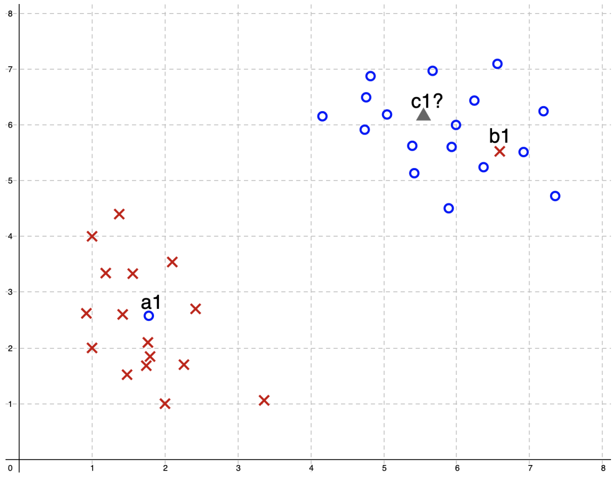

Fig. 2. Intuition of k-nearest neighbors

C. *Logistic Regression*

LR is used for predicting the probability of a target variable. Instead of giving a specific result as 0 and 1, the given probability lies between 0 and 1 [42]. It is widely used for classification problems like risk analysis [43], diabetes prediction [44], cancer detection [45], etc.

In order to map real value into another value between 0 and 1, logistic function, also known as sigmoid function, is used to model data in LR. The sigmoid function can be defined as:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{1}$$

$\sigma(z)$ in (1) is the mapping value of the input value z and e is Euler's number. Fig. 3 is the graph of the sigmoid function, which has an S shape. A threshold is set to classify the returned probability value that is assigned by the sigmoid function [46]. As shown in Fig. 3, if there are two classes, and 0.5 is the threshold value; Class A is 1 and Class B is 0, then values

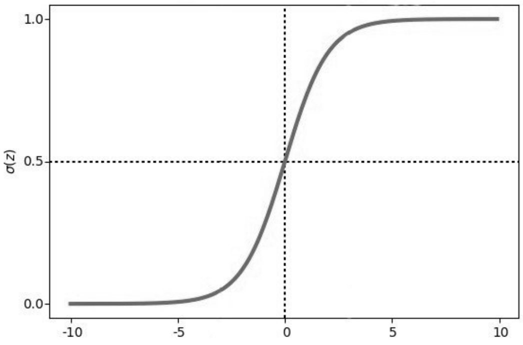

Fig. 3. Sigmoid function graph

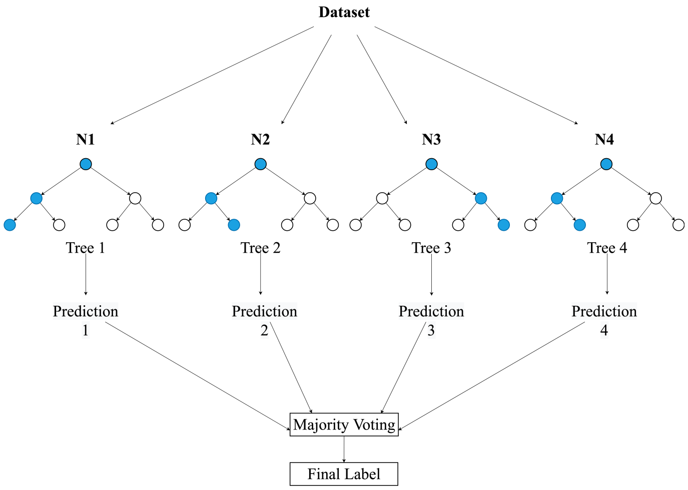

Fig. 4. Intuition of random forest algorithm

above 0.5 in the graph will be classified into Class A and values below 0.5 will be signed into Class B.

D. Random Forest

Same as DT, RF uses tree structures to make predictions; different from DT, RF uses many component trees to get many results, then a prediction is made by averaging all the results [47]. When it is used as a classifier, the output of RF is the category selected by most trees; while dealing with regression problems, RF returns the average prediction of the component trees [48].

Fig. 4 is a simplified computation process of RF. The computation process of RF can be roughly divided into two stages [49]. In stage 1, n feature sets are selected randomly and built from all m ($m > n$) features. Using the best split method builds forests for n decision trees based on n feature sets (i.e., N1, N2, ...) separately. Every tree will reach a node and that will be its prediction. And stage 2 begins after all decision trees got their own predictions. The votes for each predicted value from each tree are counted, and the predicted value that has a majority vote will be the final prediction for this RF classifier.

IV. Data & Experiments

In this study, four SL classification techniques have been applied to Intrusion Detection Evaluation Dataset (CIC-IDS2017) from Canadian Institute for Cybersecurity for DDoS detection [18].

A. Dataset

CIC-IDS2017 contains labeled datasets received by their online system: random samples of network traffic which have been categorized into DDoS attacks or benign. Though not perfect, CIC-IDS2017 is widely recognized as a reliable network traffic dataset that is fitly used as raw data or as flow-based features in CSV files [50] [51].

There are eleven criteria proposed for the building of this benchmark dataset, and they will be the criteria for dataset selection in the future work of this study. CIC-IDS2017 has not only captured original network traffic, it also provided the corresponding datasets that have dropped six special format or timestamp-related features, including “Flow ID”, “Source IP”, “Source Port”, “Destination IP”, “Timestamp”, and “Protocol”, so the datasets are more suitable for ML. The datasets were collected in half-hour increments and the dataset used in this study has 225,745 rows and 79 columns.

Among 79 columns, 78 columns are independent variables and one is dependent variable (i.e., label). All 78 independent features’ names are shown in Fig. 8.

B. Experiments

The following steps are the designed procedure steps for classification of DDoS attacks in the chosen dataset. The ultimate goal of the experiments is to examine the proposed hypothesis, which is whether feature selection is useful and necessarily needed in all ML cases and if it is always bad or good for accuracy.

Step 1: A CIC-IDS2017 DDoS attack dataset “Friday-WorkingHours-Afternoon-DDos” with all attributes is captured as the input.

Step 2: The input dataset is preprocessed. All the missing values are filled in the corresponding median of their columns according to the imputation rule. To make sure all data is meaningful, all the infinity values are replaced with “0”; all the rows with “NaN” values are dropped. Two unique categorical values in the label column, “Benign” and “DDoS”, are signed to 0 and 1 to avoid errors in the training process.

Step 3: Separate the prediction target (i.e., label column) from the dataset and name it y.

Step 4: Construct a ranking with a feature utility metric using MI score. High-mutual-information-score features in SL are considered optimal features since they can influence the predictive model towards the right prediction and increase the accuracy of the model. “It measures the average reduction in uncertainty about x that results from learning the value of y; or vice versa, the average amount of information that x conveys

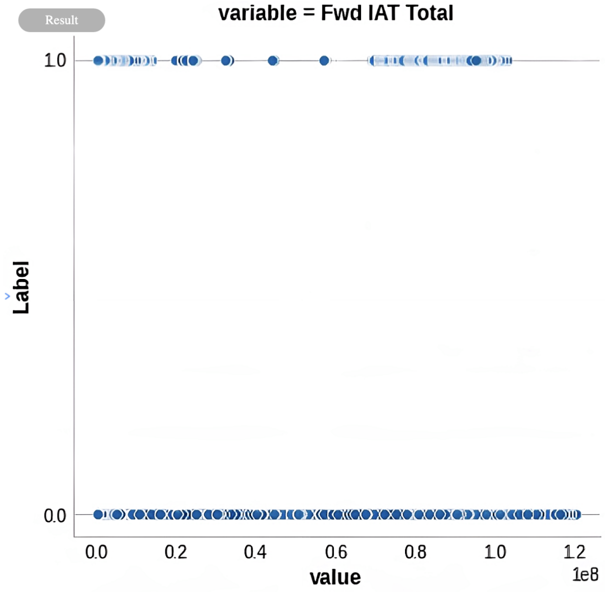

Fig. 5. Relational plot between the highest MI score feature and label

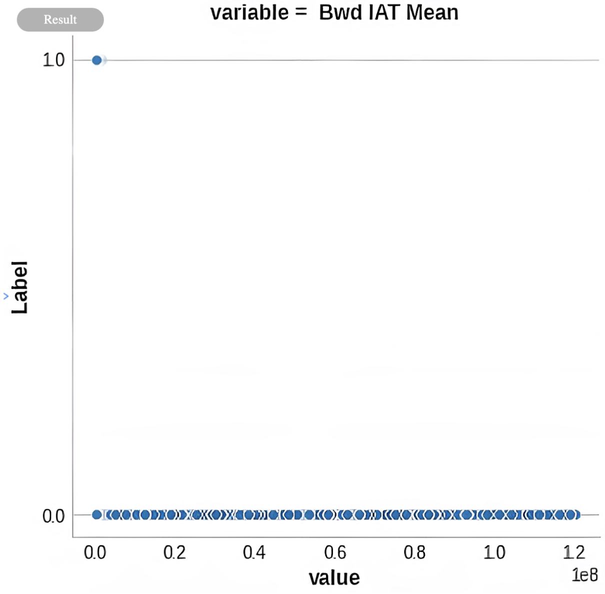

Fig. 6. Relational plot between the median MI score feature and label

about y." [52] Relational Plots of the statistical relationships between the label and three variables are visualized and correspondingly shown in Fig. 5, Fig. 6, and Fig. 7. "Fwd IAT Total" has the highest MI score, 1.418770, and there are lots of data points equal to one in the figure, which means all these traffic are DDoS attacks. The MI score of "Bwd IAT Mean" is the median among all the features, 0.330619. And Fig. 7 shows that there is nearly no "Active Std" value related to label that values 1 (i.e., it is a DDoS attack), and its score also shows the situation: 0.023413, only 1.6% of the highest MI score, it can be conjectured that the feature "Active Std" is not a good sign to label a network traffic. A ranking plot

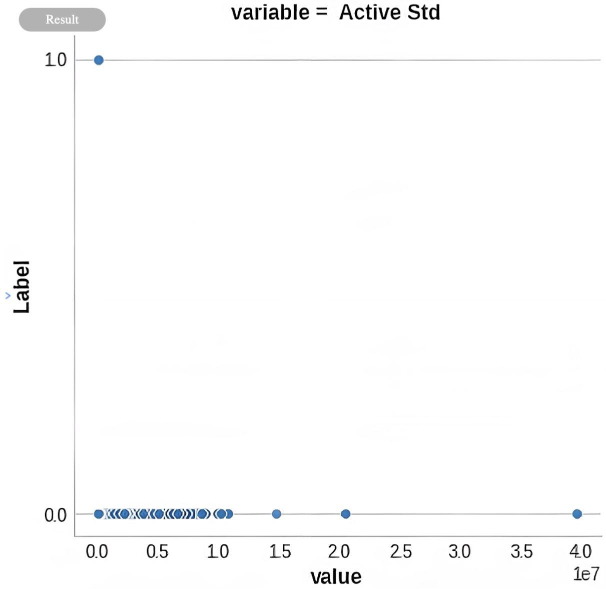

Fig. 7. Relational plot between the lowest MI score feature and label

based on the MI scores of all features is in Fig. 8.

Step 5: Build three datasets with different features based on the MI scores of features. Dataset X1 consists of features the MI score is greater than 0.5, which includes 19 features; X2 consists of features the MI score is greater than 0.2, which includes 47 features; X3 consists of all features, which includes all 78 features.

Step 6: Split the X1 dataset and break up the data into two pieces: the training part and the testing part. Do the same split to the prediction target y dataset. X1-train and y1-train are for the model fitting; X1-validation and y1-validation are prepared for model prediction and evaluation.

Step 7: Build the model. Define a DT model with scikit-learn and fit it with X1-train and y1-train.

Step 8: Make predictions for the prepared validate traffic network dataset X1-validation.

Step 9: Calculate Mean Absolute Error (MAE) of the predictions to summarize the quality of the trained model. The error of every prediction is the difference between the prediction value and the actual value (i.e., y1-validation). With the MAE metric, the absolute value of each error is used, so every prediction error is converted to a positive number. The quality of a model can be measured using the average of those absolute errors.

Step 10: Repeat steps 7 - 9 with the corresponding train datasets and validation datasets split from X2, and X3.

Step 11: Repeat steps 7 - 10 to fit KNN, LR, and RF models.

V. Results

All MAEs of the four models trained by three different feature-selected datasets are presented in tables I to IV. As shown in the figures, both DT and RF have given a significant performance in DDoS attack prediction. More importantly, DT and RF have basically verified our hypothesis: with the

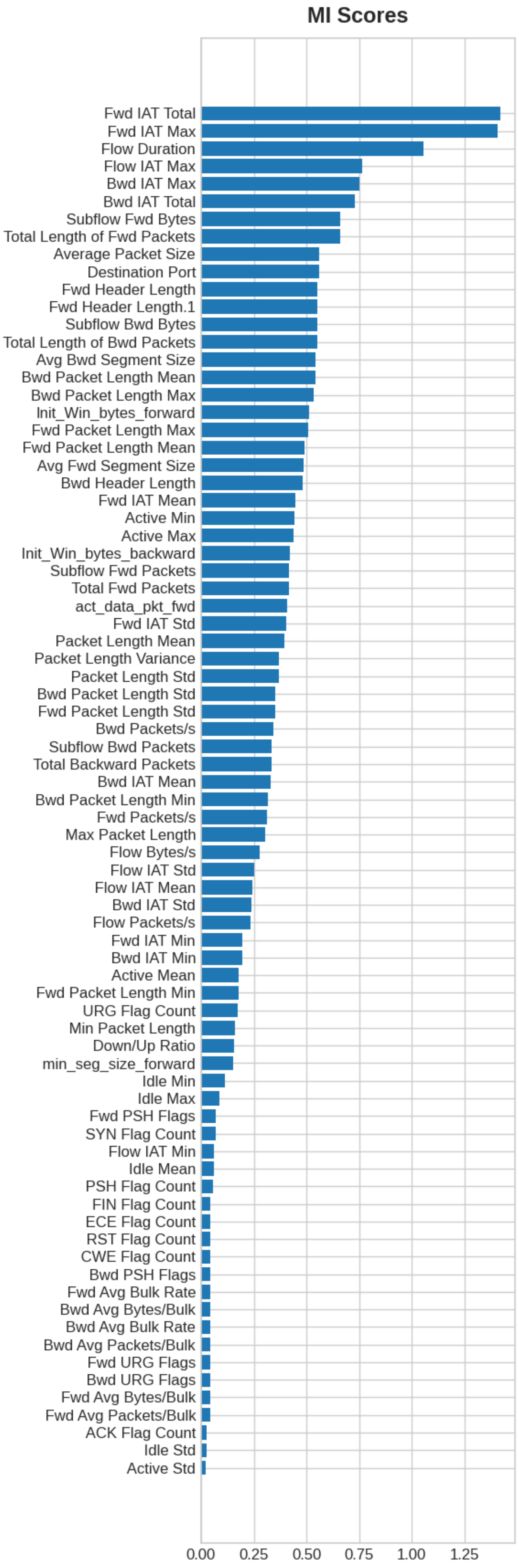

Fig. 8. Mutual information scores of all features

increment of the number of features used for the training of the model, the prediction accuracy of the models has also shown a decent improvement. RF performed well with a small number of features; the difference in prediction accuracy between using 19 features and 78 features is not a big gap. On the contrary, DT model trained with all features gave the best predicted performance among the twelve models. This situation might be related to the calculation structure of DT and RF: as previously stated, DT is making decisions using a tree with lots of leaves, and this can lead to a DT model overfitting or having a high variance. However, as deeper as it gets (i.e., the feature number increases), the prediction accuracy goes higher. In comparison, by averaging the predictions of the multiple decision trees, RF can get an accurate prediction with a few provided features; and the enhancement is not noteworthy either while providing many more features to it.

An interesting result emerged in the experiments using KNN models. When KNN model was trained only using 19 features whose MI scores are higher than 0.2, the worst performance appeared, MAE is 0.0047486577954179. The best accuracy given by KNN model is when the model was trained by 47 features whose MI scores are higher than 0.5; and different from DT and RF models, the model trained using all features is slightly inferior. As mentioned in section III, KNN algorithm is very sensitive to outliers, one mislabeled example can change the prediction result dramatically. That should be the reason why KNN model performed better without using all features. Thus, either using all features or using too less features is not the best option for KNN model training.

TABLE I
MAE of DT Models

	DT
MAE Using Features MI $\geq$ 0.5	0.00039921915569928834
MAE Using Features MI $\geq$ 0.2	0.0001417509789676985
MAE Using All 78 Features	0.00012403210659673618

TABLE II
MAE of KNN Models (K = 3)

	KNN
MAE Using Features MI $\geq$ 0.5	0.0047486577954179
MAE Using Features MI $\geq$ 0.2	0.002285734535854138
MAE Using All 78 Features	0.0023034534082251004

TABLE III
MAE of LR Models

	LR
MAE Using Features MI $\geq$ 0.5	0.02620621223665326
MAE Using Features MI $\geq$ 0.2	0.060297322678384745
MAE Using All 78 Features	0.06442581994081896

In complete contrast to the performance of the other models, the prediction accuracies of LR models are getting lower and lower with the increment of the feature number.

TABLE IV
MAE OF RF MODELS

	RF
MAE Using Features MI $\geq$ 0.5	0.0002796019075631741
MAE Using Features MI $\geq$ 0.2	0.00024062228679766823
MAE Using All 78 Features	0.00021120895866187078

It seems that the added features add random noise to LR models; the prediction power has a noticeable drop with more input variables.

VI. CONCLUSION

The conclusions from the experiments conducted for the hypothesis in this study with four different SL classification models can be summarized as follows:

1) It is not the best option to choose features only based on their MI scores. MI is a univariate metric and the interactions between features cannot be obtained by that. A feature with a low MI score could be very informative when combined with some other features, though it is not so informative all alone. Possible interaction effects between features should be investigated before building feature sets.
2) DT could be the best classification technique for a high accuracy goal when the study has abundant computation and data resources. The prediction accuracy and the number of features are positively correlated, and feature selection could be a limitation in this case.
3) There is a balance or an optimal choice existed between feature reduction and enough feature input while training KNN models; either too many or too few features can make its prediction worse.
4) Extra features can add noise to the training of LR models and lower the prediction power.
5) The performances of two of the four classification models (i.e., DT and RF) used in this study are consistent with the proposed hypothesis. It is a strong argument that the importance and the necessity of feature selection are questionable in some cases. We conclude that this hypothesis deserves further exploration.

VII. FUTURE WORK

The feature selection methods that have been tested in this study is very limited; more feature selection techniques can be tested to examine the hypothesis. Experiments completed in this study for the proposed hypothesis about feature selection are only for DDoS detection, the generalizability of this hypothesis for other problems, like general binary classification problems, multi-class classification, multi-label classification, imbalanced classification, and regression issues, needs to be further experimented.

ACKNOWLEDGMENT

This material is based in part upon work supported by the National Science Foundation under Grant No. MRI20 CNS-2018611.

REFERENCES

[1] D. Anstee, C. F. Chui, P. Bowen, and G. Sockrider, Worldwide Infrastructure Security Report, Arbor Networks Inc., Westford, MA, USA, 2017.

[2] A. Chadd. (2018). DDoS attacks: past, present and future. Network Security, 2018(7), 13-15.

[3] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, “DDoS attacks in cloud computing: Issues, taxonomy, and future directions,” Comput. Commun., vol. 107, pp. 30–48, 2017.

[4] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IOT: Mirai and other botnets,” Computer, vol. 50, no. 7, pp. 80–84, 2017.

[5] O. Yoachimik, “Cloudflare Blog: DDoS Attack Trends for 2022 Q2 ” Cloudflare, San Francisco, California, America, Accessed: July. 6, 2022. [Online]. Available: https://blog.cloudflare.com/ddos-attack-trends-for-2022-q2/

[6] C. & D. O. Foreign, “UK assesses Russian involvement in cyber attacks on Ukraine,” GOV.UK, 18-Feb-2022. [Online]. Available: https://www.gov.uk/government/news/uk-assess-russian-involvement-in-cyber-attacks-on-ukraine. [Accessed: 21-Jul-2022].

[7] P. Doyle, “Major DDoS attacks increasing after invasion of Ukraine,” SearchSecurity, 06-Jun-2022. [Online]. Available: https://www.techtarget.com/searchsecurity/news/252521150/Major-DDoS-attacks-increasing-after-invasion-of-Ukraine. [Accessed: 01-Jul-2022].

[8] N. Z. Bawany, J. A. Shamsi, and K. Salah, “DDoS attack detection and mitigation using SDN: Methods, practices, and solutions,” Arab. J. Sci. Eng., vol. 42, no. 2, pp. 425–441, 2017.

[9] D. Warburton, “2022 application protection report: DDoS attack trends,” F5 Labs, 16-Mar-2022. [Online]. Available: https://www.f5.com/labs/articles/threat-intelligence/2022-application-protection-report-ddos-attack-trends. [Accessed: 02-Aug-2022].

[10] M. Suresh and R. Anitha, “Evaluating machine learning algorithms for detecting DDoS attacks,” in Advances in Network Security and Applications, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 441–452.

[11] G. A. Jaafar, S. M. Abdullah, and S. Ismail, “Review of recent detection methods for HTTP DDOS attack,” Journal of Computer Networks and Communications, vol. 2019, pp. 1–10, 2019.

[12] Q. He et al., “A game-theoretical approach for mitigating edge DDoS attack,” IEEE Trans. Dependable Secure Comput., vol. 19, no. 4, pp. 2333–2348, 2022.

[13] Z. Li, H. Jin, D. Zou, and B. Yuan, “Exploring new opportunities to defeat low-rate DDoS attack in container-based cloud environment,” IEEE Trans. Parallel Distrib. Syst., vol. 31, no. 3, pp. 695–706, 2020.

[14] A. Singh, N. Thakur and A. Sharma, ”A review of supervised machine learning algorithms,” 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 1310-1315.

[15] C. Crisci, B. Ghattas, and G. Perera, “A review of supervised machine learning algorithms and their applications to ecological data,” Ecological Modelling, vol. 240, pp. 113–122, 2012.

[16] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, “Supervised machine learning: A review of classification techniques,” Emerg. Artif. Intell. Appl. Comput. Eng., vol. 160, pp. 3–24, Jul. 2007.

[17] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” Proceedings of the 4th International Conference on Information Systems Security and Privacy, 2018.

[18] R. Bellman, Dynamic Programming. Dover Publications, 1957.

[19] S. Visalakshi and V. Radha, “A literature review of feature selection techniques and applications: Review of feature selection in data mining,” in 2014 IEEE International Conference on Computational Intelligence and Computing Research, 2014, pp. 1–6.

[20] A. Fahad, Z. Tari, I. Khalil, A. Almalawi, and A. Y. Zomaya, “An optimal and stable feature selection approach for traffic classification based on multi-criterion fusion,” Future Gener. Comput. Syst., vol. 36, pp. 156–169, 2014.

[21] A. Fahad, Z. Tari, I. Khalil, I. Habib, and H. Alnuweiri, “Toward an efficient and scalable feature selection approach for internet traffic classification,” Comput. netw., vol. 57, no. 9, pp. 2040–2057, 2013.

[22] I. H. Sarker, A. S. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, “Cybersecurity Data Science: An overview from machine learning perspective,” Journal of Big Data, vol. 7, no. 1, 2020.

[23] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” Comput. Commun. Rev., vol. 34, no. 2, pp. 39–53, 2004.

[24] L. Zhou, Y. Zhu, T. Zong, and Y. Xiang, “A feature selection-based method for DDoS attack flow classification,” Future Gener. Comput. Syst., vol. 132, pp. 67–79, 2022.

[25] W. Dou, Q. Chen, and J. Chen, “A confidence-based filtering method for DDoS attack defense in cloud environment,” Future Gener. Comput. Syst., vol. 29, no. 7, pp. 1838–1850, 2013.

[26] I. Ko, D. Chambers, and E. Barrett, “Adaptable feature-selecting and threshold-moving complete autoencoder for DDoS flood attack mitigation,” J. Inf. Secur. Appl., vol. 55, no. 102647, p. 102647, 2020.

[27] K. J. Singh and T. De, “Efficient classification of DDoS attacks using an ensemble feature selection algorithm,” J. Intell. Syst., vol. 29, no. 1, pp. 71–83, 2017.

[28] J. Tang, S. Alelyani, and H. Liu, ”Feature selection for classification: A review,” Data Classification: Algorithms and Applications. CRC Press, 2014. pp. 37-64

[29] M. Wang, Y. Lu, and J. Qin, “A dynamic MLP-based DDoS attack detection method using feature selection and feedback,” Comput. Secur., vol. 88, no. 101645, p. 101645, 2020.

[30] P. Yang, W. Liu, B. B. Zhou, S. Chawla, and A. Y. Zomaya, “Ensemble-based wrapper methods for feature selection and class imbalance learning,” in Advances in Knowledge Discovery and Data Mining, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 544–555.

[31] K. Bouzoubaa, Y. Taher, and B. Nsiri, “Predicting DOS-DDOS attacks: Review and evaluation study of feature selection methods based on wrapper process,” Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 5, 2021.

[32] M. Belouch, S. Elhadaj, and M. Idhammad, “A hybrid filter-wrapper feature selection method for DDoS detection in cloud computing,” Intell. Data Anal., vol. 22, no. 6, pp. 1209–1226, 2018.

[33] O. Osanaiye, K. Choo and M. Dlodlo, ”Analysing Feature Selection and Classification Techniques for DDoS Detection in Cloud”, Southern Africa Telecommunication Networks and Applications Conference (SATNAC), 2016.

[34] H. Polat, O. Polat, and A. Cetin, “Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models,” Sustainability, vol. 12, no. 3, p. 1035, 2020.

[35] S. Cateni and V. Colla, “Improving the stability of wrapper variable selection applied to binary classification,” Int. J. Comput. Inf. Syst. Ind. Manag. Appl, vol. 8, pp. 214–225, 2016.

[36] A. Unler and A. Murat, “A discrete particle swarm optimization method for feature selection in binary classification problems,” Eur. J. Oper. Res., vol. 206, no. 3, pp. 528–539, 2010.

[37] U. Neumann et al., “Compensation of feature selection biases accompanied with improved predictive performance for binary classification by using a novel ensemble feature selection approach,” BioData Min., vol. 9, no. 1, p. 36, 2016.

[38] R. Jagdhuber, M. Lang, A. Stenzl, J. Neuhaus, and J. Rahnenführer, “Cost-Constrained feature selection in binary classification: adaptations for greedy forward selection and genetic algorithms,” BMC Bioinformatics, vol. 21, no. 1, p. 26, 2020.

[39] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, “An introduction to decision tree modeling,” Journal of Chemometrics, vol. 18, no. 6, pp. 275–285, 2004.

[40] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, “Intrusion detection by Machine Learning: A Review,” Expert Systems with Applications, vol. 36, no. 10, pp. 11994–12000, 2009.

[41] R. Fraiman, A. Justel, and M. Svarc, “Pattern recognition via projection-based kNN rules,” Computational Statistics & Data Analysis, vol. 54, no. 5, pp. 1390–1403, 2010.

[42] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ: Wiley-Blackwell, 2013.

[43] L. C. Kleinman and E. C. Norton, “What’s the risk? A simple approach for estimating adjusted risk measures from nonlinear models including logistic regression,” Health Services Research, vol. 44, no. 1, pp. 288–302, 2009.

[44] A. Mujumdar and V. Vaidehi, “Diabetes prediction using machine learning algorithms,” Procedia Computer Science, vol. 165, pp. 292–299, 2019.

[45] X. Zhou, K.-Y. Liu, and S. T. C. Wong, “Cancer classification and prediction using logistic regression with Bayesian Gene Selection,” Journal of Biomedical Informatics, vol. 37, no. 4, pp. 249–259, 2004.

[46] A. Pant, “Introduction to Logistic Regression,” Towards Data Science, 22-Jan-2019. [Online]. Available: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148. [Accessed: 19-Jul-2022].

[47] T. K. Ho, “Random decision forests,” in Proceedings of 3rd International Conference on Document Analysis and Recognition, 2002, vol. 1, pp. 278–282 vol.1.

[48] T. K. Ho, “The random subspace method for constructing decision forests,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 8, pp. 832–844, 1998.

[49] “Random forest classifier,” Kaggle.com, 13-Mar-2020. [Online]. Available: https://www.kaggle.com/code/prashant111/random-forest-classifier-tutorial/notebook. [Accessed: 20-July-2022].

[50] A. Rosay, E. Cheval, F. Carlier, and P. Leroux, “Network intrusion detection: A comprehensive analysis of CIC-IDS2017,” in Proceedings of the 8th International Conference on Information Systems Security and Privacy, 2022.

[51] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, “A detailed analysis of the CICIDS2017 data set,” in Communications in Computer and Information Science, Cham: Springer International Publishing, 2019, pp. 172–188.

[52] D. J. C. MacKay, Information theory, inference and learning algorithms. Cambridge, England: Cambridge University Press, pp. 139, 2003.