Artificial Intelligence for Cybersecurity Applications

CSI 5388

Project Description & Literary Review

Project: Deliverable #2

Authors
Alexandra Sklokin (300010511)
Pranav Pawar (300333613)
Anisha Deochake (300369581)
Aryan Gulati (300351365)

Professor
Dr. Paula Branco

October 23, 2023

Contents

1 Project Description 1
1.1 Motivation . . . 1
1.2 Methodology . . . 1
2 Literary Review 3
2.1 Supervised Feature Selection . . . 3
2.1.1 Filter Methods . . . 3
2.1.2 Wrapper Methods . . . 4
2.1.3 Embedded Methods . . . 4
2.2 Ensembles . . . 5
2.3 DDoS Attacks and Machine Learning . . . 5
3 References 7

1 Project Description

1.1 Motivation

Within the field of Machine Learning (ML), Feature Selection is the process of selecting a subset of features, from the available data, for building a model. There are two primary reasons for removing a feature: it is unrelated to the target variables ($x_i \sim y$) or is mutually correlated to another variable ($x_i \sim x_j$). This is an important subtask of the data preprocessing phase of training a model, and has many benefits:

- **Data Dimensionality**: Fewer features will decrease the dimensionality of the dataset.
- **Interpretability and Explainability**: Fewer features lead to better interpretability of the data, as well as the resultant model. For example, fewer features will create a simpler and smaller Decision Tree model.
- **Efficiency**: Faster training and prediction time. It is more resource-efficient to store, process, and analyze a smaller dataset.
- **Overfitting**: Reduces the risk of overfitting, by allowing for simpler models less likely to fit noise. Overfitting can also occur from mutually correlated features, thus by performing Feature Selection we can remove redundant features.
- **Model Generalizability**: A simpler model, trained on a concise feature set, might generalize better for unseen data.
- **Domain Knowledge Integration**: Easier and cheaper for experts to incorporate domain knowledge into the ML pipeline.
- **Data Privacy**: By minimizing the feature set, can aid in removing sensitive or personal data.

Feature Selection is particularly important in the field of cybersecurity, where datasets such as network logs, have many irrelevant, or noisy features. These datasets grow large due to the nature of digital traffic, where every connection can represent a single data sample, accompanied by metadata. Additionally, the challenges with distributed learning intensify the problem's complexity, by introducing redundancy and increasing the dataset size and feature set.

For the reasons listed above, Feature Selection is an unavoidable stage of ML for cybersecurity. It becomes imperative to streamline these datasets and identify the most informative features for effective threat detection. In this project, we will propose a new ensemble Feature Selection technique, effective for detecting Distributed Denial of Services (DDoS) attacks, which might have improved performance over other traditional methods.

1.2 Methodology

In this project, we propose a new Feature Selection technique called the **STable Ensemble Feature Ranking (STEF-Rank)**. In essence, **STEF-Rank** is a bagging ensemble technique, where multiple Feature Selection techniques are trained on the dataset. Each 'weak' Feature Selection technique will be applied multiple times on resamplings of the dataset (thus introducing stability). By introducing both resampling and ensembles, we hope our method is able to outperform traditional Feature Selection techniques. Pseudocode from **Algorithm 1** can be used to better understand our novel Feature Selection technique.

Here is an example scenario of **STEF-Rank**: on a particular dataset D, we will create 5 (n) subsets, using resampling, and apply the ANOVA to get 5 different feature sets. For each combination of resampling and Feature Selection technique, each feature of the dataset is given a rank out of 20 ($m * n$). We repeat this with the 3 more 'weak' Feature Selection techniques (m=4). The ensemble performs the addition of these ranks, and sorts features by importance. Finally, features with a ranking above the threshold (ex. 0.5) are returned.

We will use the following three DDoS Datasets:

- **CICEV2023**: Large dataset of simulated DDoS attacks on EV charging infrastructure, with 4 types of attack scenarios [18].
- **CIC-DDoS2019**: Simulated dataset of common DDoS attack types (Reflection and Exploitation) on networks, over two days. This dataset has already undergone feature extraction and contains many features, allowing for a good investigation of Feature Selection techniques [16].

- **CSE-CIC-IDS2018**: Dataset containing different types of cybersecurity attacks over several days, and many features. This dataset has undergone feature extractions and contains many features, relating to different types of cybersecurity attacks, and not necessarily to DDoS [15].

During the implementation phase, we will follow the following pipeline:

- **Data Preprocessing**: Data cleaning, removing time-dependent variables, normalization, balancing (oversampling or SMOTE), etc.
- **Baseline and Novel Feature Selection**: Using traditional filter/wrapper methods and the **STEF-Rank** technique to general subsets of the feature set.
- **Hyperparameter Tuning**: Before training, we will perform hyperparameter tuning on the RandomForest/XGBoost models. Note that hyperparameter tuning must be done for each different feature subset.
- **Classification & Validation**: Perform classification with the RandomForest/XGBoost models, using 5-Fold Cross-Validation.
- **Performance Metrics**: F1-Score, Precision-Recall AUC, ROC-AUC, training-time, etc.
- **Comparison to Baseline**: Tests of statistical Significance: Friedman, Wilcoxon Signed Rank, Nemenyi Post Hoc test, bonferoni-dunn post hoc test, etc.

Algorithm 1 STEF-Rank (D, FS, n, $threshold$)

```
# Input: dataset $D$; set of Feature Selection techniques $FS$; number of resamplings $n$; feature selection $threshold$
for each $FS_i$ in $FS$ do
  $SubD \leftarrow Resampling(D, n)$ # create n subsets using resampling
  for each $SubD_j$ in $SubD$ do
    $feature\_selection_j \leftarrow FS_i(SubD_j)$ # apply 'weak' feature selector to resampling
  end for
  for each $feature$ in $D.features$ do
    $feature\_rank_i[feature] \leftarrow CountOccurances(feature\_selection, feature) \div (n * m)$ # rank matrix for the 'weak' learner
  end for
end for
for each $feature$ in $D.features$ do
  $ensemble\_ranking[feature] \leftarrow Sum(feature\_rank[feature])$ # ensemble ranking
end for
$best\_features \leftarrow ensemble\_ranking.iloc(ranking \geq threshold)$ # features with high rank
return best_features # Output: dictionary with features as keys and rankings as values
```

2 Literary Review

2.1 Supervised Feature Selection

Feature Selection is a pivotal research domain in ML, thus there have been countless techniques proposed to optimize the choice of feature set. The following section is a survey of the prominent supervised Feature Selection techniques, categorized as filter, wrapper, and ensemble methods. Through this literary review, we aim to describe the use, benefits and limitations of each method.

Aside from these supervised techniques, which are the primary focus of this research project, there are other types of Feature Selection frameworks. **Figure 1** shows a exhaustive taxonomy of various Feature Selection techniques.

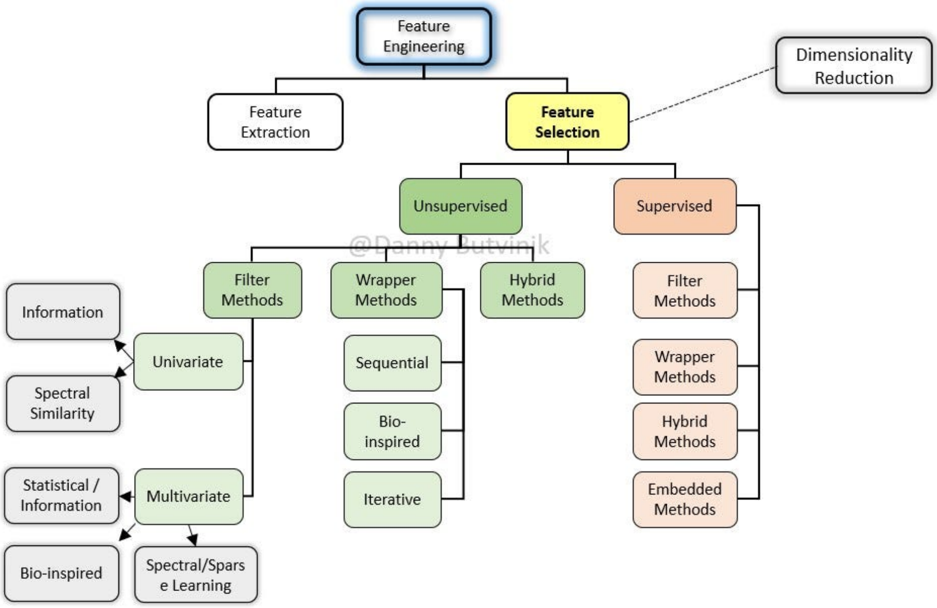

Figure 1: Taxonomy of Feature Selection Techniques [6]

2.1.1 Filter Methods

Filter methods have proven to be valuable in reducing the dimensionality of feature sets and enhancing the performance of intrusion detection models. These methods help identify features that are most relevant for discriminating between DDoS attacks and benign samples. The main limitation of Filter methods is the sensitivity to thresholds; the choice of a threshold is arbitrary and affects the selection outcome. On the other hand, Filter methods are simple, explainable, and easy to implement, thus suitable for high-dimensional datasets.

- **Analysis of Variance (ANOVA):** is a Filter Method used to check that the means of two or more groups are significantly different from each other. It assumes the hypotheses (H_0) is 'Means of all groups are equal'; and (H_a) is 'At least one mean of the groups is different'. ANOVA can be applied to find the most statistically significant features in a dataset, as well as the confidence (*p-value*). [4]
- **Pearson's Correlation Coefficient:** serves to quantify the strength and direction of the linear relationship between two continuous variables, providing crucial insight into their statistical association. The coefficient offers a normalized score that ranges between -1 and 1, representing the spectrum of correlation from a perfect negative correlation to a perfect positive correlation. [12]
- **Chi-Squared test:** is particularly useful in categorical features that are statistically independent of target features. [2]
- **Mutual Information:** measures the statistical dependence between two variables. High mutual information values indicate that the feature and the target variable share a significant amount of information. This versatile method can be applied to continuous and discrete features. [5]

- **SelectKBest:** is a feature selection method provided by the *scikit-learn* library in Python. SelectKBest selects the top k most important features from a dataset based on statistical tests, such as chi-squared (for categorical features), ANOVA F-test (for numeric features), and mutual information (for both numeric and categorical features). The features are selected based on their individual scores and ranked in descending order. [1]

Aside from the aforementioned techniques, there are many others, such as the information gain, fished score, relief, variance threshold, dispersion ratio, distance correlation, T-test, etc.

2.1.2 Wrapper Methods

Wrapper methods measure the importance of a feature based on its usefulness while training the ML model. In comparison to filter methods, wrapper methods are computationally more expensive, but provide many benefits, such as interacting with the classifier; more comprehensive search of feature space; considering feature dependencies; and better generalizability.

- **Forward Selection & Backward Elimination:** Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature that best improves our model until the addition of a new variable does not improve the performance of the model. In backward elimination, we start with all the features and remove the least significant feature at each iteration which improves the performance of the model. [11]
- **Sequential Feature Selection (SFS):** a family of greedy search algorithms that are used to reduce an initial d-dimensional feature space to a k-dimensional feature subspace where k is less than d. [14]
- **Genetic Algorithms:** uses genetic algorithms can optimize the selected features. A genetic algorithm-based Feature Selection method with a wrapper-embedded technique is expected to produce an effective feature subset. [19]
- **Simulated Annealing:** is a type of controlled random search in which the new candidate feature subset is arbitrarily chosen dependent on the state of the system. A data set can be produced to measure the performance difference between the presence and absence of each predictor after a sufficient number of iterations. [9]

Aside from the aforementioned techniques, there are many others, such as the Floating Search Methods, Exhaustive Feature Selection, etc.

2.1.3 Embedded Methods

Embedded methods for Feature Selection are algorithms that incorporate Feature Selection as part of the model training process. These methods are computationally efficient and able to account for dependencies between features

- **Recursive Feature Elimination (RFE):** wrapper-like greedy optimization algorithm that aims to find the best-performing feature subset. It repeatedly creates models and keeps aside the best or the worst performing feature at each iteration. It constructs the next model with the left features until all the features are exhausted. It then ranks the features based on the order of their elimination. [11]
- **Embedding-based Unsupervised Feature Selection (EUFS):** is a method which integrates Feature Selection into a clustering algorithm using sparse learning. The main advantage of EUFS, is the simultaneity of Feature Selection and model training, simplifying the workload of both tasks. This method struggles with scalability issues and overfitting for larger datasets. [17]
- **Classification and Regression Tree (CART):** is a decision tree algorithm capable of Feature Selection, by constructing a binary tree where nodes represent the features and edges determine the splitting criteria. CART's advantages are remaining invariant to the monotone transformation of the data, effective handling of outliers, and allowing for the repeated use of features. CART is also extremely interpretable and visualizable. CART may overlook complex interactions between features due to its individual evaluation approach. [7]

2.2 Ensembles

In ML, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent 'weak' learning algorithms alone. Although very computationally demanding, ensembles introduce many benefits:

- **Performance:** Ensemble methods often outperform the 'weak' constituent models.
- **Overfitting:** Introduce stability, through repeated training, and robustness to overfitting, by reducing variance and/or bias.
- **Handling Noisy Data:** More tolerant to noisy data and outliers.
- **Generalizability:** By reducing overfitting, ensembles can attain better generalizability on unseen data.

Figure 2 shows the two main types of ensemble methods:

- **Bagging:** is a technique for reducing the variance of ML models by combining the results from multiple independent models. Once each model is trained, some type of aggregation technique, such as summation, averaging or voting, is applied. [10]
- **Boosting:** is able to significantly enhance the performance of ML models by sequentially combining multiple weak learners, to reduce bias and variance. [10]

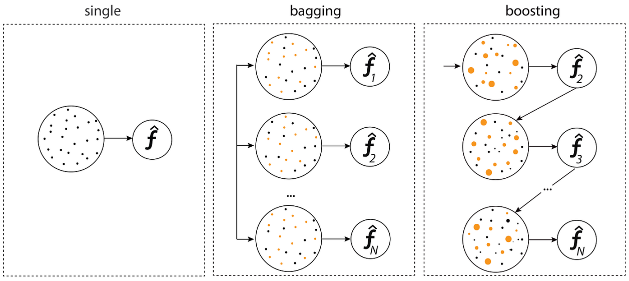

Figure 2: Bagging and Boosting Ensemble Techniques [13]

2.3 DDoS Attacks and Machine Learning

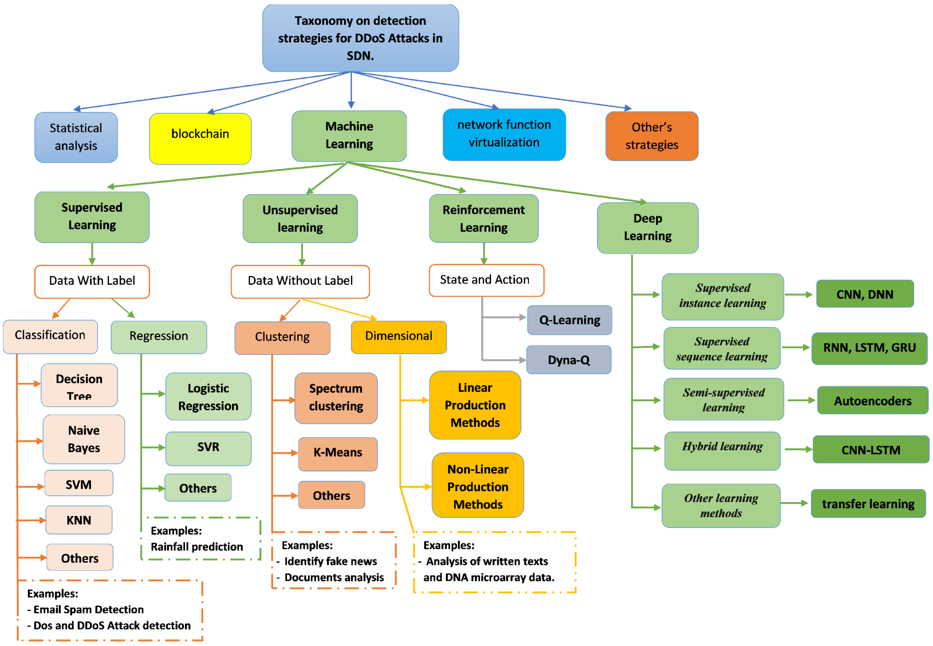

Figure 3: Taxonomy of DDoS Detection Machine Learning Techniques [3]

Distributed Denial of Services (DDoS) occurs when multiple (potentially compromised or malicious) systems overwhelm a single device or system, causing a Denial of Service (DoS), with the intent to make a networked service unavailable. The most common types of DDoS attacks are:

- **SYN Flooding:** occurs when the attacker(s) exploits a vulnerability of TCP, by sending many SYN packets from false sources. The target device or system responds with a SYN/ACK and awaits an ACK packet response that does not arrive. [8]
- **ICMP Flooding:** occurs when the attacker(s) sends a large quantity of ICMP packets with fake return addresses. The target device or system is overwhelmed and unable to respond to legitimate requests. [8]
- **UDP Flooding:** occurs when the attacker(s) sends UDP datagrams, to the false ports of the target. The target device continues to respond with ICMP 'destination unreachable' packets. [8]

Using ML algorithms, network administrators can perform early detection of attacks on large quantities of network traffic logs. By leveraging supervised and unsupervised learning, it is possible to identify patterns and take preventative measures, such as creating automated responses for various types of DDoS attacks. With the ever-evolving cybersecurity landscape, traditional identification rules and practices can no longer keep up with sophisticated attackers. **Figure 3** shows a complete taxonomy of the various ML techniques being employed for DDoS detection.

3 References

[1] *1.13 Feature Selection.* URL: https://scikit-learn.org/stable/modules/feature_selection.html#univariate-feature-selection.

[2] *15.8 - Chi-Square Distributions.* URL: https://online.stat.psu.edu/stat414/lesson/15/15.8.

[3] Tariq Emad Ali, Yung-Wey Chong, and Selvakumar Manickam. “Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review”. In: *Applied Sciences* 13.5 (2023). ISSN: 2076-3417. DOI: 10.3390/app13053183. URL: https://www.mdpi.com/2076-3417/13/5/3183.

[4] Adel Binbusayyis and Thavavel Vaiyapuri. “Comprehensive analysis and recommendation of feature evaluation measures for intrusion detection”. In: 2020. DOI: 10.1016/j.heliyon.2020.e04262.

[5] Adel Binbusayyis and Thavavel Vaiyapuri. “Identifying and Benchmarking Key Features for Cyber Intrusion Detection: An Ensemble Approach”. In: *IEEE Access* 7 (2019), pp. 106495–106513. DOI: 10.1109/ACCESS.2019.2929487.

[6] Danny Butvinik. “Feature Selection — Exhaustive Overview”. In: *Medium* (). URL: https://medium.com/analytics-vidhya/feature-selection-extended-overview-b58f1d524c1c.

[7] Qing Liu Haoyue Liu MengChu Zhou. “An Embedded Feature Selection Method for Imbalanced Data Classification”. In: 2019. URL: https://ieeexplore.ieee.org/abstract/document/8677302.

[8] Priyanka Kamboj et al. “Detection techniques of DDoS attacks: A survey”. In: *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)*. 2017, pp. 675–679. DOI: 10.1109/UPCON.2017.8251130.

[9] Max Kuhn and Kjell Johnson. “Feature Engineering and Selection: A Practical Approach for Predictive Models”. In: 2021. URL: https://bookdown.org/max/FES/.

[10] Suyash Kumar, Prabhjot Kaur, and Anjana Gosain. “A Comprehensive Survey on Ensemble Methods”. In: *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*. 2022, pp. 1–7. DOI: 10.1109/I2CT54291.2022.9825269.

[11] Sauravkaushik. “Introduction to Feature Selection methods with an example (or how to select the right variables?” In: URL: https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/.

[12] Philip Sedgwick. *Pearson’s correlation coefficient.* 2021. DOI: https://doi.org/10.1136/bmj.e.

[13] Aleyna Şenozan. “Ensemble: Boosting, Bagging, and Stacking Machine Learning”. In: *Medium* (). URL: https://medium.com/@senozanAleyna/ensemble-boosting-bagging-and-stacking-machine-learning-6a09c31df778.

[14] “SequentialFeatureSelector: The popular forward and backward feature selection approaches (including floating variants)”. In: URL: https://rasbt.github.io/mlxtend/user_guide/feature_selection/SequentialFeatureSelector/.

[15] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”. In: *International Conference on Information Systems Security and Privacy*. 2018. URL: https://api.semanticscholar.org/CorpusID:4707749.

[16] Iman Sharafaldin et al. “Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy”. In: *2019 International Carnahan Conference on Security Technology (ICCST)*. 2019, pp. 1–8. DOI: 10.1109/CCST.2019.8888419.

[17] Huan Liu Suhang Wang Jiliang Tang. “Embedded Unsupervised Feature Selection”. In: 2015. URL: https://cdn.aaai.org/ojs/9211/9211-13-12739-1-2-20201228.

[18] Kim Yoonjib, Hakak Saqib, and Ali A. Ghorbani. *DDoS Attack Dataset (CICEV2023) against EV Authentication in Charging Infrastructure.* Aug. 2023.

[19] Kridanto Surendro Yuda Syahidin Nur Ulfa Maulidevi. “FEATURE SELECTION METHOD BASED ON GENETIC ALGORITHM WITH WRAPPER-EMBEDDED TECHNIQUE FOR MEDICAL RECORD CLASSIFICATION”. In: *ICSCA*. 2023. URL: https://dl.acm.org/doi/abs/10.1145/3587828.3587856.