

Ensemble-Driven Malware Detection with Anomaly Override Capabilities

1st Viren Passi

Chandigarh College of Eng. and Tech.
Chandigarh, India
co23369@ccet.ac.in

2nd Sudhakar Kumar

Chandigarh College of Eng. and Tech.
Chandigarh, India
sudhakar@ccet.ac.in

3rd Sunil K. Singh

Chandigarh College of Eng. and Tech.
Chandigarh, India
sksingh@ccet.ac.in

4th Shivam Jindal

Chandigarh College of Eng. and Tech.
Chandigarh, India
co23360@ccet.ac.in

5th Prince Raj

Galgotias University
Greater Noida, Uttar Pradesh, India
princeraj15042005@gmail.com

6th Varsha Arya

Hong Kong Metropolitan University
Hong Kong SAR, China
varya@hkmu.edu.hk

7th Kwok Tai Chui

Hong Kong Metropolitan University
Hong Kong SAR, China
jktchui@hkmu.edu.hk

8th Brij B. Gupta

Dept. of Comp. Sci. and Info. Eng.
Asia University
Taichung, Taiwan
bbgupta@asia.edu.tw

Abstract—This paper proposes a hybrid malware detection framework that combines ensemble learning with an anomaly-based override mechanism. The ensemble component integrates Random Forest and XGBoost classifiers, while the override mechanism uses Isolation Forest to detect anomalous or uncertain samples. Each model was individually trained and evaluated on a labeled malware dataset, with Random Forest and XGBoost achieving 78.54% and 72.91% accuracy, respectively. Feature selection was performed using mutual information and ANOVA F-score to enhance model performance. In the proposed two-stage decision process, classification outputs from RF and XGBoost are fused through probabilistic averaging, and Isolation Forest acts as an override to flag potential malware. This hybrid strategy effectively reduces false negatives and enhances detection accuracy. The complete framework achieved 91.49% test accuracy and a weighted F1-score of 0.91, demonstrating its strength in detecting diverse malware types with improved reliability and adaptability.

Index Terms—Malware Detection, Ensemble Learning, Anomaly Override, Hybrid Classification, Feature Selection, Anomaly Detection

I. INTRODUCTION

Malware [1], short for malicious software, continues to pose a significant and evolving threat to cybersecurity [2]–[4] worldwide. It encompasses a broad range of software types, including viruses, worms, trojans, ransomware, and spyware designed to infiltrate, damage, or disrupt computer systems without user consent. With the rise of the Internet of Things (IoT), cloud computing, and mobile applications, the attack surface for malware has expanded dramatically [5]. Traditional malware detection systems, primarily based on signature-based methods, are increasingly inadequate in identifying newly emerging and obfuscated threats that can bypass static defenses.

To address this challenge, machine learning (ML) and artificial intelligence (AI) have emerged as promising solutions [6], [7]. These approaches allow for dynamic analysis and detection of previously unseen malware by learning patterns from vast datasets. However, most conventional ML approaches rely heavily on a single classifier or technique, which may not generalize well across diverse and complex malware families. Furthermore, challenges such as noisy features and class imbalance in malware datasets can significantly degrade model performance [8].

In this paper, a novel hybrid approach, dubbed RF-XGB-IF, which leverages the strengths of two state-of-the-art ensemble learning methods, namely Random Forest (RF) and XGBoost (XGB), as well as unsupervised anomaly detection with Isolation Forest (IF), is introduced. This combination seeks to improve detection precision and lower false negatives, which are very important in malware family categorization. In this study, we employ the Microsoft Malware Sample Dataset and apply feature selection approaches such as Mutual Information, ANOVA F-score for dimensionality reduction with the aim of keeping vital features. We performed hyperparameter tuning using a randomized search to maximize the performance.

The system makes use of both supervised and unsupervised methods and improves on the gains over single-model approaches. Our results demonstrate the viability of hybrid ensemble models to provide accurate malware detection systems that can adjust to real-world security threats. The major contributions of this study are as follows:

- 1) Propose a hybrid malware detection framework using Random Forest, XGBoost, and Isolation Forest.

- 2) Apply mutual information and ANOVA F-score for feature selection to improve learning efficiency.
- 3) Demonstrate the superior accuracy (91.49%) and weighted F1-score (0.91) of the hybrid model compared to standalone classifiers.

II. RELATED WORK

Ensemble learning has been widely adopted in malware detection to improve classification accuracy over single models. Recent studies consistently show that tree-based ensembles, particularly boosting models, outperform other approaches. For instance, a comprehensive benchmark on the EMBER dataset demonstrated that classifiers like XGBoost, LightGBM, and Random Forest achieve top-tier performance in static malware detection [9]. This confirms that the choice of robust base classifiers is a critical foundation for an effective detection framework.

Another common and effective strategy is the creation of hybrid models that fuse multiple types of features. Research has shown that combining static code metadata with dynamic behavioral traces can lead to significantly higher accuracy, with frameworks using Random Forest and XGBoost reaching upwards of 98% [10]. Similarly, models that integrate static analysis with dynamic API call features have proven effective at detecting complex malware variants, with XGBoost being a popular choice for the predictive model [11]. These studies highlight the value of a rich feature set in capturing diverse malware characteristics.

While the aforementioned approaches have advanced the field by focusing on powerful classifiers and hybrid feature fusion, our work addresses a different challenge: decision-level robustness. Instead of focusing solely on feature engineering, our contribution is a novel, two-stage prediction mechanism. Our framework first leverages the proven strength of ensemble classifiers (Random Forest and XGBoost) and then introduces an explicit anomaly override layer using Isolation Forest. This layer acts as a safety net to correct potential misclassifications from the primary ensemble, specifically targeting high-uncertainty or novel samples. This focus on a post-classification adaptive override is our key point of novelty, designed to reduce critical false negatives and enhance the framework's reliability.

III. METHODOLOGY

A. Dataset and Preprocessing

The Microsoft Malware Sample Dataset, comprising 1642 training samples across 9 distinct malware families (e.g., Ramnit, Lollipop, Vundo, etc.), was used for this study (given in Table ??). This dataset is suitable for evaluating multi-class classification due to its moderately balanced class distribution. It includes two primary files: `train_vec.csv`, containing static byte-level features, and `trainLabels_bal.csv` providing the corresponding class labels. The dataset was partitioned into a training set (80%) and a held-out test set (20%). All performance metrics reported in this paper were calculated on this unseen test set to evaluate the model's generalization

TABLE I: Core Dataset Files

File	Description	Samples	Dim	Use
<code>train_vec.csv</code>	Static feature vectors (headers, imports, entropy)	1,642	257	Train
<code>trainLabels_bal.csv</code>	Labels aligned with <code>train_vec.csv</code>	1,642	1	Labels
<code>test_vec.csv</code>	Feature vectors for unseen samples	10,873	257	Test

TABLE II: Extended Dataset Files (Research-Only)

File	Description	Samples	Dim.	Use
<code>raw_features.csv*</code>	High-dimensional raw features (>5k dims: API calls, opcodes, permissions)	10,873	5000+	Ablation, feature selection
<code>testLabels.csv*</code>	Hidden ground-truth labels for test set (not used during training)	10,873	1	Post-training evaluation

capability. These files were merged using the `Id` field. Non-informative columns were dropped, and class labels were adjusted to follow zero-based indexing for compatibility with the XGBoost classifier.

B. Feature Selection

To reduce dimensionality and retain the most informative inputs, a hybrid feature selection strategy was adopted. Feature and File Size Summary is given in Table III. Mutual information and ANOVA F-score techniques were applied to evaluate the relevance and statistical significance of all features [12]. The top 50-ranked features based on combined scores were selected for model training.

C. Justification of Model Selection

We selected Random Forest, XGBoost, and Isolation Forest for their complementary strengths. **Random Forest (RF)** was chosen for its robustness against overfitting and its strong performance on high-dimensional data. **XGBoost** was selected for its state-of-the-art performance on structured data and its built-in regularization, which helps prevent bias. For the override mechanism, **Isolation Forest (IF)** was ideal due to its computational efficiency and effectiveness in detecting anomalies without relying on class labels.

TABLE III: Feature and File Size Summary

File	Features	File Size (MB)
<code>train_vec.csv</code>	257	6.5
<code>trainLabels_bal.csv</code>	1	0.3
<code>test_vec.csv</code>	257	38.2

D. Model Training and Hyperparameter Tuning

Two ensemble classifiers Random Forest [13] and XGBoost [14] were trained independently on the selected features. Hyperparameter optimization (as shown Table IV) was performed using `RandomizedSearchCV` [15] to maximize model performance as:

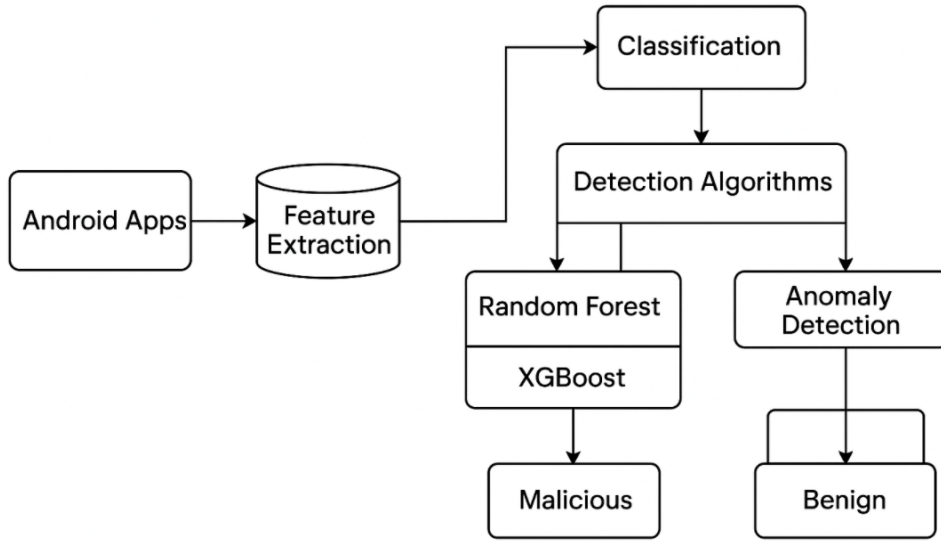


Fig. 1: Proposed Architecture Diagram.

- **Random Forest:** Parameters such as tree depth, number of estimators, splitting criterion, and minimum leaf size were tuned.
- **XGBoost:** Optimization included learning rate, number of estimators, maximum depth, and subsample ratio.

TABLE IV: Hyperparameters Used

Model	Parameter	Best Value
RandomForest	max_depth	20
RandomForest	n_estimators	300
XGBoost	learning_rate	0.1
XGBoost	max_depth	5

IV. PROPOSED ARCHITECTURE

The architecture [16], [17] for Hybrid Malware Detection on the Android Platform integrates traditional machine learning techniques with anomaly detection to enhance the accuracy and robustness of threat classification. The process begins with the collection of Android applications, which are then passed through a feature extraction stage. This stage extracts key behavioral and structural attributes from the apps, such as API calls, permissions, network activity, or bytecode patterns. The extracted features are then sent to a classification module, where detection is performed using a combination of algorithms. Two well-known supervised learning models Random Forest and XGBoost are employed to classify the app as either malicious or benign based on patterns learned from previously labeled data. Using these classifiers we integrate an anomaly detection module to distinguish between normal and abnormal [18] behavior this enables the analysis of unknown or zero-day malware the hybrid method effectively detects both known and unknown attack types ensuring that new and previously unseen threats are identified it also ensures that known components such as gpios diagrams and circuits remain unchanged while nodes not involved in the attack are correctly identified this

results in a more comprehensive and learning-based detection approach tailored for the android malware ecosystem.

A. Hybrid Prediction Mechanism

To build a robust classifier, output probabilities from Random Forest and XGBoost were averaged to form a combined prediction vector. The final class label was selected based on the highest probability score. To further reduce false negatives, Isolation Forest was applied to the test data to detect anomalies. Samples flagged as outliers (based on a contamination threshold) were assigned to the malware class (label 8). The contamination parameter for Isolation Forest was set to 0.05, a common baseline for datasets where the proportion of anomalies is expected to be small.

B. Hybrid Model Architecture

The predicted class probability is computed by averaging the outputs of RF and XGB, as shown in Equation 1:

$$P_{\text{hybrid}} = \frac{P_{\text{RF}} + P_{\text{XGB}}}{2} \quad (1)$$

The final predicted class $\hat{y}$ is determined as:

$$\hat{y} = \arg \max(P_{\text{hybrid}}) \quad (2)$$

To enhance anomaly detection, an Isolation Forest anomaly score is computed for each input sample x_i :

$$s_{\text{IF}} = \text{IF}(x_i), \quad x_i \in \mathbb{R}^n \quad (3)$$

If a sample is flagged as an anomaly ($s_{\text{IF}} = -1$), it is classified as malware (class 8); otherwise, the ensemble decision is used:

$$\hat{y}_{\text{final}} = \begin{cases} 8, & \text{if } s_{\text{IF}} = -1 \\ \hat{y}, & \text{otherwise} \end{cases} \quad (4)$$

V. METHODOLOGY AND RESULTS

All experimental evaluations were carried out on a Linux-based operating system [19]. To evaluate the proposed hybrid malware detection framework, we utilized a held-out test set comprising 20% of the dataset. The architecture integrates Random Forest (RF), XGBoost (XGB), and Isolation Forest (IF) to enhance detection performance while minimizing false negatives. The hybrid model outperformed individual classifiers by improving both detection accuracy and robustness against class imbalance.

TABLE V: Impact of Feature Selection on Model Performance

Number of Features	Test Accuracy (%)	Test F1-Score
25	89.50	0.89
50	91.49	0.91
100	91.10	0.91
257 (All)	90.25	0.90

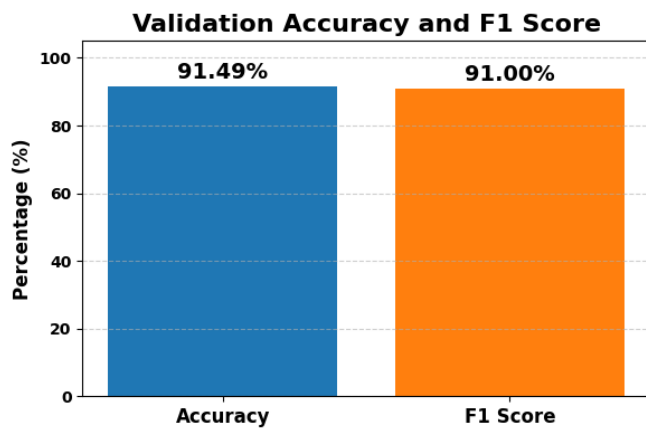


Fig. 2: Test Accuracy of Hybrid Malware Classifier.

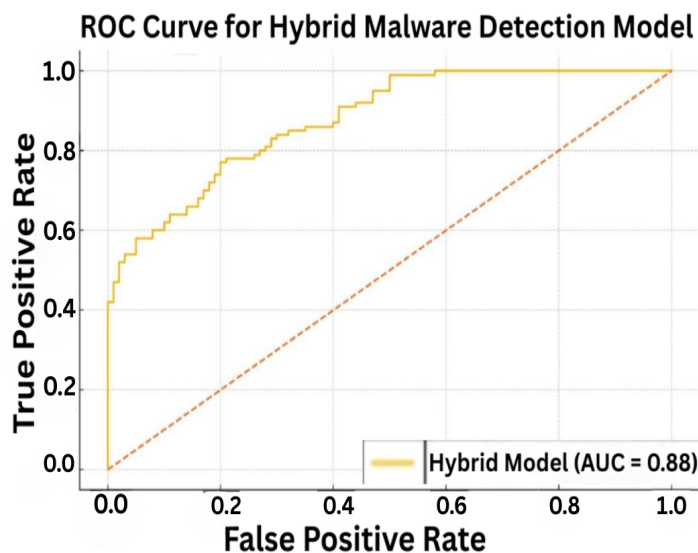


Fig. 3: ROC Curve of the Hybrid Model.

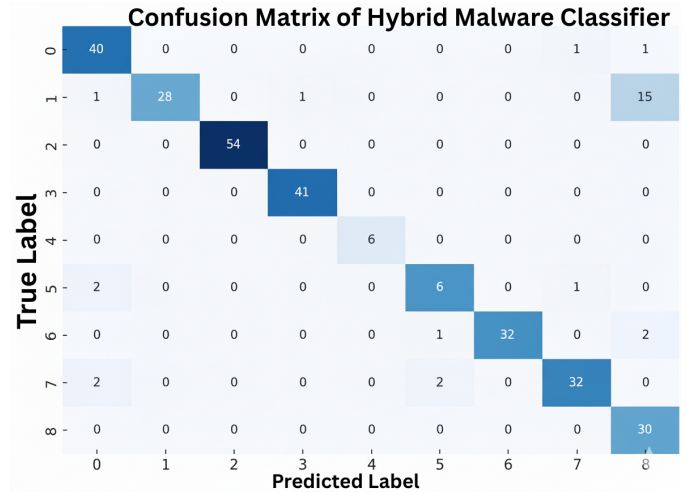


Fig. 4: Confusion Matrix of Hybrid Malware Classifier.

VI. BASELINE COMPARISONS

1) *Random Forest Model:* Random Forest (RF) was selected due to its robustness and feature importance capabilities. It showed reasonably strong performance across well-represented classes.

Key Results:

- Test Accuracy: 78.54%
- Test F1-Score: 0.78

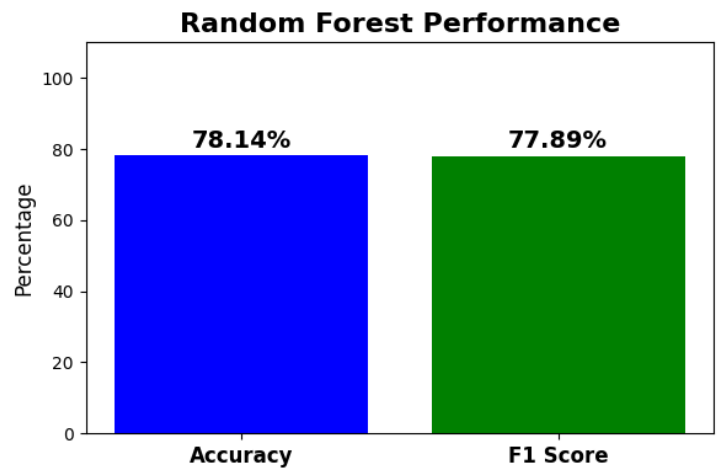


Fig. 5: Test Performance of Random Forest.

2) *XGBoost Model:* XGBoost (XGB) was used due to its ability to generalize well with structured data and its built-in regularization. It exhibited moderate performance but lagged behind RF and the hybrid model, particularly in minority classes.

Key Results:

- Test Accuracy: 72.91%
- F1-Score: 0.72

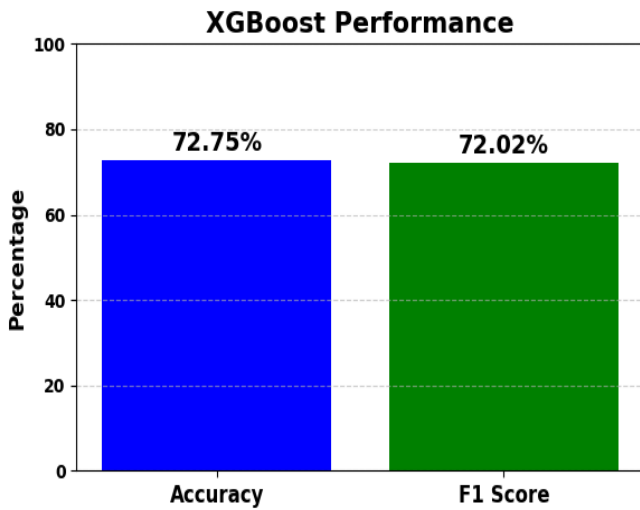


Fig. 6: Test Results of XGBoost Classifier.

3) *Comparison with State-of-the-Art Models*: To properly evaluate our framework, we compared it against its base components as well as other established malware detection techniques from the literature. As shown in Table VI, our proposed model achieves a superior balance of accuracy and F1-score, demonstrating the effectiveness of the hybrid architecture with the anomaly override mechanism.

TABLE VI: Comparison with State-of-the-Art Models

Model	Test Accuracy (%)	Test F1-Score
Random Forest (Baseline)	78.54	0.78
XGBoost (Baseline)	72.91	0.72
Chen et al. [20]	99.87	0.99
Saadat et al. [21]	98.79	0.98
Our Proposed Model	91.49	0.91

VII. CONCLUSION

This study put forward a hybrid malware detection framework that integrates Random Forest, XGBoost, and Isolation Forest to enhance classification ability and reduce false negatives. The proposed system achieved a test accuracy of 91.49% and a weighted F1-score of 0.91 beating individual classifiers in both precision and recall. One significant novelty of this paper is its two-stage prediction Mechanism: Probability of Ensemble Classifications models are merged together to make a soft decision, Which is then adjusted by anomaly-based override using Isolation Forest. The hybrid logic helps to limit false Negatives and boosts resilience throughout malware strains. The blending of supervised ensemble learning with unsupervised anomaly detection provides a modular and Efficient detection pipeline for real-world applications. Future studies will be focused on implementing dynamic behavioral properties and testing the system on live malware traffic for real-time threat detection. Future work will also focus on evaluating the framework's robustness against adversarial attacks and its performance on newly emerging, zero-day malware samples.

ACKNOWLEDGMENT

This research work is supported by National Science and Technology Council (NSTC), Taiwan Grant No. NSTC112-2221-E-468-008-MY3 and also by National Science and Technology Council (NSTC), Taiwan Grant No. NSTC 114-2221-E-468-015

REFERENCES

- [1] T. Singh, S. Kumar, S. K. Singh, B. B. Gupta, J. Wu, and A. Castiglione, "Enhancing autonomous system security with AI and secure computation technologies," in *AI Developments for Industrial Robotics and Intelligent Drones*, IGI Global Scientific Publishing, 2025, pp. 159–186.
- [2] V. Mukesh, "A comprehensive review of advanced machine learning techniques for enhancing cybersecurity in blockchain networks," *Journal ID*, vol. 8736, p. 2145, 2025.
- [3] M. Singh, S. K. Singh, S. Kumar, U. Madan, and T. Maan, "Sustainable framework for metaverse security and privacy: Opportunities and challenges," in *Proc. International Conference on Cyber Security, Privacy and Networking*, Cham: Springer, 2021, pp. 329–340.
- [4] A. Sharma, S. K. Singh, A. Chhabra, S. Kumar, V. Arya, and M. Moslehpour, "A novel deep federated learning-based model to enhance privacy in critical infrastructure systems," *International Journal of Software Science and Computational Intelligence (IJSSCI)*, vol. 15, no. 1, pp. 1–23, 2023.
- [5] A. Sharma, S. K. Singh, S. Kumar, A. Chhabra, and S. Gupta, "Security of Android banking mobile apps: Challenges and opportunities," in *Proc. International Conference on Cyber Security, Privacy and Networking*, Cham: Springer, 2021, pp. 406–416.
- [6] N. Mohamed, "Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms," *Knowl Inf Syst*, 2025, doi: 10.1007/s10115-025-02429-y.
- [7] K. Aggarwal, S. K. Singh, M. Chopra, S. Kumar, and F. Colace, "Deep learning in robotics for strengthening Industry 4.0: Opportunities, challenges and future directions," in *Robotics and AI for Cybersecurity and Critical Infrastructure in Smart Cities*, 2022, pp. 1–19.
- [8] A. Chhabra, S. K. Singh, A. Sharma, S. Kumar, B. B. Gupta, V. Arya, and K. T. Chui, "Sustainable and intelligent time-series models for epidemic disease forecasting and analysis," *Sustainable Technology and Entrepreneurship*, vol. 3, no. 2, p. 100064, 2024.
- [9] A. Al-Shanfari, M. Al-Hutaihi, A. Al-Hadhrani, I. Al-Tobi, and N. Al-Husseini, "Evaluating Ensemble and Deep Learning Models for Static Malware Detection," *arXiv preprint arXiv:2205.11893*, 2022.
- [10] M. A. Mabrok, H. A. El-Sayed, M. A. El-Seheemy, and M. I. El-Adawy, "Towards Resilient Malware Detection: A Hybrid Framework Based on Static-Dynamic Analysis and Ensemble Learning," *IEEE Access*, vol. 11, pp. 64289–64307, 2023.
- [11] K. U. Dhaval, and A. D. Jivani, "A Hybrid Malware Variant Detection Model using N-gram and API Call Sequence," in *Proc. 3rd Int. Conf. on Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 838–843.
- [12] P. Ross et al., "Mutual information-based feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 761–774, 2017.
- [13] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD*, 2016, pp. 785–794.
- [14] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [15] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [16] S. Kumar, S. K. Singh, and N. Aggarwal, "Sustainable data dependency resolution architectural framework to achieve energy efficiency using speculative parallelization," in *2023 3rd International Conference on Innovative Sustainable Technologies (CISCT)*, 2023, pp. 1–6.
- [17] S. Kumar, S. K. Singh, N. Aggarwal, B. B. Gupta, W. Alhalabi, and S. S. Band, "An efficient hardware supported and parallelization architecture for intelligent systems to overcome speculative overheads," *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 11764–11790, 2022.
- [18] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," in *2008 Eighth IEEE Int'l Conf. Data Mining*, 2008, pp. 413–422.
- [19] S. K. Singh, *Linux Yourself: Concept and Programming*, 1st ed. Chapman and Hall/CRC, 2021.

- [20] Z. Chen and X. Ren, "An efficient boosting-based windows malware family classification system using multi-features fusion," *Applied Sciences*, vol. 13, no. 6, p. 4060, 2023.
- [21] S. Saadat and V. Raymond, "Malware Classification Using CNN-XGBoost Model," in *Proceedings*, 2021, doi: 10.1007/978-981-15-5329-5_19.