Feature Popularity Between Different Web Attacks with Supervised Feature Selection Rankers

Richard Zuech, John Hancock, Taghi M. Khoshgoftaar
Florida Atlantic University
Boca Raton, FL 33431
Email: rzuech@fau.edu, jhancoc4@fau.edu, khoshgof@fau.edu

***Abstract*—We introduce the novel concept of feature popularity with three different web attacks and big data from the CSE-CIC-IDS2018 dataset: Brute Force, SQL Injection, and XSS web attacks. Feature popularity is based upon ensemble Feature Selection Techniques (FSTs) and allows us to more easily understand common important features between different cyberattacks, for two main reasons. First, feature popularity lists can be generated to provide an easy comprehension of important features across different attacks. Second, the Jaccard similarity metric can provide a quantitative score for how similar feature subsets are between different attacks. Both of these approaches not only provide more explainable and easier-to-understand models, but they can also reduce the complexity of implementing models in real-world systems. Four supervised learning-based FSTs are used to generate feature subsets for each of our three different web attack datasets, and then our feature popularity frameworks are applied. For these three web attacks, the XSS and SQL Injection feature subsets are the most similar per the Jaccard similarity. The most popular features across all three web attacks are: Flow_Bytes_s, Flow_IAT_Max, and Flow_Packets_s. While this introductory study is only a simple example using only three web attacks, this feature popularity concept can be easily extended, allowing an automated framework to more easily determine the most popular features across a very large number of attacks and features.**

***Index Terms*—Feature Popularity, Feature Similarity, Feature Selection, Big Data, Intrusion Detection, Web Attacks.**

I. Introduction

With consumers spending over $600 billion on e-commerce in the United States during 2019 [1], cybersecurity is becoming increasingly important to help defend against attackers. Machine learning [2], [3] can be employed to help in detecting cyberattacks. Feature selection [4] is a common technique used by machine learning practitioners. Benefits of feature selection include improving classification efficiency by training models with fewer features which requires less computing resources, and feature selection can sometimes even improve classification performance.

Another benefit of feature selection, in the context of cybersecurity, is it can help practitioners better understand the attack detection process. This can be accomplished because feature selection can identify the most important features during the model building process. Gaining a better understanding of the most important features not only helps during the machine learning model building process, but it can be even more helpful when machine learning models are deployed and implemented into real-world products and systems.

Various Feature Selection Techniques (FSTs) can generate very different feature lists on the same dataset (which we identify throughout this study). However, finding "common features" between different FSTs can sometimes help us find an even better feature subset through the diversity of different FSTs acting in an ensemble [5]. While ensemble FSTs can sometimes improve classification performance, they might still be desirable even with minor degradations in classification performance. Reasons for using ensemble FSTs with minor decreases in performance may include: feature stability, reducing model complexity for real-world implementations, and simply providing models which are easier to understand and are more explainable.

Feature similarity is a concept which is implicit in the ensemble FST process. For example, an ensemble FST can find similar (or "common") features between the different FSTs by identifying features which appear in common among the "Top N" feature importance lists from different FSTs. However, we extend this concept and introduce the notion of "feature popularity" by also finding common features across different datasets. For example, in cybersecurity there are different types of cyberattacks and we can generate different datasets which are based upon those different attacks.

To explore feature popularity, we utilize the CSE-CIC-IDS2018 dataset which was created by Sharafaldin et al. [6]. CSE-CIC-IDS2018 is a more recent version of the popular CIC-IDS2017 dataset [7], which was also created by Sharafaldin et al. The CSE-CIC-IDS2018 dataset includes over 16 million instances which includes normal instances, as well as the following family of attacks: web attack, Denial of Service (DoS), Distributed Denial of Service (DDoS), brute force, infiltration, and botnet.

Given its richness in containing many different attack labels, CSE-CIC-IDS2018 [8] is a good dataset for investigating feature popularity. To do so, we only evaluate the following three different web attacks from CSE-CIC-IDS2018: Brute Force (BF), Cross-Site Scripting (XSS), and SQL Injection (which we commonly refer to only as "SQL" throughout this document). Basically, we generate three new datasets by combining each of these three attack labels with all of the normal traffic from the full CSE-CIC-IDS2018 dataset. We then compare feature popularity results between these three different web attack datasets.

Brute Force web attacks correspond to brute force login

attacks targeting web pages. Next, the XSS web attack refers to where attackers inject malicious client-side scripts into susceptible web pages targeting web users which view those pages. Finally, the SQL Injection web attack represents a code injection technique where attackers craft special sequences of characters and submit them to web page forms in an attempt to directly query the back-end database of that website. The feature popularity techniques which we introduce to CSE-CIC-IDS2018 in this study allows us to visually explain and quantify common features across these three different web attacks.

The remaining sections of this paper are organized as follows. The Related Work section studies existing literature for feature popularity with CSE-CIC-IDS2018 data. Then, the Methodologies section describes the data preparation, classifiers and supervised FSTs, performance metrics, and feature popularity techniques applied in our experiments. The Results section provides our results and analysis. Finally, the Conclusion section concludes our work.

II. Related Work

Sarhan et al. [9] focus on how to explain models with feature selection and the eXplainable Artificial Intelligence (XAI) method using the CSE-CIC-IDS2018 dataset. Their motivation in using this XAI method is to be able to more easily explain the attack detection process through a better understanding of what the most important features are after applying the feature selection process. To score the most important features, they assign a Shapley value to each of the features. The Shapley value "is the weighted average of the respective contribution of a feature value" (which is essentially the amount a feature contributes towards making a prediction).

Random Forest and Deep Feed Forward classifiers are employed by Sarhan et al. to score the top 20 features of CSE-CIC-IDS2018 with Shapley values. These two different classifiers produce two very different lists of top 20 features for each of the classifiers. For example, the top ranked feature from the Deep Feed Forward classifier is Bwd_Packets_s, while the Random Forest classifier ranks this same feature as the 16th best feature overall. It is difficult to ascertain how similar the two feature subsets are between the two different classifiers. Their research does not compare the feature similarity between these two different feature subsets like our research does. They do not benchmark the classification performance of only using the top 20 features versus all of the features, but their main motivation is to better explain and interpret the classification models by understanding the most important features used to generate those models.

Leevy et al. [10] apply an ensemble feature selection technique to the full CSE-CIC-IDS2018 dataset, and employ binary classification by merging the multiple attacks to one attack label. The ensemble feature selection technique considers seven different FSTs, of which three are filter-based FSTs and four are supervised FSTs. Different feature subsets are generated by finding common features among the seven different FSTs where a certain number of the FSTs agree. This ensemble FST concept is similar to this current study, but the main difference is that study only considers one attack dataset while this current study extends the approach by also finding common features across multiple attack datasets. In other words, this current study is different as it not only finds common features for a single attack dataset, but it also finds common features across multiple attacks. Also, the current study introduces the Jaccard similarity for quantifying feature subset similarity between different attacks.

Fitni et al. [11] compare two different feature selection techniques with the full CSE-CIC-IDS2018 dataset and map the multiple attacks to a binary classification problem with only attack and normal labels. Their two feature selection techniques are Chi-Squared (top 22 features) and Spearman's rank correlation coefficient (top 23 features), and they compare these two FSTs with Logistic Regression (LR) and Decision Tree (DT) classifiers. The Spearman's rank correlation coefficient technique performed better with F1 scores of 0.983 and 0.974, as compared to the Chi-Squared results of F1 scores of 0.791 and 0.974. Also, the full feature set performed best with the Decision Tree yielding an Area Under the Receiver Operating Characteristic Curve (AUC) score of 0.975.

Based on the results of these two classifiers (LR and DT), Fitni et al. performed further experimentation using seven classifiers with only the Spearman's rank correlation coefficient FST and the full feature set. They do a nice job displaying their feature subsets which provides good insight into the attack detection process (similar to the XAI motivations of Sarhan et al. [9]). However, their research does not consider feature popularity concepts like our study does.

Beechey et al. [12] apply feature selection to the Goldeneye and Slowloris Denial of Services attacks from CSE-CIC-IDS2018. Their dataset only considers one day out of the ten days of available network traffic from CSE-CIC-IDS2018 with 1,048,575 instances, while our experiment considers all ten days of normal traffic encompassing over 13 million instances. Eight feature selection techniques were employed with at least six classifiers, and their Table 6 indicates perfect AUC scores for several combinations of FSTs and classifiers (sometimes overfitting can be associated with perfect classification). While Beechey et al. and others [13] apply feature ranking techniques to CSE-CIC-IDS2018, they do not explore the notion of feature popularity between different attacks or common features between different FSTs.

We thoroughly surveyed Google Scholar to find related works to CSE-CIC-IDS2018 and our feature popularity research, and we searched for terms like "feature popularity", "CSE-CIC-IDS2018 feature similarity", and "CSE-CIC-IDS2018 feature selection". First, Google Scholar did not provide any results significantly related to our "feature popularity" focus from any application domain, and so we believe we are the first to conceive the feature popularity concept. Second, after reviewing more than 261 CSE-CIC-IDS2018 works at the time of this writing, only the works by Sarhan et al. [9] and Leevy et al. [10] had aspects which were remotely similar to our feature popularity research. While the rest of the

CSE-CIC-IDS2018 corpus did contain some feature selection aspects, we only included [11], [12] as those had the most compelling details of feature ranking with CSE-CIC-IDS2018.

The XAI method highlighted by Sarhan et al. provides good insights, and we agree that a better understanding and explanation of feature subsets is important to the attack detection process when implementing machine learning models in the real world. Moreover, this is especially important when considering different attacks like we have done, as different attacks can generate very different feature subsets. To the best of our knowledge, ours is the first study to define the concept of feature popularity, especially in the context of network intrusion detection. In addition, it is the first study to utilize the Jaccard similarity metric for examining feature similarity between different attack types.

III. Methodologies

A. Data Preparation

Table I indicates the instances used in this experiment with big data for the three different web attacks: Brute Force, SQL Injection, and XSS. Of the total 79 independent features available from CSE-CIC-IDS2018, we dropped the following 13 features: Timestamp, Protocol, Bwd_PSH_Flags, Bwd_URG_Flags, Fwd_Avg_Bytes_Bulk, Fwd_Avg_Packets_Bulk, Fwd_Avg_Bulk_Rate, Bwd_Avg_Bytes_Bulk, Bwd_Avg_Packets_Bulk, Bwd_Avg_Bulk_Rate, Flow_Duration, Init_Win_bytes_forward, and Init_Win_bytes_backward. Due to space limitations, we cannot elaborate on the reasons for filtering out these features or other data preparation issues.

To treat the severe class imbalance encountered throughout this experiment, we employ random undersampling (RUS) [14]. Every model trained throughout this experiment first applies RUS at a 1:1 sampling ratio during the model training process (sampling is not applied during the model testing process). Further details for how we apply RUS can be found in [15].

TABLE I
Web Attacks Used in this Experiment from CSE-CIC-IDS2018

Attack Type	Attack Instances	Normal Instances
Brute Force Web	611	13,390,234
Sql Injection Web	87	13,390,234
XSS Web	230	13,390,234

B. Classifiers and Supervised Feature Selection

Four different classifiers are utilized in this experiment: Random Forest (RF), CatBoost (CB), LightGBM (LGB), and XGBoost (XGB). These classifiers are used for two different purposes in our experiment. The first purpose these classifiers are employed is to implement feature selection, and the second purpose these classifiers are employed is to train and test our models. In other words, a classifier could first be used to apply feature selection and then later that same (or different) classifier could be used to train and test our model.

All four Feature Selection Techniques (FSTs) from this study are supervised rankers that use our four classifiers: RF, CB, LGB, and XGB. These supervised rankers offer a simple feature selection technique in the sense that they are implemented by the feature importance lists from their respective RF, CB, LGB, and XGB Python libraries. While we selected the top 20 features from the supervised rankers for this study, future work could also experiment with different cutoff points for feature importance lists and vary the number of features to be included.

We trained and tested our models with the LGB and XGB classifiers, and future work can utilize additional classifiers for model training and testing (as well as evaluating additional FSTs). For all models trained and tested in this study, stratified 5-fold cross validation is used. Stratified refers to evenly splitting each training and test fold so that each class is proportionately weighted across all folds equally.

To account for randomness, each stratified 5-fold cross validation was repeated 10 times. Therefore, all of our AUC results are the mean values from 50 measurements (5 folds x 10 repeats). All classifiers from this experiment are implemented with Scikit-learn [16] and respective Python modules. Next, our four classifiers are described.

Random Forest is an ensemble of independent decision trees. Each instance is initially classified by every individual decision tree, and the instance is then finally classified by consensus among the individual trees (e.g., majority voting). Diversity among the individual decision trees can improve overall classification performance.

CatBoost is based on gradient boosting, and is essentially another ensemble of tree-based learners. It utilizes an ordered boosting algorithm [17] to overcome prediction shifting difficulties which are common in gradient boosting.

LightGBM, or Light Gradient Boosted Machine, is another learner based on Gradient Boosted Trees (GBTs) [18]. To optimize and avoid the need to scan every instance of a dataset when considering split points, LightGBM implements Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) algorithms [19].

XGBoost is another ensemble based on GBTs. To help determine splitting points, XGBoost utilizes a Weighted Quantile Sketch algorithm [20] to improve upon where split points should occur. Additionally, XGBoost employs a sparsity aware algorithm to help with sparse data to determine default tree directions for missing values.

Unless specified otherwise here, default values were utilized for hyper-parameters with the classifiers. For Random Forest, both n_estimators = 5 and max_depth = 6 were set to prevent overfitting. When using CatBoost both iterations = 4 and max_depth = 5 were assigned to prevent overfitting, while thread_count = 8 was set to take advantage of parallel processing functionality. Finally, with XGBoost both n_estimators = 4 and max_depth = 5 were set to prevent overfitting, and n_jobs = 8 was assigned to leverage parallel processing functionality. Also, objective = "binary logistic" was set to specify the objective function for binary classification with XGB.

C. Classification Performance Metrics

AUC is a metric which measures the area under the Receiver Operator Characteristic (ROC) curve. The ROC curve is a plot of the True Positive Rate (TPR) along the y-axis versus the False Positive Rate (FPR) along the x-axis. The area under this ROC curve corresponds to a numeric value ranging between 0.0 to 1.0, where an AUC value equal to 1.0 would correspond to a perfect classification system. An AUC value of 0.5 would represent a classifier system performing as well as a random guess, similar to flipping a coin. The AUC metric scores how effective a classification system is in terms of comparing TPR to FPR across all classification thresholds [21].

D. Feature Popularity

To evaluate feature popularity between the three different web attacks (BF, SQL, and XSS) and their feature subsets, we evaluate their feature similarity with two different techniques. The first technique is listing common features that appear most frequently between the three different web attacks, and the second technique is the Jaccard similarity to score feature popularity.

With the first technique, we determine which features are most popular between the three different web attacks. This is done by identifying the common features for which a certain number of supervised FSTs agree. For example, if the criteria is to have two FSTs agree, then all features where two of the FSTs agree are added to a "feature popularity list". To find features which are even more popular across the different attacks, the number of FSTs to agree can be increased and the resulting list of "popular features" will indicate features which are even more popular features (as the criteria for the number of FSTs to agree is increased). Later, the Results section of this paper provides a visual example of this.

For the second technique, the Jaccard similarity [22] scores pairs of sets of features. The Jaccard similarity of two sets is the ratio of the size of the intersection of the sets to the size of the union of the sets. That is, for two sets A and B, their Jaccard similarity is:

$$\frac{|A \cap B|}{|A \cup B|}$$

We calculate the Jaccard similarity for pairs of different web attacks, and present those Jaccard similarity scores in tables in the Results section. These tables give the reader a quantitative score indicating the degree to which different FSTs agree between pairs of different web attacks. In other words, the Jaccard similarity gives us a quantitative sense for how similar feature subsets are for one web attack dataset as compared to a different web attack dataset after an ensemble of FSTs has been applied to each dataset.

IV. Results

A. Feature Rankings

In this section, we first provide the feature rankings of the four different supervised rankers (XGB, RF, CB, and LGB) for our three different web attacks: Brute Force, SQL Injection, and XSS. These rankings enable us to visualize the output of the four supervised feature selection techniques. Only the top 20 features are included for each supervised ranking technique, and sometimes less than 20 features are included because less than 20 features were deemed important according to the ranker. For example, in some cases XGB and CB produced less than 20 features as those rankers may consider less than 20 features to be important.

In Table II, we present feature rankings for Brute Force web attacks from supervised feature selection techniques with CSE-CIC-IDS2018 data. Next, Table III includes feature rankings for SQL Injection web attacks from our four supervised feature selection techniques. Finally, Table IV lists feature rankings for XSS web attacks based on these same four supervised feature selection techniques.

Table V provides the classification performance results for our three different web attacks with the LightGBM and XGBoost classifiers in terms of AUC scores. Mainly, the classification performance is compared when no Feature Selection Technique (FST) ("None") has been applied as compared to our four supervised FSTs (an example of this would be the LGB classifier with "None" as the FST). Also, all four FSTs (CB, LGB, RF, XGB) are employed with our two classifiers (LGB and XGB). Additionally, the average of the four supervised FSTs AUC scores is indicated by "Avg4FST" for both of the LGB and XGB classifiers. The classification performance scores are similar before and after the supervised feature selection rankers are applied, even though the supervised rankers only include the top 20 features (at most) as compared to the full feature set.

In three out of the six cases from Table V, the average of the four supervised rankers performs better than when no feature selection is applied for the three web attacks. For the other three out of six cases, applying no feature selection performs better than the average of the four supervised rankers. Even though we have achieved similar classification performance by applying feature selection, we emphasize that the focus of this study is not comparing classification performance between feature selection techniques. Instead, our focus is to gain an understanding of feature popularity between the three different web attacks (BF, SQL, and XSS) in order to better explain important and common features between these three different web attacks.

B. Feature Popularity Results

Next, we provide results for feature popularity between the three subsets of different web attacks. First, we provide results with features that are common between the three different web attacks. Then, we quantitatively evaluate the feature popularity between these three web attacks with the Jaccard similarity.

In Table VI we provide the common features between the three different web attacks with respect to how many of their supervised rankers agree. The superscripts in this table correspond to the number of supervised rankers which agree for each of the attacks. For example, the superscript of "4" on the Fwd_IAT_Total feature for the Brute Force web attack

TABLE II
TOP 20 FEATURES FOR BRUTE FORCE WEB ATTACKS RANKED BY SUPERVISED-BASED FEATURE IMPORTANCE LISTS

XGBoost	Random Forest	CatBoost	LightGBM
Fwd_IAT_Min	Avg_Fwd_Segment_Size	Idle_Min	Fwd_Packet_Length_Mean
RST_Flag_Count	Total_Length_of_Bwd_Packets	Fwd_Packet_Length_Std	Flow_IAT_Min
Fwd_IAT_Total	Max_Packet_Length	ECE_Flag_Count	Fwd_IAT_Min
Flow_Packets_s	Fwd_Packet_Length_Mean	Flow_Bytes_s	Bwd_IAT_Min
Flow_IAT_Max	RST_Flag_Count	Fwd_IAT_Total	Flow_IAT_Std
act_data_pkt_fwd	Total_Length_of_Fwd_Packets	RST_Flag_Count	Fwd_Packet_Length_Std
Bwd_IAT_Mean	Flow_Bytes_s	Active_Max	Flow_Bytes_s
Bwd_IAT_Min	Bwd_IAT_Max	Fwd_Header_Length	Fwd_IAT_Total
	Subflow_Fwd_Bytes	Bwd_Packet_Length_Std	Bwd_Packets_s
	Fwd_Packet_Length_Std	Idle_Std	Bwd_IAT_Mean
	Avg_Bwd_Segment_Size	Min_Packet_Length	Flow_Packets_s
	Bwd_IAT_Mean	Active_Min	Fwd_IAT_Std
	Fwd_IAT_Std	Total_Length_of_Fwd_Packets	Fwd_Packets_s
	Idle_Mean	act_data_pkt_fwd	Flow_IAT_Max
	Subflow_Bwd_Packets		Total_Length_of_Fwd_Packets
	Bwd_IAT_Std		Idle_Std
	Fwd_IAT_Total		Fwd_Header_Length
	Bwd_Packet_Length_Mean		Flow_IAT_Mean
	Fwd_IAT_Min		Fwd_Packet_Length_Max
	Flow_Packets_s		Total_Backward_Packets

TABLE III
TOP 20 FEATURES FOR SQL INJECTION WEB ATTACKS RANKED BY SUPERVISED-BASED FEATURE IMPORTANCE LISTS

XGBoost	Random Forest	CatBoost	LightGBM
Max_Packet_Length	Bwd_Packet_Length_Max	ECE_Flag_Count	Flow_IAT_Min
Bwd_Packet_Length_Max	Flow_Packets_s	Fwd_IAT_Max	Flow_Bytes_s
Bwd_IAT_Min	Flow_IAT_Mean	PSH_Flag_Count	Fwd_IAT_Min
Total_Length_of_Fwd_Packets	RST_Flag_Count	Max_Packet_Length	Bwd_Packet_Length_Std
RST_Flag_Count	Fwd_Packet_Length_Max	Total_Fwd_Packets	Bwd_Packets_s
Packet_Length_Mean	Avg_Bwd_Segment_Size	Bwd_IAT_Max	Bwd_IAT_Min
Flow_IAT_Max	Total_Backward_Packets	Subflow_Bwd_Bytes	Fwd_Packet_Length_Mean
Fwd_Header_Length	Max_Packet_Length	Bwd_Packet_Length_Std	Fwd_IAT_Total
Fwd_Packet_Length_Std	ECE_Flag_Count	Avg_Bwd_Segment_Size	Flow_IAT_Mean
Bwd_Packets_s	Bwd_Packet_Length_Mean	Fwd_Packet_Length_Max	Fwd_Packet_Length_Max
Packet_Length_Std	Subflow_Bwd_Packets	Subflow_Bwd_Packets	Flow_Packets_s
Bwd_IAT_Max	Fwd_IAT_Total	Bwd_Header_Length	Flow_IAT_Max
Fwd_Packet_Length_Max	Bwd_Packet_Length_Std	Total_Length_of_Bwd_Packets	Fwd_IAT_Mean
Flow_Bytes_s	Fwd_IAT_Max	Down_Up_Ratio	Total_Length_of_Fwd_Packets
Bwd_IAT_Mean	Fwd_Packet_Length_Mean		Fwd_Packet_Length_Std
Fwd_IAT_Std	Flow_IAT_Std		Fwd_Packets_s
Fwd_IAT_Min	min_seg_size_forward		Fwd_IAT_Std
Fwd_Packet_Length_Mean	Packet_Length_Variance		Flow_IAT_Std
Flow_Packets_s	Packet_Length_Std		Fwd_IAT_Max
Fwd_IAT_Mean	Flow_IAT_Max		Packet_Length_Variance

indicates that all four of the supervised rankers agree that this feature is common between our top 20 feature importance lists of Tables II to IV.

Based on Table VI, we are able to generate Table VII which is a feature popularity list. It indicates features that are common to all three web attacks where two out of four

TABLE IV
TOP 20 FEATURES FOR XSS WEB ATTACKS RANKED BY SUPERVISED-BASED FEATURE IMPORTANCE LISTS

XGBoost	Random Forest	CatBoost	LightGBM
Bwd_Packet_Length_Mean	Bwd_Packet_Length_Max	Flow_Bytes_s	Flow_IAT_Min
Total_Backward_Packets	Bwd_Packet_Length_Mean	Max_Packet_Length	Fwd_IAT_Min
Total_Length_of_Fwd_Packets	Max_Packet_Length	Destination_Port	Flow_IAT_Mean
FIN_Flag_Count	Fwd_Packets_s	Fwd_IAT_Std	Fwd_Packets_s
Total_Fwd_Packets	Packet_Length_Std	Flow_IAT_Mean	Flow_Bytes_s
Flow_IAT_Min	ECE_Flag_Count	Bwd_IAT_Max	Flow_Packets_s
Bwd_Packet_Length_Max	Bwd_IAT_Std	Bwd_IAT_Mean	Flow_IAT_Max
Bwd_IAT_Mean	Idle_Min	Avg_Bwd_Segment_Size	Bwd_Packets_s
Flow_Packets_s	Avg_Bwd_Segment_Size	Total_Length_of_Fwd_Packets	Fwd_IAT_Total
Down_Up_Ratio	Subflow_Bwd_Packets	Idle_Std	Fwd_IAT_Std
Packet_Length_Std	Fwd_IAT_Total	Bwd_IAT_Std	Fwd_IAT_Max
Flow_Bytes_s	Flow_IAT_Mean	Flow_IAT_Max	Flow_IAT_Std
ACK_Flag_Count	Flow_IAT_Max	Active_Min	Bwd_IAT_Std
RST_Flag_Count	Fwd_IAT_Min	act_data_pkt_fwd	Fwd_IAT_Mean
Bwd_IAT_Max	Bwd_Header_Length	min_seg_size_forward	Bwd_Packet_Length_Min
Idle_Max	Flow_IAT_Min		Fwd_Header_Length
Fwd_IAT_Std	Active_Mean		Bwd_IAT_Min
Packet_Length_Mean	Idle_Max		Packet_Length_Std
Active_Min	Packet_Length_Mean		Bwd_IAT_Total
Bwd_Packets_s	Fwd_IAT_Max		Packet_Length_Variance

TABLE V
CLASSIFICATION PERFORMANCE OF SUPERVISED FEATURE SELECTION TECHNIQUES FOR 3 WEB ATTACKS (BF, SQL, AND XSS)

Classifier	FST	BF : AUC	SQL : AUC	XSS : AUC
LGB	None	0.94182	0.94017	0.94449
LGB	CB	0.94082	0.93002	0.97216
LGB	LGB	0.93827	0.93415	0.94526
LGB	RF	0.93724	0.92212	0.94685
LGB	XGB	0.93209	0.9291	0.94621
LGB	Avg4FST	0.93711	0.92885	0.95262
XGB	None	0.93797	0.899	0.93096
XGB	CB	0.93045	0.91162	0.96828
XGB	LGB	0.93879	0.90397	0.93178
XGB	RF	0.93712	0.90792	0.93612
XGB	XGB	0.91722	0.9118	0.93065
XGB	Avg4FST	0.93090	0.90883	0.94171

supervised rankers agree. In other words, the seven features from Table VII appear in feature importance lists for all three web attacks where at least two of four supervised FSTs agree.

Continuing this example, we extend our feature popularity concept to a more restrictive scenario where three out of four supervised rankers must agree. Table VIII indicates the feature popularity list where three out of four supervised rankers must agree for each web attack, but they only need to agree for two of the three web attacks. No features are common to all three web attacks where three out of four supervised rankers agree, and so we had to relax the criteria and allow common features between only two of three web attacks where three out of their four supervised rankers agree.

Table IX provides Jaccard similarity scores between our three different web attacks where two out of four supervised feature selection techniques agree. The SQL Injection and XSS web attacks have the most features in common according to their Jaccard similarity score of 0.44118 where two out of four supervised rankers agree. Next, the SQL Injection and Brute Force web attacks have the second most features in common among their feature subsets where two of four supervised rankers agree with a Jaccard similarity score of 0.36667. The Brute Force and XSS web attacks have the lowest Jaccard similarity score of 0.26667 where two out of four supervised rankers agree.

Finally, Table X indicates Jaccard similarity scores for the three web attacks where three out of four supervised rankers agree. Again, the SQL Injection and XSS web attacks have the highest Jaccard similarity score when three out of four supervised rankers agree. The SQL/BF and XSS/BF feature subsets both achieve the same Jaccard similarity scores where three out of four supervised rankers agree. Based on the Jaccard similarity scores from both Tables IX and X, we can see the SQL Injection and XSS web attacks have the most features in common from their four supervised rankers.

V. CONCLUSION

Feature popularity is a novel concept that we introduce in this study, and we implement it with the CSE-CIC-IDS2018 dataset and the following three web attacks: Brute Force, SQL Injection, and XSS. Feature importance lists with the top 20 features are generated for these three web attacks and the

TABLE VI
COMMON FEATURES FROM 4 SUPERVISED RANKERS FOR 3 DIFFERENT WEB ATTACKS

Brute Force	SQL Injection	XSS
act_data_pkt_fwd [2]	Avg_Bwd_Segment_Size [2]	Active_Min [2]
Bwd_IAT_Min [2]	Bwd_Packet_Length_Max [2]	Avg_Bwd_Segment_Size [2]
Flow_IAT_Max [2]	Bwd_IAT_Min [2]	Bwd_IAT_Max [2]
Fwd_Header_Length [2]	Bwd_Packets_s [2]	Bwd_IAT_Mean [2]
Fwd_IAT_Std [2]	Bwd_IAT_Max [2]	Bwd_Packet_Length_Max [2]
Fwd_Packet_Length_Mean [2]	ECE_Flag_Count [2]	Bwd_Packet_Length_Mean [2]
Idle_Std [2]	Flow_Bytes_s [2]	Bwd_Packets_s [2]
Bwd_IAT_Mean [3]	Flow_IAT_Mean [2]	Flow_Packets_s [2]
Flow_Bytes_s [3]	Flow_IAT_Std [2]	Fwd_IAT_Max [2]
Flow_Packets_s [3]	Fwd_IAT_Mean [2]	Fwd_IAT_Min [2]
Fwd_IAT_Min [3]	Fwd_IAT_Min [2]	Fwd_IAT_Total [2]
Fwd_Packet_Length_Std [3]	Fwd_IAT_Std [2]	Fwd_Packets_s [2]
RST_Flag_Count [3]	Fwd_IAT_Total [2]	Idle_Max [2]
Total_Length_of_Fwd_Packets [3]	Fwd_Packet_Length_Std [2]	Max_Packet_Length [2]
Fwd_IAT_Total [4]	Packet_Length_Std [2]	Packet_Length_Mean [2]
	Packet_Length_Variance [2]	Total_Length_of_Fwd_Packets [2]
	RST_Flag_Count [2]	Packet_Length_Std [3]
	Subflow_Bwd_Packets [2]	Bwd_IAT_Std [3]
	Total_Length_of_Fwd_Packets [2]	Flow_IAT_Mean [3]
	Bwd_Packet_Length_Std [3]	Flow_IAT_Max [3]
	Flow_IAT_Max [3]	Flow_IAT_Min [3]
	Flow_Packets_s [3]	Flow_Bytes_s [3]
	Fwd_IAT_Max [3]	Fwd_IAT_Std [3]
	Fwd_Packet_Length_Mean [3]	
	Max_Packet_Length [3]	
	Fwd_Packet_Length_Max [4]	

TABLE VII
FEATURE POPULARITY WHERE 2 OUT OF 4 SUPERVISED RANKERS AGREE, FOR 3 OUT OF 3 WEB ATTACKS (BF, SQL, AND XSS)

Flow_Bytes_s
Flow_IAT_Max
Flow_Packets_s
Fwd_IAT_Min
Fwd_IAT_Std
Fwd_IAT_Total
Total_Length_of_Fwd_Packets

TABLE VIII
FEATURE POPULARITY WHERE 3 OUT OF 4 SUPERVISED RANKERS AGREE, FOR 2 OUT OF 3 WEB ATTACKS (BF, SQL, AND XSS)

Flow_Bytes_s
Flow_IAT_Max
Flow_Packets_s

following four supervised rankers: RF, CB, LGB, and XGB. After applying these four FSTs, classification performance is relatively similar for these feature subsets with the top 20 features as compared to the full feature sets.

Ensemble FSTs are utilized to generate feature popularity lists, which allows us to more easily visualize and understand the most important common features across the three different web attacks. Our feature popularity list with the most restrictive criteria (where three out of four supervised rankers must agree for at least two of the three web attacks), produces

TABLE IX
JACCARD SIMILARITIES WHERE 2 OUT OF 4 SUPERVISED RANKERS AGREE

	XSS	BF	SQL
XSS	1.00000	0.26667	0.44118
BF	0.26667	1.00000	0.36667
SQL	0.44118	0.36667	1.00000

TABLE X
JACCARD SIMILARITIES WHERE 3 OUT OF 4 SUPERVISED RANKERS AGREE

	XSS	BF	SQL
XSS	1.00000	0.07143	0.07692
BF	0.07143	1.00000	0.07143
SQL	0.07692	0.07143	1.00000

the following three most popular features: Flow_Bytes_s, Flow_Packets_s, and Flow_IAT_Max. All three of these most popular features are based on time with respect to flow attributes. Flow_Bytes_s is the number of bytes per second in a flow, Flow_Packets_s is the number of packets per second in a flow, and Flow_IAT_Max is the maximum time between two flows.

The Jaccard similarity quantitatively indicates the SQL Injection and XSS feature subsets are most similar to each other among the three web attacks when considering their most popular features. Overall, we can also see the SQL Injection web attacks are the most similar to the other two web attacks from inspecting the Jaccard similarity tables. Likewise, we can also learn the Brute Force web attacks are the least similar to the other two web attacks when considering the most popular features with these Jaccard similarity tables.

While higher Jaccard similarity scores may be desired to indicate cyberattacks which are related to each other in terms of common features, readers should not dismiss lower scores as sometimes only a few popular features may sufficiently establish models which are good enough. This is the beauty of the feature popularity framework, and we can adjust our popularity criteria to be more or less restrictive in two dimensions when building our "ensemble of ensembles". In our first "agreement" dimension, we can adjust the number of FSTs that must agree. We can also adjust the number of cyberattack datasets which must agree in our second dimension. In this manner, we can tune our feature popularity lists until desired classification performance is achieved. This is only an introductory work proposing our feature popularity concepts, and future work can evaluate classification performance over different feature popularity thresholds for various cyberattacks.

We must emphasize that employing Jaccard similarities is a powerful framework when comparing popular features across different attack datasets, as we could not actually realize how similar or different the feature subsets from attacks were from each other even after carefully studying the many feature lists we generated through ensemble FSTs. Not only are these feature popularity techniques beneficial for more explainable and easier-to-implement models in the real-world. They also open the door to a bounty of various research topics. For example, with an enormous variety of different types of attacks, we might find one model cannot reasonably accommodate dozens or even hundreds different types of cyberattacks. With feature popularity, we may be able to systematically deploy different models to accommodate different families of attacks where their feature popularity subsets are very divergent from each other.

Future work can include additional types of attacks, FSTs, classifiers, and datasets. Additionally, different cutoff points for the "Top N" features of the feature importance lists could be investigated. Finally, future research could develop these feature popularity frameworks in an automated manner through open-source tools to allow easier exploration of popular features across different cyberattacks or datasets.

REFERENCES

[1] J. Young, "Us ecommerce sales grow 14.9% in 2019," accessed: 2020-11-28. [Online]. Available: https://www.digitalcommerce360.com/article/us-ecommerce-sales/

[2] R. Wald, F. Villanustre, T. M. Khoshgoftaar, R. Zuech, J. Robinson, and E. Muharemagic, "Using feature selection and classification to build effective and efficient firewalls," in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*. IEEE, 2014, pp. 850–854.

[3] M. M. Najafabadi, T. M. Khoshgoftaar, and N. Seliya, "Evaluating feature selection methods for network intrusion detection with kyoto data," *International Journal of Reliability, Quality and Safety Engineering*, vol. 23, no. 01, p. 1650001, 2016.

[4] R. Zuech and T. M. Khoshgoftaar, "A survey on feature selection for intrusion detection," in *Proceedings of the 21st ISSAT International Conference on Reliability and Quality in Design*, 2015, pp. 150–155.

[5] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, and A. Alonso-Betanzos, "Ensemble feature selection: homogeneous and heterogeneous approaches," *Knowledge-Based Systems*, vol. 118, pp. 124–139, 2017.

[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *ICISSP*, 2018, pp. 108–116.

[7] "Cicids2017 dataset," accessed: 2020-08-28. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[8] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data," *Journal of Big Data*, vol. 7, no. 1, pp. 1–19, 2020.

[9] M. Sarhan, S. Layeghy, and M. Portmann, "An explainable machine learning-based network intrusion detection system for enabling generalisability in securing iot networks," *arXiv preprint arXiv:2104.07183*, 2021.

[10] J. L. Leevy, J. Hancock, R. Zuech, and T. M. Khoshgoftaar, "Detecting cybersecurity attacks across different network features and learners," *Journal of Big Data*, vol. 8, no. 1, pp. 1–29, 2021.

[11] Q. R. S. Fitni and K. 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)*. IEEE, 2020, pp. 118–124.

[12] M. Beechey, K. G. Kyriakopoulos, and S. Lambotharan, "Evidential classification and feature selection for cyber-threat hunting," *Knowledge-Based Systems*, vol. 226, p. 107120, 2021.

[13] Y. Hua, "An efficient traffic classification scheme using embedded feature selection and lightgbm," in *2020 Information Communication Technologies Conference (ICTC)*. IEEE, 2020, pp. 125–130.

[14] J. Prusa, T. M. Khoshgoftaar, D. J. Dittman, and A. Napolitano, "Using random undersampling to alleviate class imbalance on tweet sentiment data," in *2015 IEEE international conference on information reuse and integration*. IEEE, 2015, pp. 197–202.

[15] R. Zuech, J. Hancock, and T. M. Khoshgoftaar, "Investigating rarity in web attacks with ensemble learners," *Journal of Big Data*, vol. 8, no. 1, pp. 1–27, 2021.

[16] "Scikit-learn website," accessed: 2021-01-30. [Online]. Available: https://scikit-learn.org/stable/

[17] J. T. Hancock and T. M. Khoshgoftaar, "Catboost for big data: an interdisciplinary review," *Journal of big data*, vol. 7, no. 1, pp. 1–45, 2020.

[18] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[19] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[20] J. T. Hancock and T. M. Khoshgoftaar, "Gradient boosted decision tree algorithms for medicare fraud detection," *SN Computer Science*, vol. 2, no. 4, pp. 1–12, 2021.

[21] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[22] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of jaccard coefficient for keywords similarity," in *Proceedings of the international multiconference of engineers and computer scientists*, vol. 1, no. 6, 2013, pp. 380–384.