A Comparative Study of Public Network Intrusion Detection Hybrid Machine Learning Approaches

N. S. Vasantha
Department of Electronics and Communication Engineering, School of Engineering and Technology, Jain University, Bangalore, Karnataka, India
vasanthareddyns@gmail.com (corresponding author)

R. Sukumar
Centre of Research for Cyber Security (CRCE), School of Engineering and Technology, Jain University, Bangalore, Karnataka, India
r.sukumar@jainuniversity.ac.in

Shridhar Allagi
Department of Computer Science and Engineering, KLE Institute of Technology, Hubballi, Karnataka, India
shridharallagi@kleit.ac.in

Tessy Tom
Centre of Research for Cyber Security (CRCE), School of Engineering and Technology, Jain University, Bangalore, Karnataka, India
tessy.tom@jainuniversity.ac.in

Received: 15 January 2026 | Revised: 11 April 2026 and 22 April 2026 | Accepted: 23 April 2026

 DOI: https://doi.org/10.48084/etasr.17560

ABSTRACT

Security in public networks has always been a challenge. With an ever-expanding landscape of cyberattacks, it is imperative to explore new/alternate mechanisms of implementing intrusion detection. Machine learning and deep learning techniques have emerged as promising for intrusion detection in the recent past, with increased efficiency. This study explores machine learning algorithms, namely Random Forest, Naive Bayes, and Decision Tree, for the detection of web-based attacks in public networks, using a combination of Principal Component Analysis and Haar Wavelet Transform for feature extraction. The results of these models are compared, and related issues and approaches to alleviate them are explored.

Keywords-Principal Component Analysis (PCA); Random Forest (RF); Machine Learning (ML); Intrusion Detection System (IDS)

I. INTRODUCTION

The scale of internet usage has grown rapidly over the last decade, driven by the increasing number of connected devices and digital services. As a result, modern networks generate massive volumes of traffic, which makes monitoring and securing them more difficult than before. This shift has also expanded the opportunities available to attackers, exposing weaknesses in traditional security mechanisms.

Conventional Intrusion Detection Systems (IDS), especially those based on predefined signatures, struggle to keep up with new and evolving threats. They are effective when the attack pattern is already known, but tend to fail when faced with previously unseen or modified attacks. Due to this limitation, there has been a growing interest in using machine learning techniques, which can learn patterns directly from data and adapt over time without requiring constant manual updates.

This work investigates the combined use of Principal Component Analysis (PCA) and Haar Wavelet Transform (HWT) for dimensionality reduction and feature quality enhancement, followed by classification using Random Forest (RF), Decision Tree (DT), and Naive Bayes (NB) algorithms. While each of these components has individually been studied in prior literature, their specific integration in the context of

web attack detection within the CICIDS2018 dataset [1, 2] with a systematic comparative evaluation represents the targeted contribution of this study. The motivation for employing HWT alongside PCA stems from the multi-resolution decomposition capability of wavelets, which can isolate both coarse and fine-grained structural variations in traffic feature vectors that PCA alone may not fully capture.

The study in [3] aimed to enhance IDS performance through advanced feature selection and comparative analysis of different ML and DL techniques using recent datasets. This study addressed the increasing need for effective IDSs due to the growing volume of data and associated security threats in modern networks. A hybrid feature selection technique combined the Pearson correlation coefficient and RF to improve the efficiency of the IDS. The DT model provided optimal accuracy among the ML models examined, while the MLP model performed best among DL models. Various parameters, such as accuracy, precision, and recall, were considered to evaluate model effectiveness.

In [4], ML algorithms for IDSs were explored, focusing on five widely-used IDS datasets: CSE-CIC IDS-2018, UNSW-NB15, ISCX-2012, NSL-KDD, and CIDDS-001. These datasets were normalized and classified using SVM, KNN, and DT algorithms. The study ran 100 iterations of SVM, DT, and KNN classifiers, measuring accuracy, precision, geometric mean, and F-measure. In [5], the need for advanced techniques in malware detection was highlighted. In this study, data were preprocessed to remove missing values. The extraction stage selected the 13 dominant features out of the total 56 features. Five algorithms were compared for classification, with RF showing the highest accuracy.

In [6], growing challenges of cybersecurity in the Industrial Internet of Things (IIoT) were discussed, specifically focusing on malware detection and software piracy. This study introduced a deep learning approach with a CNN to identify and mitigate these threats efficiently. In this work, DL algorithms far outperformed their ML counterparts, achieving high classification accuracy and F-measure scores, indicating effective detection of both malware and software piracy with reduced computational costs. In [7], a machine learning algorithm design was introduced to improve malware detection through advanced data analysis and feature extraction. This model aimed to address the challenges of increased volumes of data and the subsequent increase in malware attacks, achieving better performance in the detection of malware compared to traditional signature-based methods, and improving cybersecurity measures in IoT environments.

In [8], a deep learning-based system was developed for web attack detection on edge devices. Traditional firewalls and IDSs are not well-equipped to handle attacks in contemporary network environments. Cloud-based systems centralize the data and, hence, attract more attacks. Distributed deep learning models can analyze URLs to detect web attacks in edge devices. The accuracy of the system for attack detection was 99.41%, but this study had the limitation of not being tested against different algorithms and attack styles.

In [9], the detection of Android malware using deep learning techniques was reviewed. This study used two datasets, CIC-AndMal2017 for dynamic analysis and CIC-InvesAndMal2019 for static analysis. Various deep learning models, including LSTM and CNN-LSTM, were evaluated. The LSTM model achieved 0.988 accuracy in static analysis, while the CNN-LSTM model achieved 0.953 accuracy in dynamic analysis, both outperforming existing methods in the literature.

II. PROPOSED MODEL

Figure 1 gives an overview of the proposed model. The model pipeline consists of a data ingestion stage, a feature preprocessing stage, and a final classification stage that categorizes each network flow as either benign or malicious.

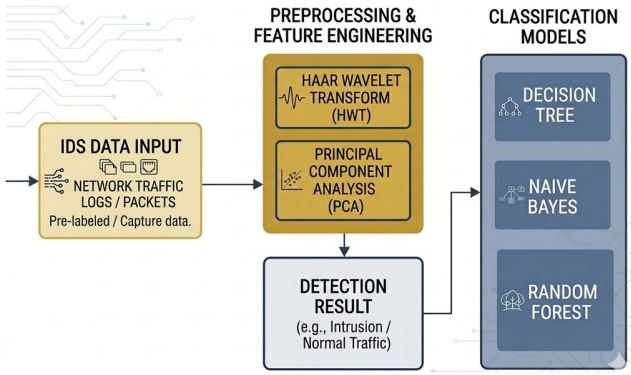

Fig. 1. Proposed architecture.

Feature preprocessing plays a key role in how well a model performs, especially when working with high-dimensional network data, as not all extracted features contribute equally to classification, and some may introduce noise or redundancy. Principal Component Analysis (PCA) is commonly used to reduce dimensionality by transforming the original feature space into a smaller set of uncorrelated components. Instead of working with all variables, the model focuses only on those that capture most of the variation in the data. This makes the dataset easier to handle and often improves computational efficiency. The Haar Wavelet Transform (HWT), on the other hand, operates differently. It breaks down the input signal into components that represent both coarse trends and finer variations. In the context of network traffic features, this helps in separating useful patterns from noise. When used together, PCA and HWT tend to complement each other. PCA reduces redundancy across features, while HWT helps refine the transformed data by filtering out high-frequency noise. The resulting feature set is smaller, cleaner, and more suitable for classification.

Feature importance analysis, illustrated in Figure 2, and the inter-feature correlation matrix in Figure 3, guided the selection of preprocessing parameters and confirmed the presence of substantial redundancy that PCA and HWT together address.

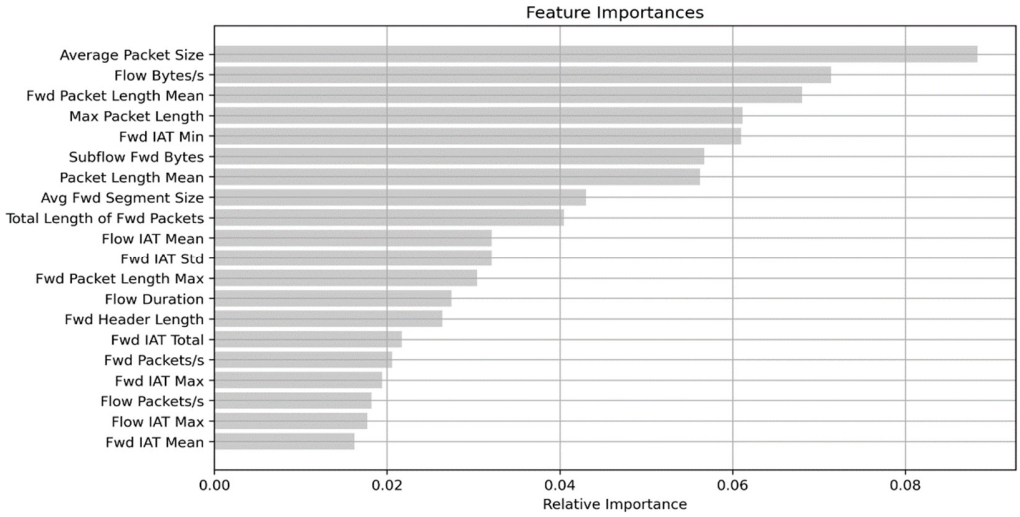

Fig. 2. Relative importance of each feature.

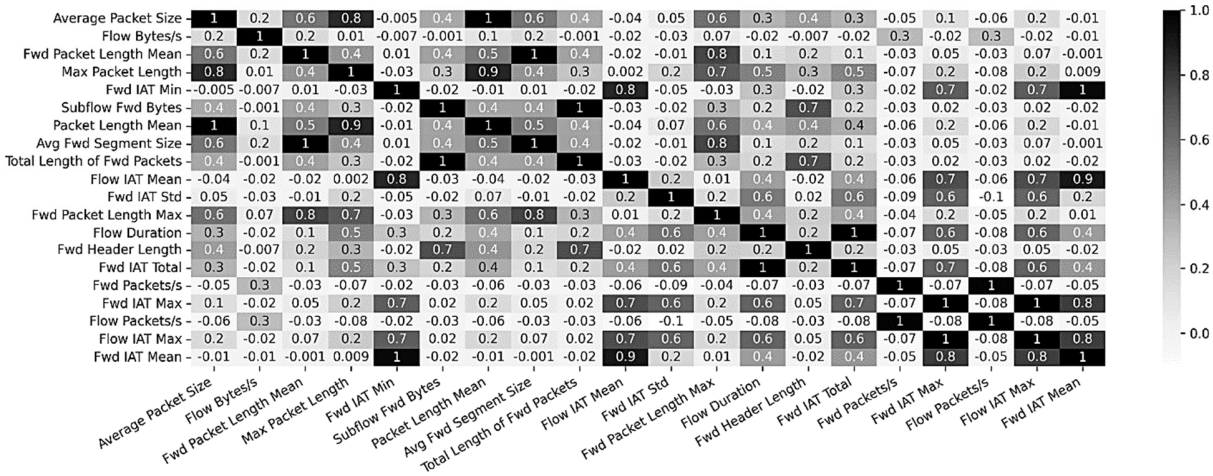

Fig. 3. Correlation between each feature.

A. Dataset Description

The CICIDS2018 dataset [1, 2] is a widely used benchmark for network intrusion detection research. It was generated by the Canadian Institute for Cybersecurity and captures realistic network traffic comprising both benign activity and multiple attack categories, including brute force, web attacks, DoS, DDoS, and infiltration. The dataset provides over 80 flow-level features derived from raw packet captures, covering attributes such as packet length statistics, inter-arrival times, flow duration, flag counts, and protocol information.

For this study, the web attack subset of CICIDS2018 was selected. The dataset is structured as a four-class classification problem: Class 0 represents benign (normal) traffic, Class 1 corresponds to Brute Force-Web attacks, Class 2 corresponds to Brute Force-XSS (Cross-Site Scripting) attacks, and Class 3 corresponds to SQL Injection attacks. Although these span four distinct class labels, they all belong to the same overarching attack family of web-based intrusion techniques. This approach can be characterized as a single attack category evaluation.

Web attacks present distinctive statistical signatures in flow-level features that make them a suitable and practically relevant target for assessing feature extraction efficacy. In addition, Web attacks are among the most prevalent threat vectors in contemporary public networks, and their shared reliance on application-layer exploitation makes them a well-defined and practically significant target for evaluating the proposed feature extraction pipeline. Future work will extend the evaluation to additional attack categories within the dataset.

B. Feature Extraction and Preprocessing

Data preprocessing began with the removal of records containing infinity values, NaN entries, and blank fields, which are commonly introduced during traffic capture. After cleaning, the features were normalized using z-score to ensure that no single feature dominated the subsequent dimensionality reduction steps due to scale differences. PCA was applied first to reduce the high-dimensional feature space. The number of principal components was determined by retaining components that cumulatively explain at least 95% of the total variance in

the training data, which yielded approximately 30 components from the original 80 features. PCA achieves this by computing the covariance matrix of the normalized features, extracting its eigenvectors, and projecting the data onto the subspace spanned by the top-ranked eigenvectors.

The Haar Wavelet Transform (HWT) was subsequently applied to the PCA-reduced feature vectors to further enhance signal quality. The Haar wavelet is the simplest orthonormal wavelet and decomposes a discrete signal x of length N into approximation coefficients c_A (capturing low-frequency trends) and detail coefficients c_D (capturing high-frequency variations), defined respectively as:

$$c_A[k] = x[2k] + x[2k+1])/\sqrt{2} \text{ and}$$

$$c_D[k] = (x[2k] - x[2k+1]) / \sqrt{2}.$$

Applied to the PCA components, HWT captures multi-resolution structural information within the feature vectors that PCA's purely variance-based decomposition may leave latent. The approximation coefficients from a single-level decomposition were retained as the final feature representation, yielding a final dimensionality of approximately 21 features used for classifier training. This two-stage approach is motivated by the complementary nature of the two transforms: PCA removes inter-feature redundancy through variance-based projection, while HWT suppresses within-feature noise through frequency-domain decomposition.

C. Classification

Three classifiers were evaluated on the processed feature set: DT, NB, and RF. The dataset was partitioned into 80% for training and 20% for testing using a stratified split to preserve the class distribution across both subsets. Each classifier was trained and evaluated independently on the same partition.

RF is an ensemble method that constructs a large number of DTs during training, each built on a bootstrap sample of the training data with a randomly selected subset of features at each split. The final prediction is determined by majority voting across all trees. Here, RF was configured with 100 estimators, using the Gini impurity criterion for split selection, with no maximum depth constraint to allow full tree growth, and a minimum of two samples required to split an internal node. The random subspace method (*max_features* = *sqrt*) was applied at each split to reduce inter-tree correlation and improve ensemble diversity. This combination of bagging and feature randomization makes RF highly resistant to overfitting and well-suited to high-dimensional, noisy network traffic data.

The DT classifier recursively partitions the feature space by selecting the attribute and threshold that maximise information gain (measured by Gini impurity) at each node. This work used the CART (Classification and Regression Tree) algorithm. The tree was allowed to grow without a fixed maximum depth, with splits terminated when nodes contained fewer than two samples or when no further impurity reduction was achievable. DTs are computationally efficient and produce interpretable models; however, their tendency to overfit on training data, particularly when trees are deep and the data contains noise, limits generalization performance compared to ensemble approaches.

NB is a probabilistic classifier based on Bayes' theorem, with the conditional independence assumption that all features are mutually independent given the class label. This study employed the Gaussian variant of NB, which models each feature's class-conditional distribution as a Gaussian with parameters estimated from training data. Although this assumption rarely holds exactly in practice, particularly for correlated network flow features, NB offers computational efficiency and reasonable performance as a baseline. Its relatively lower accuracy in this study is consistent with prior findings on structured network traffic data, where inter-feature correlations are non-trivial, and the independence assumption introduces systematic bias in posterior probability estimates.

The classification results are reported across four classes. Class 0 represents benign (normal) network traffic. Class 1 corresponds to Brute Force-Web attacks, Class 2 corresponds to Brute Force-XSS (Cross-Site Scripting) attacks, and Class 3 corresponds to SQL Injection attacks. All three attack classes fall within the same overarching category of web-based intrusions. The severe class imbalance in the dataset—with 312,683 benign instances against only 113, 45, and 20 instances for Classes 1, 2, and 3, respectively—is an important factor that influences recall for the minority attack classes across all evaluated classifiers.

D. Decision Tree (DT) vs. Naive Bayes (NB)

Figures 4 and 5 show the comparative results between DT and NB. While the DT model achieves a significantly higher accuracy (≈99.97%) compared to NB (≈63.45%), a closer inspection of the classification metrics reveals important differences. DT demonstrates balanced performance across classes, with macro-averaged precision, recall, and F1-score values of approximately 0.88, 0.84, and 0.86, respectively. In contrast, NB struggles with minority class prediction, exhibiting extremely low precision and F1-scores despite moderate recall. This indicates that NB produces a large number of false positives and is less reliable in distinguishing between attack classes.

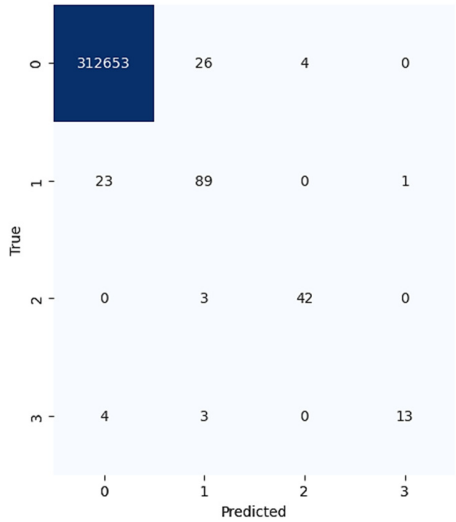

Fig. 4. Confusion matrix for DT.

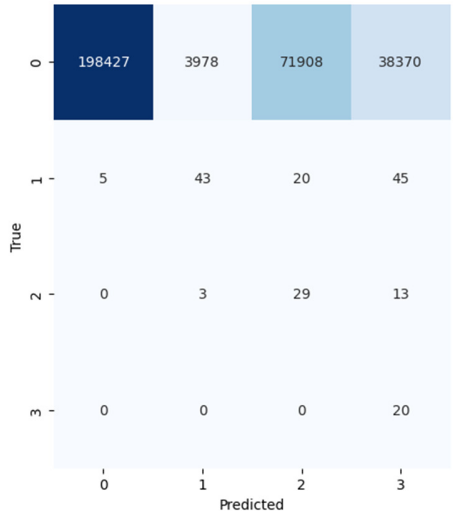

Fig. 5. Confusion matrix for NB.

TABLE I. CLASSIFICATION REPORT FOR DT

	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	312683
1	0.74	0.79	0.76	113
2	0.91	0.93	0.92	45
3	0.93	0.65	0.76	20

TABLE II. CLASSIFICATION REPORT FOR NB

	Precision	Recall	F1-score	Support
0	1.00	0.63	0.78	312683
1	0.01	0.38	0.02	113
2	0.00	0.64	0.00	45
3	0.00	1.00	0.00	20

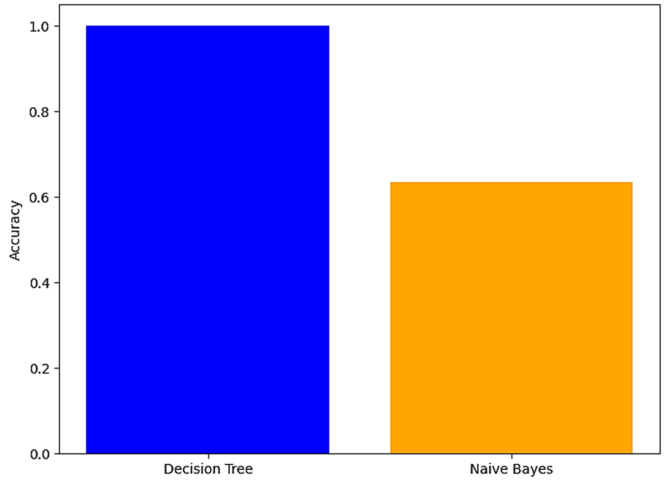

Fig. 6. Accuracy comparison between DT and NB.

E. Random Forest (RF) vs. Naive Bayes (NB)

As illustrated in Figures 7 and 8, the RF model significantly outperformed NB, achieving an overall accuracy of approximately 99.98%, with strong class-wise performance. In particular, the macro-averaged precision, recall, and F1-score for RF were 0.93, 0.79, and 0.85, respectively. Class-wise analysis shows near-perfect performance for the majority class (precision and recall ≈1.00), while minority classes maintain high precision (up to 0.98) but comparatively lower recall (as low as 0.55). This suggests that while the model is highly precise in identifying attacks, some minority instances remain undetected. This reduced recall for minority attack classes is directly attributable to the severe class imbalance in the dataset: with only 113, 45, and 20 instances for Classes 1, 2, and 3, respectively, against 312,683 benign instances, all classifiers are trained predominantly on benign examples and consequently show a bias toward the majority class. The low recall for attack classes represents a limitation of the current study; in a real-world IDS deployment, missed attack detections (false negatives) carry significant operational risk. Addressing this imbalance through oversampling techniques such as SMOTE (Synthetic Minority Over-sampling Technique) or ADASYN, or through cost-sensitive learning that penalises misclassification of minority classes more heavily, is identified as a priority direction for future work.

TABLE III. CLASSIFICATION REPORT FOR RF

	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	312683
1	0.90	0.69	0.78	113
2	0.98	0.93	0.95	45
3	0.85	0.55	0.67	20

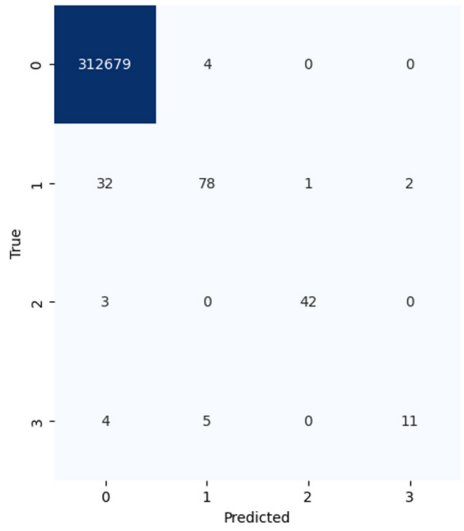

Fig. 7. Confusion matrix for RF.

F. Random Forest (RF) vs. Naive Bayes (NB) vs. Decision Tree (DT)

Figure 9 presents a combined comparison of all three models. Although both RF and DT achieve similarly high accuracy levels, RF provides a slightly better balance between precision and recall for minority classes. Although DT shows marginally higher recall for certain classes, RF maintains consistently higher precision, which is particularly important in intrusion detection scenarios to reduce false alarms. The NB classifier, in comparison, performs significantly worse across all evaluation metrics.

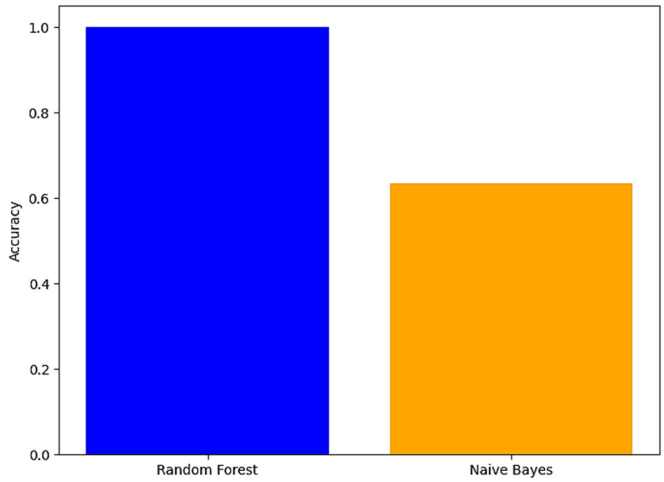

Fig. 8. Accuracy comparison between RF and NB.

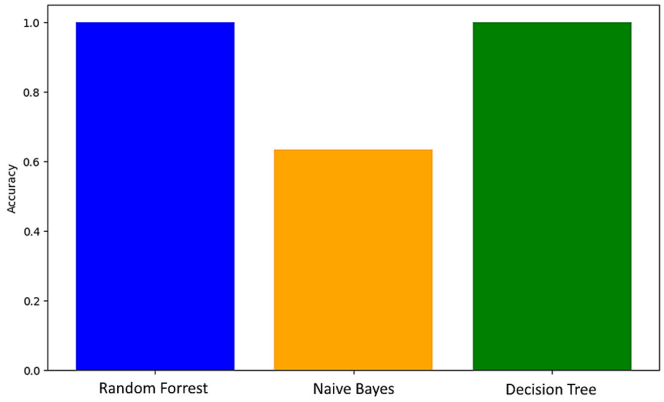

Fig. 9. Accuracy comparison for all models.

III. CONCLUSION

This paper examined the combined use of PCA and HWT for feature preprocessing, followed by classification using RF, DT, and Gaussian NB on the web attack subset of the CICIDS2018 dataset. The two-stage feature reduction pipeline reduced the original 80-feature space to approximately 21 informative components, with PCA addressing inter-feature redundancy through variance-based projection and HWT suppressing within-feature noise through single-level frequency decomposition. Among the evaluated classifiers, RF consistently achieved the highest accuracy and F1-score. This advantage is attributed to the ensemble mechanism inherent to RF; by aggregating predictions from 100 independently trained trees, each exposed to a random feature subset, the model effectively reduces variance without sacrificing bias, making it robust to the overlapping feature distributions that characterize web attack versus benign traffic. DT performed comparably in some configurations but showed greater susceptibility to noise in the feature space, while NB underperformed due to the strong conditional independence assumption being violated by correlated flow-level features.

However, several limitations must be acknowledged. The evaluation is confined to a single attack category within CICIDS2018, and the absence of cross-validation means that variance in the reported metrics across different data splits was not assessed. Additionally, no comparison with deep learning baselines, such as LSTM or CNN-based IDS models, was conducted, which limits the ability to contextualize the results relative to the broader literature.

Future work will extend the evaluation across all attack categories in CICIDS2018, apply k-fold cross-validation to obtain statistically robust performance estimates, employ oversampling techniques to address the class imbalance, and benchmark the proposed pipeline against recent deep learning-based intrusion detection approaches. The integration of attention mechanisms or hybrid CNN-wavelet architectures for automated feature learning from raw flow statistics also represents a promising direction for further investigation.

DECLARATION OF COMPETING INTERESTS

Not applicable to this work.

ACKNOWLEDGMENT

The authors acknowledge the help of the Electronics department, Jain University. This research received no external funding.

DATA AVAILABILITY

The dataset used in this study is publicly available at [2].

REFERENCES

[1] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116, https://doi.org/10.5220/0006639801080116.

[2] "IDS 2018." Canadian Institute for Cybersecurity, [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html.

[3] I. Hidayat, M. Z. Ali, and A. Arshad, "Machine Learning-Based Intrusion Detection System: An Experimental Comparison," *Journal of Computational and Cognitive Engineering*, vol. 2, no. 2, pp. 88–97, July 2022, https://doi.org/10.47852/bonviewJCCE2202270.

[4] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Computer Networks*, vol. 188, Apr. 2021, Art. no. 107840, https://doi.org/10.1016/j.comnet.2021.107840.

[5] S. Judy and R. Khilar, "Detection and Classification of Malware for Cyber Security using Machine Learning Algorithms," in *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Apr. 2023, pp. 1–6, https://doi.org/10.1109/ICONSTEM56934.2023.10142575.

[6] F. Ullah *et al.*, "Cyber Security Threats Detection in Internet of Things Using Deep Learning Approach," *IEEE Access*, vol. 7, pp. 124379–124389, 2019, https://doi.org/10.1109/ACCESS.2019.2937347.

[7] N. Z. Gorment, A. Selamat, L. K. Cheng, and O. Krejcar, "Machine Learning Algorithm for Malware Detection: Taxonomy, Current Challenges, and Future Directions," *IEEE Access*, vol. 11, pp. 141045–141089, 2023, https://doi.org/10.1109/ACCESS.2023.3256979.

[8] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A Distributed Deep Learning System for Web Attack Detection on Edge Devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, Mar. 2020, https://doi.org/10.1109/TII.2019.2938778.

[9] E. C. Bayazit, O. K. Sahingoz, and B. Dogan, "Deep Learning based Malware Detection for Android Systems: A Comparative Analysis," *Tehnicki vjesnik - Technical Gazette*, vol. 30, no. 3, June 2023, https://doi.org/10.17559/TV-20220907113227.