Detecting Anomalies in Network Traffic Using a Hybrid of Linear-based and Tree-based Feature Selection Approaches

Armaya'u Zango Umar
Dept. Soft. Eng. and Cybersec.
Al-Qalam University Katsina
Katsina, Nigeria
azumar@auk.edu.ng

Yakubu Galadima Ibrahim
Dept. of Computer Science
Al-Qalam University Katsina
Katsina, Nigeria
yakubuigaladima@auk.edu.ng

Ahmad Ndanusa
Dept. of Computer Sci.
University of Abuja
Abuja, Nigeria
abndanusa@gmail.com

***Abstract*—Intrusion detection is a critical component in ensuring the security of services deployed on the internet. Conventional intrusion detection systems have proved ineffective in stopping sophisticated attacks. Recently, intrusion detection based on machine learning have demonstrated the potential of learning normal and anomalous pattern of network traffic. However, due to the large number of redundant features in the training dataset, intrusion detection based on machine learning could, unfortunately, wronglly classify normal traffic as abnormal - false positive. This research paper proposes a novel hybrid feature selection strategy to address the challenges of network anomalies and malicious traffic detection. By combining linear-based and tree-based feature selection approaches, a powerful feature representation is created, which enhances the accuracy and efficiency of intrusion detection systems (IDS). The approach is evaluated through experiments conducted on diverse datasets, demonstrating impressive results with accuracy levels exceeding 99% in all cases. The proposed approach significantly reduces the number of features to up to 78% while maintaining or improving detection accuracy.**

***Index Terms*—Feature selection, Detecting anomalies, Malicious traffic, Intrusion detection system**

I. Introduction

Pervasive computing devices and digital infrastructures have revolutionized the way we deploy and access wide range of services over the internet. From internet banking and electronic commerce to smart living enabled by the Internet of Things (IoT). These services have become integral parts of our daily lives. However, with their increased availability, there comes a significant risk for cyber security. These risks can be both tangible, such as the risk of financial loss, and intangible, such as the risk of compromising our privacy. [1]. Number of cyber-attacks can be lunch using internet traffic disguised as a normal request for legitimate services [16] but generally aim at intruding the system hosting the service. Intrusion is the malicious act of compromising privacy or confidentiality of other service users. Intrusion can also disrupt the availability of services, denying legitimate users access [11]. To counter against the threats, academia and industry have developed many intrusion detection systems (IDS). IDS monitor the traffic on the network and detect anomalies and handle it based on the predefined response policy such as blocking the traffic or alerting administrators to take manual action. Traditionally, measures such as Access Control and Firewall were being used as IDS. Due to increasing sophistication of cyber-attacks, malicious traffic has succeeded in in bypassing such conventional measures [4], [26]. The network intrusion detection tasks have simply become more complex, in recent years, as new network attacks continue to emerge.

Anomaly-based IDS using machine learning has demonstrated potential of learning normal and anomalous network traffic and using the knowledge to detect attacks in real time [3], [10], [19]. However, the anomaly-based Intrusion Detection System (IDS) has faced challenges due to the presence of redundant and numerous features in training datasets. This has resulted in the generation of false positives [9]. Additionally, the anomaly-based IDS is often computationally complex [13], further adding to its limitations.

One of the techniques to improve the accuracy of machine learning model is to select only the most relevant features from large number of features in a dataset. In addition to improving the detection accuracy, feature selection increases the training and testing speed and reduces the computational overhead [6].

Reducing the computation overhead is also crucial as some IDS are meant to be deployed in real-time applications that require low latency. Hence, predictions or decisions need to be made within strict time constraints [18]. Computation overhead can delay the response time, making it challenging to meet real-time requirements. Similarly, in some real-time deployments, computing resources may be limited, such as edge devices, IoT devices, or embedded systems. These devices have constrained processing power, memory, and energy resources, making it necessary to optimize the model and reduce computation overhead.

Numerous studies have been undertaken to extract a smaller yet more relevant set of features from a large feature set. These studies have shown reasonable enhancements in the accuracy of trained models. However, many of these studies fail to strike a reasonable balance between improving accuracy, on one hand, and maintaining model simplicity and minimizing

time complexity, on the other hand

To this end, the goal of this paper is to improve accuracy, reduce computational overhead and with as simple model as possible. There are three (3) broad categories of feature selection: *wrapper, filter and embedded* [23]. However, the actual mechanisms that are popular for determining which feature to select are either (i) coefficients from a linear model or (ii) feature importance from a tree-based model.

We proposed a hybrid of filter-based and wrapper-based feature selection methods. In the filter-based, coefficients from a linear model was used to determine the features to be included. Thus. features with higher absolute coefficients are considered more important. On the other hand, the wrappe-based, feature importance from a tree-based model was used. The rest of this paper is structured as follows: In Section II, the review of related work in the field is presented. Section III provides a comprehensive discussion of the methodology used. Section IV presents the results obtained from the study, followed by a detailed analysis and discussion of these results in Section IV. Finally, in Section V, the conclusions of the paper as well future research was presented.

II. Related Works

To minimize false positive in IDS while minimizing computation overhead, researchers have advocated a multi-pronged approach. This approach involves the selection of multiple features and the training of multiple models to identify the optimal combination of features and the most effective performing model. For instance, [24] presented an approach of detecting and classifying Distributed Denial of Service (DDoS) attacks on network traffics. They employed all filter-based feature selection strategies such as chi-square and Analysis of Variance (ANOVA). They trained multiple machine learning models to select the best. The results of their approach achieved 99.9% accuracy for DDoS attack detection and over 85% accuracy for DDoS attack classification. The paper focused on DDOS, and the key approach used was utilizing a taxonomy-based divide and conquer strategy in which the the bigger problem of DDoS attack classification was broken- down into smaller binary classification problems. Similarly,
[20] conducted a comprehensive evaluation of various feature selection techniques, applying them to both Machine Learning (ML) and Deep Learning (DL) models. Their approach involved the development of an ensemble method that combined the outcomes of fifteen distinct feature selection strategies. They reported a notable reduction of features from 42 to just 15 which is about 64% reduction from the original feature set. This approach considered too many feature selection strategies in addition to the multiple model training. Nonetheless, the reported accuracy for their best-performing model was only 87.2%. Along the same line, [12] proposed an ensemble feature selection and classification model using XGBoost to improve the identification and classification of internet traffic in network management. They also constructed multiple models and combined them sequentially to form the final model. They did not consider hybrid feature selection. They reported 96% reduction in features and upto 98.51% accuracy.

In [14], a hybrid machine learning model with a feature selection method driven by coefficients from ch2 model was proposed for building Intrusion Detection System (IDS). They combined K means, as unsupervised machine learning model, with Light Gradient Boosting Machine (LGBM), as supervised machine learning, for the hybrid model. Rather than training multiple models, [25] used hybrid and two-phase feature selection method. In the first phase, they combined two filter-based methods: information gain (IG) and random forest (RF), to reduce the feature subset search space. In the second phase, a wrapper method that provides a recursive feature elimination (RFE) was used to further reduce features. They reported 36% reduction in features and improved accuracy from 82.25% to 84.24%.

To summarize, looking at the literature, studies mostly proposed multi-model approach and or traded simplicity, time and space complexities with slight improvement in accuracy. Recognizing that, [21] used wrapper-based feature selection and used the selected features to build neurotree to achieve a lightweight IDS within reasonable accuracy.

To simplify the feature selection process, [2] employed Principal Component Analysis (PCA). The PCA method has the advantage of transforming the original features into orthogonal components, which can be helpful in simplifying the relationships among features. This can lead to improved model performance, especially for algorithms that assume feature independence. However, they only investigated kernels for Support Vector Machine (SVM) in which they found r Gausian radial basis function superior to the likes of Sigmoid, Linear, and polynomial kernels.

III. Methodology

This section presents the detail of our approach to pursuing the goal of improving the accuracy of the machine learning model with the best feature combination and with less complicated model. Fig.1 presents the overview of the approach, and the next subsequent sections explain the activities involved begining with dataset used for the study.

A. The dataset

We used the publicly accessible network intrusion detection dataset provided by the Canadian Institute of Cybersecurity [22]. The dataset serves as a benchmark for comparing different intrusion detection methods. In particular, we used four different datasets from CIC-IDS2017 and each with different incidences such as DDoS, Port Scan, Bot, SQL Injection, and Cross Site Scripting (XSS). Table 1 present the summary of the datasets used in this paper.

B. Data Preprocessing

1) cleaning: At the Data Pre-processing stage, the data cleaning activity involved removing rows with values that were not valid numbers; Missing values were replaced with the mean value the feature. Similarly, infinite values were replaced

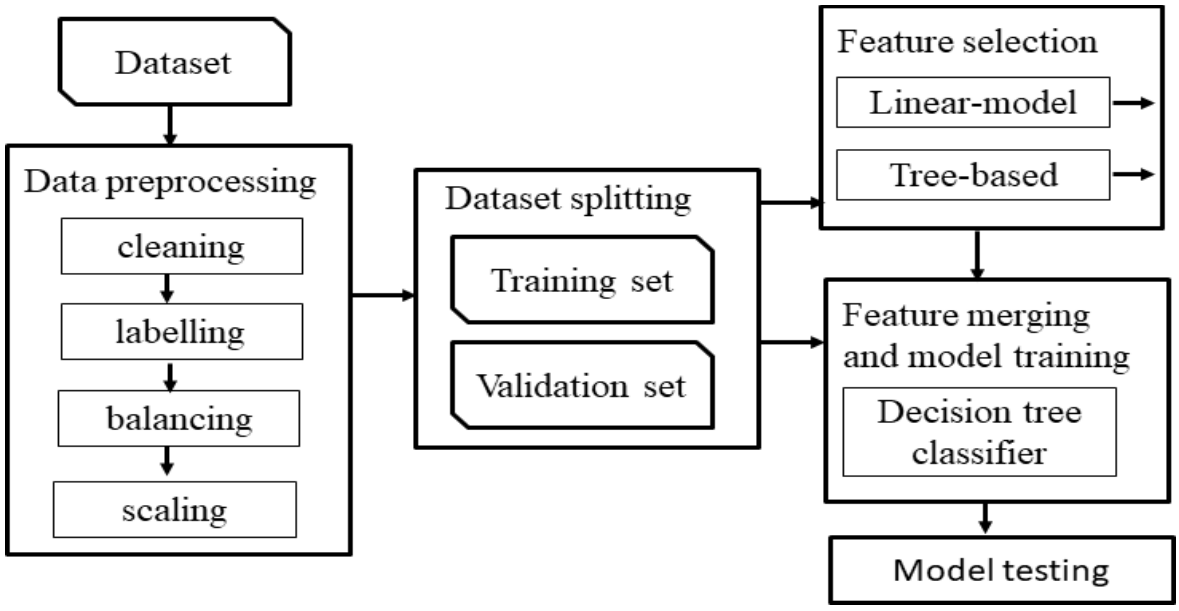

Fig. 1. Overview of the process to detect anomalies in network traffic using a hybrid of linear-based and tree-based feature selection approaches

Dataset name	**#of records**	**#of features**	**incidences**
Friday-WorkingHours-Afternoon-DDos.pcap ISCX (DDoS)	225,745	78	DDoS
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX (Portscan)	286,467	78	PortScan
Friday-WorkingHours-Morning.pcap_ISCX (Bot)	191,033	78	Bot
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX (Web attack)	170,366	78	Brute Force, SQL Injection, Cross Site Scripting

TABLE I
CIC-IDS2017 DATASETS USED FOR THE STUDY

with maximum finite values that can be represented by a 64-bit floating-point number.

2) labelling: For the labelling, in all the datasets, except *Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX dataset*, the target variable was coded as 0 and 1, with the former representing absence of the attack and the later representing the presence of the attack. In the *Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX* dataset, the values for the target variable were coded as 0, 1, 2,3 to represent normal, Brute Force, SQL Injection, Cross Site Scripting respectively.

3) balancing: In some cases, the datasets were balanced using Synthetic Minority Over-sampling Technique (SMOTE) [8] to address the problem of having significantly fewer instances of the minority classes compared to majority class.

Fig.2 and Fig.3 is the bar chart representation of *Friday-WorkingHours-Afternoon-DDos.pcap_ISCX* before and after balancing, respectively; Fig.4 and Fig.5 is the bar chart representation of *Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX* before and after balancing, respectively. Fig.6 and Fig.7 is the bar chart representation of *Friday-WorkingHours-Morning.pcap_ISCX* before and after balancing, respectively; Fig.8 and Fig.9 is the bar chart representation of *Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX* before and after balancing, respectively; In Fig.2 through Fig.7, the horizontal axis shows the target variable *Label* with a value of 0 indicating no attack and a value of 1 indicating attack. Similarly, in Fig.8. and Fig.9, the horizontal axis shows the target variable *Label* with a value of 0 indicating no attack and a value of 1, 2, and 3 indicating Brute Force, SQL Injection, and Cross Site Scripting attacks, respectively

4) scaling: The feature scaling was used to transform the features of the dataset to a consistent scale to ensure that no single feature dominates the learning algorithm due to differences in their magnitudes. The specific scaling technique used was standardization which transformed the features to have zero mean and unit variance. Feature scaling helps improve the performance and convergence of machine learning algorithms by ensuring that all features contribute equally to the learning process.

5) Dataset splitting: After the pre-processing, the dataset was split into separate subsets for training and evaluation purposes. The training set was used for both the feature selection and for training the machine learning model, while the validation set was also used to evaluate the performances of both models. This splitting was important to assess how well the models generalized to unseen data and to avoid overfitting, where the model performs well on the training data but poorly on new, unseen data

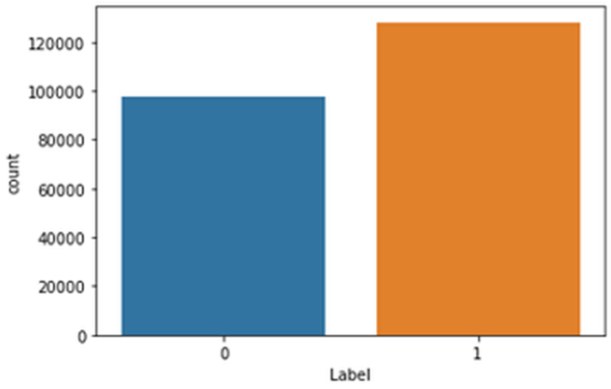

Fig. 2. Friday-WorkingHours-Afternoon-DDos.pcap_ISCX before balancing

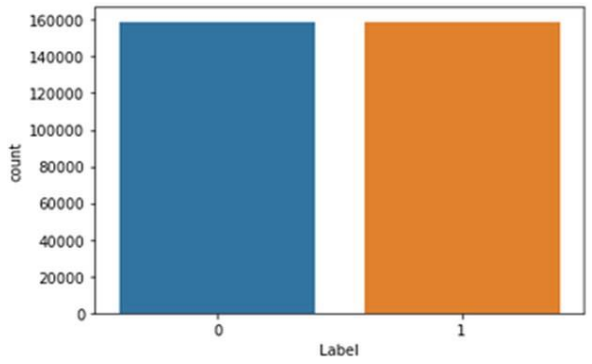

Fig. 3. Friday-WorkingHours-Afternoon-DDos.pcap_ISCX after balancing

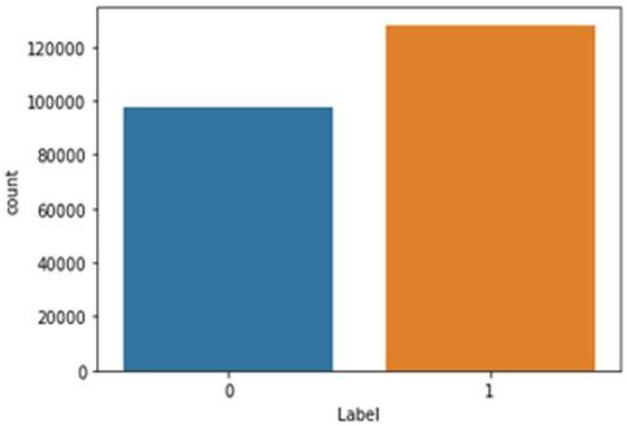

Fig. 4. Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX before balanc-ing

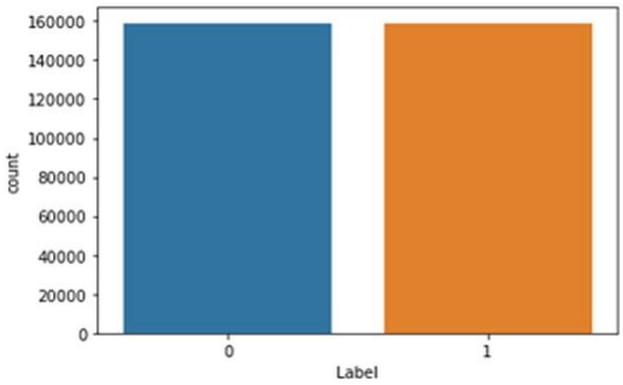

Fig. 5. Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX after balancing

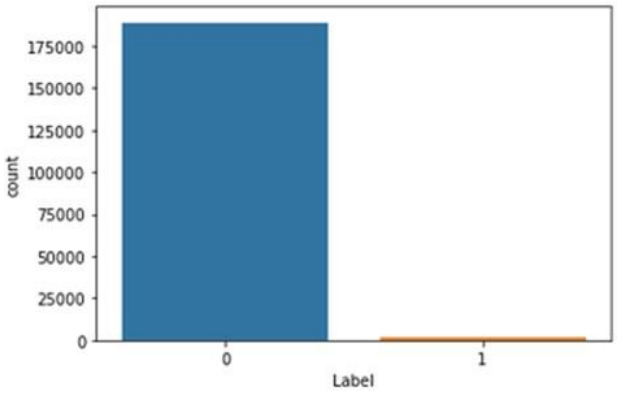

Fig. 6. Friday-WorkingHours-Morning.pcap_ISCX before balancing

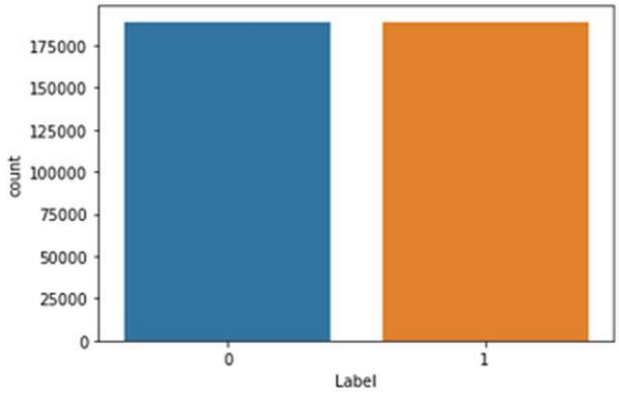

Fig. 7. Friday-WorkingHours-Morning.pcap_ISCX after balancing

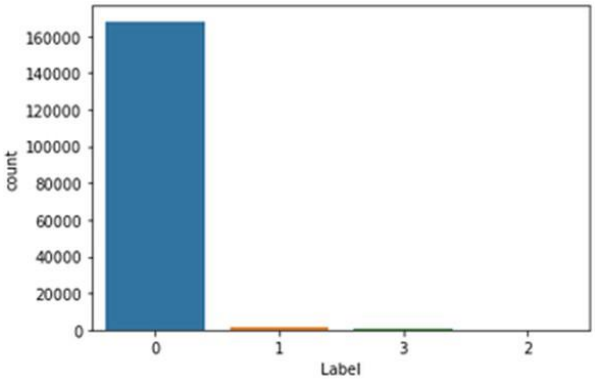

Fig. 8. Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX before balancing

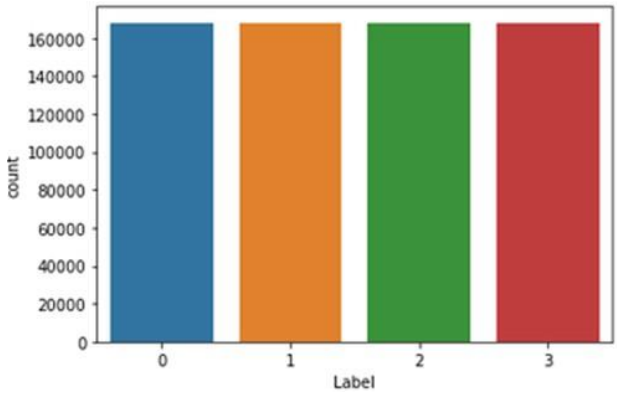

Fig. 9. Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX after balancing

C. Feature Selection

In this paper, we considered two (2) popular approaches of feature selection: i) coefficients from chi2 as the chosen linear model; and ii) feature importance as used in the Decision Tree [15]. For the chi2, we employed it as a scoring function in `SelectKBest` [5]. For the tree-based, we choose Recursive Feature Elimination (RFE) [7], as it has the advantage of considering feature interactions and dependencies during the elimination process. RFE is a flexible technique that can work with various machine learning algorithms and metrics.

1) Feature selection using SelectBest and chi2: The SelectKBest algorithm is a filter-based feature selection algorithm in machine learning. It selects the k features with the highest scores, according to a scoring function. The chi2 is the default scoring function that is used with `SelectKBest` for classification problems. It measures the independence between a feature and the target variable. The scoring function is a measure of how well each feature is able to predict the target variable. It is a popular choice for feature selection because it is simple to implement and interpret. It is also relatively efficient, making it a good choice for large datasets.

The process of feature selection using `SelectBest` can be explained with Algorithm 1.

Algorithm 1 Feature Selection using Chi-square

Require:
Dataset D with n instances and m features
Target variable Y with k classes
Scoring function χ^2
Number of features to select p
Ensure:
Selected features S of size p
procedure FEATURESELECTION(D, Y, χ^2, p)
  Calculate the chi-square statistic for each feature:
  for $i = 1$ to m **do**
    for $c = 1$ to k **do**
      Calculate O_{ic} and E_{ic} using the dataset D and class c
    end for
  end for
  Rank features based on their chi-square scores in descending order
  Select the top p features from the ranked list
  return selected features S
end procedure

The chi2 statistics for feature f_i and class c is calculated as follows:

$$\chi^2(f_i, c) = \frac{(O_{ic} - E_{ic})^2}{E_{ic}}$$

2) Feature selection using RFE: The process starts by training a model on the entire set of features. Once the initial model is trained, it assigns importance scores to each feature based on their contribution to the model's performance. At the elimination stage, the least important feature(s) are then removed from the dataset. This removal could be based on a predetermined number of features to eliminate or a certain threshold of importance scores. After eliminating features, a new model is trained using the remaining features. This new model might have a slightly different performance than the initial model. The elimination and training are repeated iteratively. In each iteration, the least important features are removed, and a new model is trained with the remaining features. The iterations continue until a stopping criterion is met. The final selected features are those that have been retained through the iterations. The final model is then trained using only these selected features.

The process of feature selection using RFE can be codified with the following Algorithm 2. Table II explains presents the explanation of the variables used in Algorithm 2.

Algorithm 2 Feature Selection using Recursive Feature Elimination Algorithm

1) $X_0 = X$
2) M_0 = TrainInitialModel(X_0, y)
3) I = CalculateFeatureImportance(M_0)
4) X_1 = EliminateLeastImportantFeatures(X_0, I)
5) M_1 = TrainModel(X_1, y)
6) **Repeat:**
   a) X_{i+1} = EliminateLeastImportantFeatures(X_i, I)
   b) M_{i+1} = TrainModel(X_{i+1}, y)
7) **Until:** $X = k$
8) SelectedFeatures = FeaturesFrom(X_{final})

TABLE II
DESCRIPTION OF VARIABLES USED IN ALGORITHM 1

Variable	Description
X_0	Initial feature matrix (a matrix where each row represents a sample and each column represent a feature)
M_0	Initial Model
M_i	Model at the ith iteration
X_i	Feature matrix at the ith iteration
y	Target variable (a vector containing the labels or values you're trying to predict)
k	Number of features to select or retain

D. Features merging and model training

We combined the feature sets selected by both the Recursive Feature Elimination (RFE) using a tree-based method and the Chi-squared (chi2) method in SelectKBest. These merged features were used to train and test a Decision Tree model. Leveraging this hybridized feature selection approach, we harnessed the strengths of both models: the Chi-squared model's ability to assign weights to features and the tree-based model's capability to determine feature importance.

IV. RESULTS

We conducted a comparison between two different feature selectors: the Chi-squared (chi2) method in SelectKBest and the feature importance of the decision tree as used with RFE. The results, presented in Fig.10, showed that the accuracy exceeded 99% in all cases. When using SelectBest with chi2 as the scoring function, the lowest accuracy (99.32%) was observed on the Web Attack dataset, while the highest accuracy (99.98%) was observed on the PortScan dataset. Similarly, for RFE with feature importance of the decision tree, the lowest accuracy (99.33%) was observed on the Web Attack dataset, and the highest accuracy (99.98%) was observed on the PortScan dataset. Despite selecting only a few common features, the accuracy from both selectors was comparable in all cases.

To evaluate the overall performance of our approach, we used the work in [17] as the benchmark. Table III presents the comparison of our approach with the benchmark.

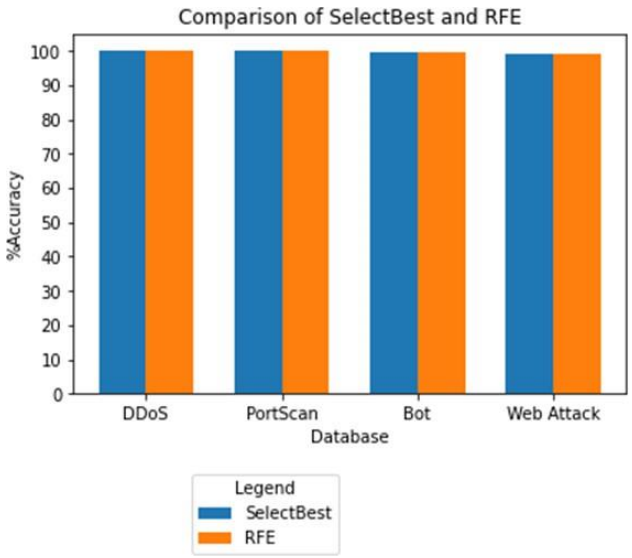

Fig. 10. comparison between RFE (with feature importance of decision tree) and SelectBest (with chi2 as scoring function) feature selectors

V. DISCUSSION

In each of our experimental scenarios, our proposed approach consistently achieves accuracy levels surpassing 99%, thereby becoming comparable or even outperforming conventional feature selection methods. Comparing with the existing studies in the literature, this work has demonstrated the efficiency of our approach. For instance, the study by [24] that employed various filter-based feature selection strategies, including chi-square and Analysis of Variance (ANOVA), and also trained multiple machine learning models recorded only 85% accuracy for DDoS attack classification.

Similarly, [20] with their ensemble method that combined the outcomes of fifteen distinct feature selection strategies in which 64% in feature reduction was achieved, their best-performing model only achieved an accuracy of only 87.2%. The literature predominantly leans toward proposing multi-model approaches or trading simplicity, time efficiency, and space complexity for marginal improvements in accuracy.

VI. CONCLUSION

In this study, we have proposed and evaluated a hybrid feature selection strategy designed to enhance the accuracy and efficiency of Intrusion Detection Systems (IDS). Our comprehensive experimental results reveal the remarkable efficacy of this approach, with accuracy levels consistently surpassing the 99% threshold across all experimental scenarios. The pivotal achievement of our hybrid approach lies in its ability to significantly reduce the number of features required for detection, culminating in improved computational efficiency and reduced storage demands.

Looking ahead, further refinement and optimization of our hybrid approach present exciting avenues for future research. Additionally, the application of our strategy to real-world network environments and the evaluation of its performance

TABLE III
COMPARISON OF APPROACHES ON DIFFERENT DATASETS

Dataset	Approaches	Accucary (%)	Precision (%)	Recall (%)	Feature Reduced	% Reduction	% Improvement in Accuracy
DDos	benchmark	99.83255094	99.8325509	99.82900706	NA	NA	NA
	Our Approach	99.92345365	99.92345365	99.92345365	59	75.64	0.090
Port Scan	benchmark	99.98394159	99.98359252	99.9825452	NA	NA	NA
	Our Approach	99.98238218	99.98238218	99.98238218	60	76.92	-0.0016
BOT	benchmark	99.91047268	99.90314298	99.90366654	NA	NA	NA
	Our Approach	99.42051897	99.42051897	99.42051897	60	76.92	-0.49
Web attack	benchmark	96.00766505	96.00766505	96.00766505	61	NA	NA
	Our Approach	99.41361539	99.41361539	99.41361539	61	78.21	3.41

under evolving threat landscapes remain promising areas of exploration.

REFERENCES

[1] Albahar, M. (2019). Cyber attacks and terrorism: A twenty-first century conundrum. Science and engineering ethics, 25, 993-1006.

[2] Almaiah, M. A., Almomani, O., Alsaaidah, A., Al-Otaibi, S., Bani-Hani, N., Hwaitat, A. K. A., ... & Aldhyani, T. H. (2022). Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels. Electronics, 11(21), 3571.

[3] Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge-Based Systems, 189, 105124.

[4] Appelt, D., Nguyen, C. D.,& Briand, L. (2015, April). Behind an application firewall, are we safe from SQL injection attacks?. In 2015 IEEE 8th international conference on software testing, verification and validation (ICST) (pp. 1-10). IEEE.

[5] Ayyanar, M., Jeganathan, S., Parthasarathy, S., Jayaraman, V., & Lakshminarayanan, A. R. (2022, April). Predicting the Cardiac Diseases using SelectKBest Method Equipped Light Gradient Boosting Machine. In 2022 6th International conference on trends in electronics and informatics (ICOEI) (pp. 117-122). IEEE

[6] Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. Neurocomputing, 300, 70-79.

[7] Chen, X. W., & Jeong, J. C. (2007, December). Enhanced recursive feature elimination. In Sixth international conference on machine learning and applications (ICMLA 2007) (pp. 429-435). IEEE.

[8] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[9] Dong, G., & Liu, H. (Eds.). (2018). Feature engineering for machine learning and data analytics. CRC press.

[10] Drewek-Ossowicka, A., Pietrołaj, M., & Rumin´ski, J. (2021). A survey of neural networks usage for intrusion detection systems. Journal of Ambient Intelligence and Humanized Computing, 12, 497-514.

[11] Kaur, T., Malhotra, V., & Singh, D. (2014). Comparison of network security tools-firewall, intrusion detection system and Honeypot. Int. J. Enhanced Res. Sci. Technol. Eng, 200204.

[12] Manju, N., Harish, B. S., & Prajwal, V. (2019). Ensemble feature selection and classification of internet traffic using XGBoost classifier. International Journal of Computer Network and Information Security, 11(7), 37.

[13] Majeed, A. (2019). Improving time complexity and accuracy of the machine learning algorithms through selection of highly weighted top k features from complex datasets. Annals of Data Science, 6, 599-621.

[14] Mazumder, A. M. R., Kamruzzaman, N. M., Akter, N., Arbe, N., & Rahman, M. M. (2021, February). Network intrusion detection using hybrid machine learning model. In 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT) (pp. 1-8). IEEE.

[15] Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society, 18(6), 275-285.

[16] Peng, T., Leckie, C., & Ramamohanarao, K. (2003, May). Protection from distributed denial of service attacks using history-based IP filtering. In IEEE International Conference on Communications, 2003. ICC'03. (Vol. 1, pp. 482-486). IEEE.

[17] Rahman, M. A., Al-Saggaf, Y., & Zia, T. (2020, November). A data mining framework to predict cyber attack for cyber security. In 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA) (pp. 207-212). IEEE.

[18] Raponi, S., Caprolu, M., & Di Pietro, R. (2019). Intrusion detection at the network edge: Solutions, limitations, and future directions. In Edge Computing–EDGE 2019: Third International Conference, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, 2019, Proceedings 3 (pp. 59-75). Springer International Publishing.

[19] Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. Computers and Electrical Engineering, 99, 107810.

[20] Saha, S., Priyoti, A. T., Sharma, A., & Haque, A. (2022, January). Towards an optimal feature selection method for AI-based DDoS detection system. In 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC) (pp. 425-428). IEEE.

[21] Sindhu, S. S. S., Geetha, S., & Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. Expert Systems with applications, 39(1), 129-141.

[22] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp, 1, 108-116.

[23] Stan´czyk, U. (2015). Feature evaluation by filter, wrapper, and embedded approaches. Feature Selection for Data and Pattern Recognition, 29-44.

[24] Thorat, O., Parekh, N., & Mangrulkar, R. (2021). TaxoDaCML: Taxonomy based Divide and Conquer using machine learning approach for DDoS attack classification. International Journal of Information Management Data Insights, 1(2), 100048.

[25] Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., & Kwak, J. (2023). IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. Journal of Big Data, 10(1), 1-26.

[26] Zhang, F., Kodituwakku, H. A. D. E., Hines, J. W., & Coble, J. (2019). Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. IEEE Transactions on Industrial Informatics, 15(7), 4362-4369.