Android Malware Detection Using Semantic Permission Analysis

Ali Hussein Mohammed Ali and Musaab Riyadh Abdulrazzaq
Department of Computer Science, College of Science, Mustansiriyah University, 10052 Baghdad, Iraq
ali.hussein.mohammed@uomustansiriyah.edu.iq, m.shaibani@uomustansiriyah.edu.iq

Keywords: Android Malware Detection, Static Analysis, Semantic Categories, Entropy Weighting, CRITIC, TOPSIS.

Abstract: Scalable Static analysis is progressively considered as the basis of Android malware detection, but the raw permission vectors are sparse, noisy and hard to interpret in the changing app ecosystems. In this paper, the authors of the research hypothesise a semantic permission-based detection pipeline to compress Android permissions into understandable capability sets and combine objective weighting with multi-criteria decision making (MCDM) to extract audit risk indicators. Using a conservative VirusTotal-based labeling policy to construct a balanced AndroZoo subset (38,062 apps) and test ablation baselines (becoming successively more inclusive of raw permission binary, semantic category statistics, objective weights (Entropy and CRITIC) and MCDM risk scores (TOPSIS/VIKOR). Using stratified holdout split, the optimal configuration (hybrid features with Random Forest) has an Accuracy of 83.16 and ROC-AUC of 91.73 (F1 = 83.25, MCC = 0.663), indicating that the inclusion of MCDM-derived risk signals can both increase the performance in terms of Discriminative and Increase the interpretability. The suggested design is aimed at large-scale screening processes whereby predictive and transparent risk cues should be present besides prediction. It has limitations: it depends on VirusTotal-style labels (subject to engine disagreement and temporal drift) and the existing evaluation is not yet external-dataset-generalization, temporal split validation, or robustness against obfuscation, and these are put as prioritized extensions to future work.

1 INTRODUCTION

Android is still a major mobile platform, and its openness and vast app ecosystem still draws malware and repackaged apps that take advantage of the distribution of apps in the app-store and the third-party components [1] Scalable detection has emerged as a viable need as malware has grown and become a large-scale operation. Of the most typical analysis paradigms (static, dynamic, hybrid), static analysis is especially appropriate in the case of high-throughput screening since it analyzes artifacts of an application but does not execute them [2]. Nevertheless, it is difficult to completely align studies because of the variability of features selection, labeling, and experimental procedures, and little information is available regarding the split techniques and leakage provisions in most studies [3]. Permissions have become a popular first-line static signal, as one can be effectively extracted out of the android manifest, and they have demonstrated discriminative ability in identifying malware [4], [5]. However, permissions are capabilities that are requested and not observed, and they might be subject to ecosystem drift and Android permission model evolution [6]. Emerging studies are thus moving more and more towards richer and more context-sensitive representations, highlighting how the meaning of permission can be situation-sensitive and better represented by higher level abstractions like semantic grouping [7], [8]. Based on these findings, the article suggests an interpretable framework of static detection, which (i) projects raw permissions into small category of semantic capabilities to reduce sparsity and enhance interpretability, and (ii) uses objective and data-driven features weighing to rank security-relevant categories and reduce redundancy. The paper tests the framework using a large AndroZoo subset [9], [10] and clearly, the research are cognizant of known VirusTotal-based labeling limitations (engine disagreement and temporal drift) that impact comparability. The main contributions are:

1) We introduce a semantic permission encoding that maps raw Android permissions into 8 interpretable capability categories and generates compact presence/count descriptors;

2) We compute objective category weights using Entropy and CRITIC, reducing ad-hoc feature prioritization and improving auditability;
3) We integrate MCDM (TOPSIS/VIKOR) to produce explicit risk scores that complement classification and support transparent decision-making;
4) We provide an ablation-driven evaluation on a balanced AndroZoo subset, quantifying the accuracy-interpretability trade-off across raw, semantic, weighted, and hybrid representations.

2 RELATED WORK AND MOTIVATION

Large-scale Android malware screening is popularly done by permission based static detection since permission is cheap to extract and can be represented in binary vectors to run them with classical ML models [5], [6]. Permission requests are claims of capabilities, however, which do not reflect empirical behavior, and their usefulness may be sensitive to data artifacts, ecosystem dynamics and the effect of obfuscation [5], [6]. In order to add more robustness than naive binary indicators, previous research has investigated structural and dependency-sensitive representations of permissions like permission-pair modeling through composition ratios [11] and correlated real permission couples learned through permission interactions [12]. Simultaneously, a number of studies also stress the context-specificity of permission meaning, and this drives category-conscious modeling learning permission usage patterns across app categories [7], and sensitivity-constrained weighting schemes distinguishing between over-privilege and missing-privilege evidence in a category context [8], [12]. In a broader sense, semantic clustering at API level has been also cited to be drift-aware and served as a slow-aging approach in the applications of drift-aware abstraction and serves to enforce the importance of representations that do not change with time but instead are grounded on fixed vocabularies [13]. Even though multi-view methods may enhance representational coverage by combining complementary behavioral perspectives, they generally make the triage procedures more complex and less transparent, which is a significant attribute of scalable and auditable triage processes [14]. AndroZoo is a dataset with the properties that, up to now, allows constructing large-scale subsets using the constantly growing repository and metadata index [9] although VT-based labeling is still noisy and time-dependent, thus explicit VT thresholding and scan-context reporting are needed to allow fair comparison across studies. Driven by these observations, this paper integrates types of semantic permission capabilities with objective feature weighting to create an auditable risk signal on an AndroZoo-derived subsample with explicit VT thresholding. Tables 1 provides the comparison of the checklist with the most topical related studies.

Although permission-based detectors are extensively analyzed, most previous studies assume permissions as binary direct vectors (or frequency features) and are mainly interested in predictive metrics optimization. MCDM is in contrast seldom employed in Android malware detection (i) many studies do not explicitly represent detection as a criteria-based decision process, (ii) subjective weighting is often inadvisable due to the dependence on judgments by experts and (iii) evaluation protocols are typically designed to measure the performance of a classifier instead of interpretable, re-weightable risk scoring. We fill this void with the semantic permission categories that are coupled with objective weights (Entropy/CRITIC) and explicit MCDM scoring layer (TOPSIS/VIKOR), which allows us to prioritize transparently and adjust the thresholds about changing the labeling circumstances.

Table 1: Checklist comparison between this article and related studies.

Study	Permission features	Semantic Patterning	Feature weighting	Risk score (MCDM)	Aging / drift discussed
[5]	✓	X	X	X	X
[6]	✓	X	X	X	X
[7]	✓	✓	X	X	X
[8]	✓	✓	✓	X	X
[11]	✓	X	X	X	X
[12]	✓	X	X	X	X
[13]	X	✓	X	X	✓
[14]	X	✓	X	X	X
This Article	✓	✓	✓	✓	✓

3 PROPOSED METHODOLOGY

The workflow starts from AndroZoo metadata using sha256 as the unique identifier, along with vt_detection and permissions (1) [9], where duplicate records are removed via sha256-based de-duplication (2) [9]. Samples are then labeled using VirusTotal detection counts: apps with vt_detection = 0 are treated as benign, apps with vt_detection ≥ 10 are treated as malware, and the ambiguous range (1-9) is excluded to reduce label noise and antivirus disagreement. After labeling, a balanced subset is constructed 38,062 apps: 19,031 malware and 19,031 benign (3) and (4) [15], [16] to avoid class-imbalance bias in evaluation. Next, static permissions are extracted and normalized from permissions standardizing the raw permission strings to ensure consistent matching. Instead of keeping sparse raw permission binaries only, convert each permission into a semantic permission type using the reference mapping permission clusters by type, producing type IDs (1-8) representing higher-level categories. Based on these semantic types, generating compact interpretable features for each app, including presence indicators (has_c) and frequency features (cnt_c) per semantic category and keep the raw permissions only when needed, otherwise rely on the semantic representation. Then, computing objective feature weights using Entropy weighting for information-content driven and CRITIC weighting for standard deviation and correlation-driven. In addition, normalizing the resulting weights, and applying MCDM risk scoring (TOPSIS) and (VIKOR) to obtain a per-application risk score aligned with generated scoring feature, permission weight score, and TOPSIS score. Finally, evaluate lightweight classifiers using Random Forest and Logistic Regression under an ablation design with six baselines: B0 raw permission binaries, B1 semantic category features (has_/cnt_) with equal weights, and B2 semantic category features weighted by Entropy and CRITIC with TOPSIS-based risk scoring. In addition, B2+ is a combination of semantics and MCDM risks. However, to improve the concept B3 works on semantics and raw to check the validity of model. Performance is reported using Accuracy, Precision, Recall, F1-score, and AUC.

3.1 Dataset Subset Construction Labeling and Evaluation Protocol

This article constructs a labeled dataset using a VirusTotal detection-count policy to reduce label ambiguity and improve the reliability of downstream evaluation. Specifically, an application is labeled Benign when vt_detection = 0 and labeled Malware when vt_detection ≥ 10. Samples with intermediate detections (1 ≤ vt_detection ≤ 9) are excluded to avoid uncertain or noisy labels that can distort comparisons and inflate performance as shown in Figure 1. After labeling, remove duplicates using a unique application identifier by cryptographic hash (1), (2) to prevent the same app from appearing across training and testing partitions. Finally, evaluate all configurations using a stratified 80/20 holdout split, and additionally report stratified 5-fold cross-validation to assess stability beyond a single split. Table 2 shows the detection rules used in the methodology.

Table 2: Labeling policy and evaluation protocol.

Class	Label rule	Samples
Benign	vt_detection = 0	19,031
Malware	vt_detection >= 10	19,031
Excluded	1 <= vt_detection <= 9	N/A

$$\mathcal{D} = \{(h_i, vt_i, P_i)\}_{i=1}^{N}. \quad (1)$$

where:

- h_i is the SHA-256 identifier;
- vt_i is the VirusTotal detection count;
- P_i is the permission set extracted from Dataset.

$$\mathcal{D}_{uniq} = \{(h, vt(h), P(h)) \mid h \in \text{Unique}(\{h_i\})\}, \quad (2)$$

$$N_b = \min(N_0, N_1). \quad (3)$$

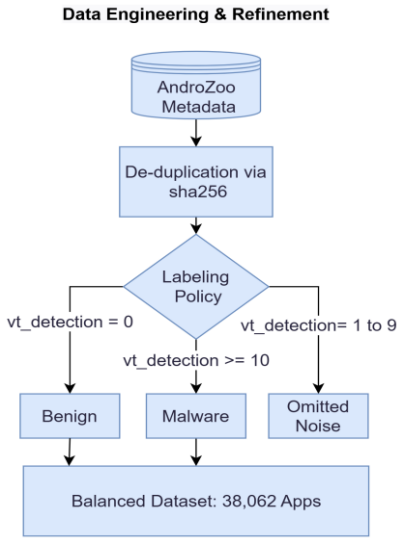

Figure 1: Data preparation and labeling protocol.

This step removes duplicate records by keeping a single canonical entry per unique SHA-256. Let N_0and N_1denote the number of benign and malware apps after filtering, resulting in 2N_bapps (38,062 apps). A balanced subset is formed as:

$$\mathcal{D}_{bal} = \text{Sample}(\mathcal{D}_{y=0}, N_b) \cup \text{Sample}(\mathcal{D}_{y=1}, N_b). \quad (4)$$

To minimize weak-label noise and eliminate unclear samples near the decision boundary, we apply a labeling strategy on VirusTotal which is conservative. As we observe, VirusTotal evidence is a time-dependent type of evidence: engine participation and signatures change over scan dates and may cause time bias and label drift. The number of engines at scan time is not always available on all samples in our metadata snapshot so we are using vt_detection as a convenient weak-label measure and regard this as a limitation. In spite of the fact that our primary assessment relies on stratified splits to compare controlled ablation studies, time (time-split) validation has to explicitly test concept drift, and is intended as a future addition.

3.2 Semantic Permission Category Encoding

From each application, extracting the requested permissions and encoding them into two complementary views to balance discriminative power and interpretability. First, construct a raw-permission baseline by representing permission as a sparse binary vector that indicates whether each permission is requested by the app presence or absence (5), (6) [17]. Second, map each permission into 8 semantic categories and derive compact category-level descriptors, including per-category counts (cnt_*) and binary presence flags (has*) (7), (8), (9) [18] as shown in Figure 2. This semantic encoding reduces dimensionality and enables category-level reasoning, which is later exploited by objective weighting and risk scoring components of the proposed pipeline (10) [19]. Table 3 provides 8 selected categories with its description. Let {q_1,…,q_K } be the global vocabulary of unique normalized permissions. For app i:

$$b_{i,k} = \begin{cases} 1 & \text{if } q_k \in P_i^{norm} \\ 0 & \text{otherwise} \end{cases}. \quad (5)$$

$$\mathrm{b}_i = [b_{i,1}, \ldots, b_{i,K}] \in \{0,1\}^K. \quad (6)$$

$$t = g(p^{norm}), t \in \{1, \ldots, C\}. \quad (7)$$

where $g(p^{norm})$ is the lookup mapping from permission_clusters_by_type, and C=8 semantic categories.

$$\mathrm{has}_{i,c} = \begin{cases} 1 & \text{if } \exists\, p \in P_i^{norm} : g(p) = c \\ 0 & \text{otherwise} \end{cases}. \quad (8)$$

$$\mathrm{cnt}_{i,c} = \sum_{p \in P_i^{norm}} \mathbb{I}[g(p) = c]. \quad (9)$$

$$\mathrm{x}_i = [\mathrm{has}_{i,1}, \ldots, \mathrm{has}_{i,C}, \mathrm{cnt}_{i,1}, \ldots, \mathrm{cnt}_{i,C}]. \quad (10)$$

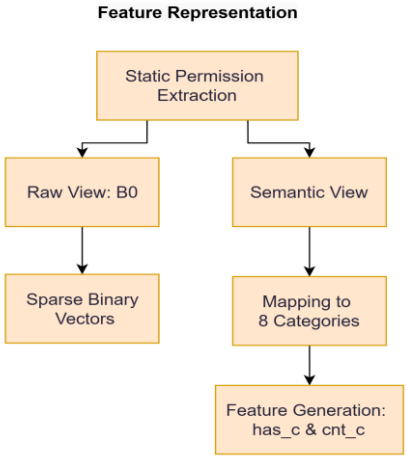

Figure 2: Features representations.

Table 3: Semantic categories and examples.

Semantic categories	Description
Location	Permissions related to GPS/network location access
Contacts	Permissions for reading/writing contacts and account-related contact data.
SMS	Permissions for sending/receiving/reading SMS and related messaging actions.
Camera	Permissions to access the device camera and capture media
Storage	Permissions for reading/writing external storage and managing files
Phone	Permissions related to calls, phone state, and telephony identifiers
System	System-level or sensitive device settings/actions
Network	Permissions related to connectivity state, internet access, and network communication

The category set is meant to be interpretable and to cover the largest possible number of the most security relevant capabilities of the Android. As opposed to the hundreds of sparse raw permissions, we cluster them into eight capability families which are often related to privacy/safety impact e.g., communication, location, storage, device identifiers and generate compact presence/count descriptors that can be screened at scale. In order to prevent arbitrary feature inclusion, we also compare top-k sensitivity

to ablation by (i) comparing (ii) the full semantic representation versus reduced variants with only the most informative components (top-k) with the objective weighting stage. The findings reveal that the performance is steadily increasing with a transition to augmented features with weights/MCDM, which suggests that the chosen category layer preserves the dominant discriminative signal and can be audited. We report these top-k variants explicitly in the ablation results, which provides an empirical sensitivity check for compact representations.

3.3 Objective Weighting

In order to avoid subjective prioritization of permission categories. The study computes objective weights for the eight semantic categories using Entropy and CRITIC. Entropy assigns higher importance to categories with richer information content across the dataset, whereas CRITIC emphasizes categories that exhibit stronger contrast while accounting for redundancy through inter-feature relationships as shown in Figure 3. The final weight profile used in pipeline is obtained by a simple aggregation of these two objective schemes (11) [20], defined as:

$$w_{\text{combined}} = \frac{1}{2}(w_{\text{entropy}} + w_{\text{critic}}). \quad (11)$$

Followed by normalization so that the weights sum to one. This combined weight vector is then used to construct weighted semantic descriptors (weighted_cnt_*) and to support downstream MCDM-based risk scoring (12) [21]. For completeness, also report an AHP-based profile as an optional expert prior; however, the final combined weights used in this study are computed only from Entropy and CRITIC. Figure 3 shows permission weight in selected subset of AndroZoo dataset.

Let $X = [x_{i,j}]$ be the matrix of semantic features $(j = 1, \dots, m)$ over $n$apps.

$$r_{i,j} = \frac{x_{i,j} - \min_i x_{i,j}}{\max_i x_{i,j} - \min_i x_{i,j}}. \quad (12)$$

The Entropy technique of Shannon is applied in order to define the objective significance of each type of permission. In this method of calculating the weights, the values are computed using the extent of the disparity of the feature values. An increase in the value of entropy means that the feature itself offers much information to discriminate against the classification process, and thus subjective bias when it comes to selection of the feature is minimized as shown in (13), (14) [22], [23].

$$e_j = -k \sum_{i=1}^{n} p_{i,j} \ln(p_{i,j}), k = \frac{1}{\ln(n)}. \quad (13)$$

$$w_j^{(E)} = \frac{d_j}{\sum_{j=1}^{m} d_j}. \quad (14)$$

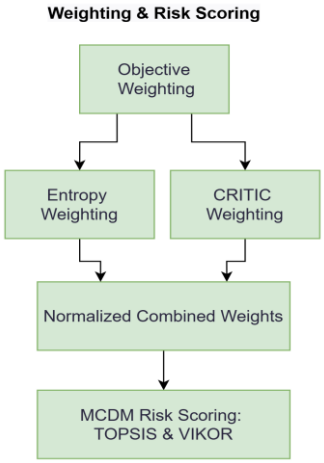

Figure 3: Weighting and risk scoring.

CRITIC (Criteria Importance Through Inter-Criteria Correlation) technique which is used to refine the feature weights based on the contrast intensity as well as the conflict between permission features. This approach uses the correlation matrix to penalize redundant features as the relevant ones that have independent signals are given precedence in the malware detection pipeline as shown in (15), (16), (17) [24].

$$\rho_{j,k} = \text{corr}(r_{\cdot,j}, r_{\cdot,k}). \quad (15)$$

$$C_j = \sigma_j \sum_{k=1}^{m} (1 - \rho_{j,k}). \quad (16)$$

$$w_j^{(C)} = \frac{C_j}{\sum_{j=1}^{m} C_j}. \quad (17)$$

we adopt an equal-weight fusion in (18) [20].

$$w_j = \frac{1}{2}\left(w_j^{(E)} + w_j^{(C)}\right). \quad (18)$$

3.4 Risk Scoring via MCDM

Building on the weighted semantic representation, derive continuous risk indicators using multi-criteria decision-making (MCDM). Specifically, compute TOPSIS and VIKOR scores over the weighted

category descriptors to produce two complementary features, risk_score_topsis and risk_score_vikor. In addition, report a correlation heatmap (Fig. 5) to provide a diagnostic view of how the MCDM risk indicators relate to the weighted semantic category features. The results show a very high correlation between TOPSIS and VIKOR ($\rho \approx 0.97$), indicating that both methods produce largely consistent rankings under feature design. also observe strong associations between the risk scores and certain weighted categories, most notably the System category ($\rho \approx 0.91$ with TOPSIS and $\rho \approx 0.95$ with VIKOR) which supports the interpretation that high-risk scores are primarily driven by system-level permission behaviors. This analysis is included to document feature relationships and potential redundancy; however, both scores are retained as auxiliary continuous features since the downstream classifier can handle correlated inputs. These scores summarize the permission-driven risk of an application on a continuous scale and provide an interpretable signal that can be inspected independently. Importantly, do not treat risk scores as standalone decision rules; instead, they are appended as additional features within the learning model to support both predictive performance and explainable reporting. Figure 5. Correlation heatmap between the MCDM (Fig. 4)-derived risk scores (TOPSIS and VIKOR) and the weighted semantic category counts. The heatmap is used as a diagnostic view to examine redundancy and the dominant contributing categories behind the risk indicators.

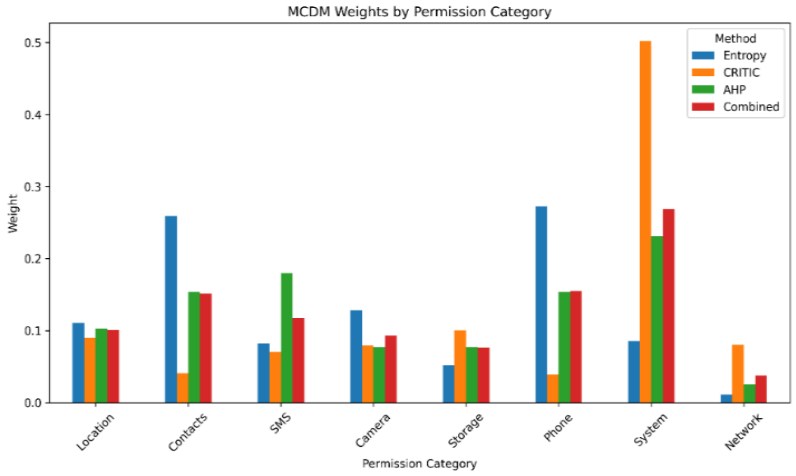

Figure 4: MCDM weights by permission category.

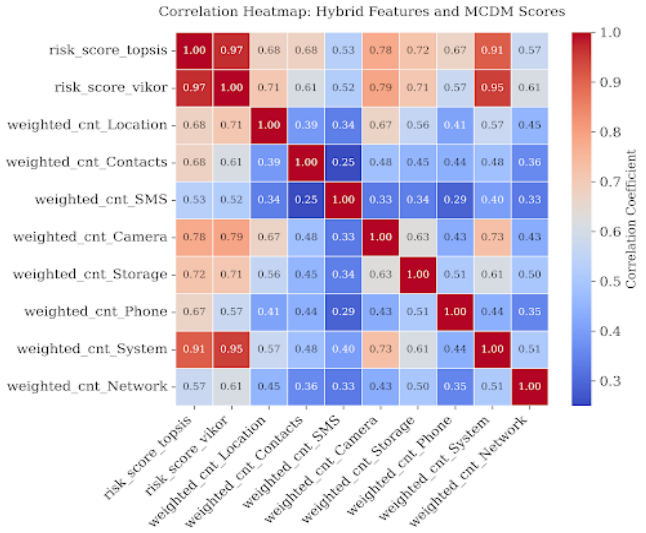

Figure 5: Correlation structure of MCDM risk scores and weighted semantic features.

A multi-criteria ranking is carried out in order to measure the risk posed by a particular application by TOPSIS and VIKOR algorithms. The approaches calculate a comprehensible Risk Score by calculating the geometric distance between the feature vector of an app and an ideal malicious example and a negative-ideal benign example. The score is a continual indicator that improves interpretation of the final decision of the model as shown in (19), (20), (21), (22) [25], [26]. Let $i \in \{1, \dots, n\}$denote an application (alternative) and $j \in \{1, \dots, m\}$denote a criterion (feature). The raw value of criterion $j$for application $i$is $x_{i,j}$, and its min-max normalized value is $r_{i,j}$. The objective weight of criterion $j$is w_j, and the weighted normalized value is $v_{i,j} = w_j r_{i,j}$. The positive and negative ideal solutions are $A^+ = \{v_j^+\}$and $A^- = \{v_j^-\}$, where $v_j^+ = \max_i v_{i,j}$and $v_j^- = \min_i v_{i,j}$for benefit-type criteria (swapped for cost-type criteria). The Euclidean distances from application $i$to the ideals are $D_i^+$and D_i^-. Finally, the TOPSIS closeness score is $S_i = \frac{D_i^-}{D_i^+ + D_i^-}$, where larger $S_i$indicates closer proximity to the ideal best.

$$v_{i,j} = w_j \, r_{i,j}. \tag{19}$$

For benefit features:

$$A^+ = \left\{v_j^+ = \max_i \quad v_{i,j}\right\}, A^- = \left\{v_j^- = \min_i \quad v_{i,j}\right\}. \tag{20}$$

$$D_i^+ = \sqrt{\sum_{j=1}^{m} (v_{i,j} - v_j^+)^2}, D_i^- = \sqrt{\sum_{j=1}^{m} (v_{i,j} - v_j^-)^2}. \tag{21}$$

$$S_i = \frac{D_i^-}{D_i^+ + D_i^-}. \tag{22}$$

3.5 Hybrid Feature Sets and Learning Setup

To quantify the contribution of each design component, evaluate a structured set of feature configurations under a unified learning protocol. B0 serves as the raw-permission baseline using sparse binary indicators of requested permissions. B1 augments the semantic encoding with unweighted category descriptors (per-category counts and presence flags) and risk scores, while B2 introduces objective weighting to emphasize the most informative categories. B2+ extends this explainability-oriented track by incorporating both MCDM risk indicators (risk_score_topsis and risk_score_vikor) alongside the weighted semantic descriptors. To maximize predictive performance while retaining interpretability, define hybrid configurations: B3 concatenates raw permissions with the B2+ semantic-risk features, and B3-lite-k is a compact hybrid variant that retains B2+ while adding only the Top-k most informative raw permissions (k ∈ {20, 50, 100}) selected using training data only. For classification, adopt Random Forest as the primary model due to its strong performance on sparse/binary representations and its compatibility with importance-driven analysis; additional baselines can be compared under the same feature-set definitions. All experiments follow a stratified 80/20 holdout split and stratified 5-fold cross-validation, reporting Accuracy and ROC-AUC alongside complementary metrics such as Precision, Recall, F1-score, and confusion-matrix-based error analysis. Table 4 represents a comparison between the suggested baselines with the categories chosen. B1 uses unweighted semantic descriptors (cnt_*, has_*), B2 uses weighted semantic counts (weighted_cnt_*), B2+ appends TOPSIS/VIKOR risk scores, and hybrid settings (B3/B3-lite) combine raw-permission indicators with semantic-risk features; Top-k permissions are selected using training data only. The experimental design to be used is aimed at isolating the impact of semantic permission encoding and the MCDM-based scoring in a controlled AndroZoo protocol (stratified holdout and cross-validation).

Table 4: Semantic categories and examples.

Feature set	Raw permissions	Semantic cnt *	Semantic has *	Objective weights (weighted_cnt *)	TOPSIS risk	VIKOR risk	Top-k selection
B0	✓	X	X	X	X	X	X
B1	X	✓	✓	X	X	X	X
B2	X	X	X	✓	X	X	X
B2+	X	X	X	✓	✓	✓	X
B3	✓	X	X	✓	✓	✓	X
B3-lite-20	✓	X	X	✓	✓	✓	✓
B3-lite-50	✓	X	X	✓	✓	✓	✓
B3-lite-100	✓	X	X	✓	✓	✓	✓

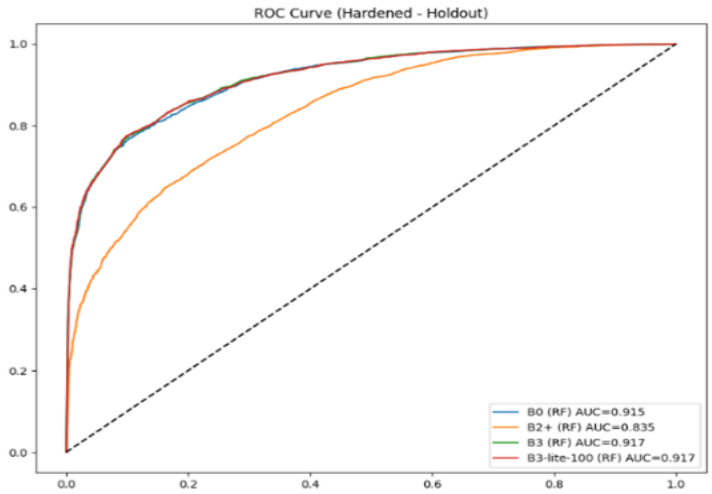

Figure 6 ROC curve.

We recognize that to have stronger external validity, it is necessary: (i) to be tested on an external dataset (e.g. Drebin /AMD-style corpora), (ii) to be split validation (trained on earlier scan periods, tested on later periods) to explicitly estimate concept drift, (iii) to be resistant to repackaging and permission inflation/deflation (testing and validation), and (iv) to compare with deep learning baselines under the same labeling and split protocol. These tests have been given the first priority due to follow-up validations to determine the generalization outside the current environment as well as reinforce deployment-oriented evidence.

4 EXPERIMENTAL RESULTS

This section reports the empirical evaluation of the proposed permission-based malware detection pipeline under a unified protocol. present results for both the performance-oriented hybrid representations (B3 and B3-lite) and the explainability-oriented semantic representations (B1/B2/B2+), aiming to quantify (i) predictive performance on an unseen test split, (ii) stability under cross-validation, and (iii) the practical contribution of MCDM-derived risk scores when integrated as features. The superiority of the hybrid model (B3) is attributed to its ability to combine high-level semantic categories with fine-grained raw permission signals, providing a comprehensive feature space for the Random Forest classifier. All experiments follow a stratified 80/20 holdout split and stratified 5-fold cross-validation, and report Accuracy and ROC-AUC as primary metrics, complemented by F1-score and other standard indicators. Figure 6. summarizes the statements above.

4.1 Holdout Performance

The best overall result is obtained by the proposed hybrid representation B3 with Random Forest, achieving Accuracy = 83.16% and ROC-AUC = 91.73%. The compact hybrid variant B3-lite-100 with Random Forest achieves a nearly identical outcome (Accuracy = 83.16%, ROC-AUC = 91.66%), indicating that combining semantic-risk features with a limited set of highly informative raw permissions can preserve most of the discriminative signal. The raw-permission baseline B0 with Random Forest remains strong (Accuracy = 82.52%, ROC-AUC = 91.41%), whereas the explainability-oriented semantic track (B2+ with Random Forest) yields lower but still competitive performance (Accuracy 73.97%, ROC-AUC 83.53%), reflecting the trade-off between compact interpretability and fine-grained raw-permission detail. Table 5 and Table 6 summarize holdout performance across all feature-set configurations and model backbones. However, Figure 7 shows Accuracy and ROC-AUC across ablation baselines.

Notably, B3-lite-100 achieves nearly identical performance to B3 while reducing inference time by approximately 2×, making it a practical choice for large-scale screening.

While ROC-AUC summarizes threshold-independent ranking performance, deployment-oriented screening also depends on the precision-

recall trade-off. Therefore, we additionally report a precision-recall profile across the ablation baselines for the RF model as shown in Figure 8.

Table 5: Holdout results.

Feature_Set	Model	Accuracy	AUC	F1
B3	RF	83.16%	91.73%	83.25%
B3-lite-100	RF	83.16%	91.66%	83.24%
B3-lite-50	RF	82.83%	91.54%	82.95%
B0	RF	82.52%	91.41%	82.49%
B3-lite-20	RF	81.43%	90.33%	81.54%
B2+	RF	73.97%	83.53%	73.96%
B2	RF	73.81%	83.50%	73.89%
B1	RF	73.69%	83.40%	73.75%

The hybrid baseline (B3) has the best precision and recall at the same time, semantic-only baselines (B1 to B2+) have lower values, which is the anticipated trade-off between small interpretability and discriminative detail. This diagnostic image will supplement the ROC-AUC comparison and explain the behavior of the suggested representations in operations.

Figure 9 reports the confusion matrix for the best holdout configuration (B3 + RF): TN = 3130, FP = 677, FN = 621, and TP = 3185. Accordingly, the false positive rate is FPR = FP/(FP+TN) = 677/(677+3130) = 0.1778 (17.78%), and the false negative rate is FNR = FN/(FN+TP) = 621/(621+3185) = 0.1632 (16.32%). These values indicate relatively balanced error behavior, which is desirable for malware triage where both missed malware and excessive false alarms are costly. On the holdout test set (N = 7613), the best setting (B3 + RF) achieves Accuracy = 83.16% with an approximate 95% binomial CI of [82.30%, 83.98%]. For ROC-AUC = 0.9173, a standard large-sample approximation gives an approximate 95% CI of [0.9108, 0.9238]. The McNemar test takes per-sample paired prediction of two models; check because this form of comparison returns aggregate values, we have McNemar comparison in the extended evaluation by storing per-instance prediction records.

4.2 Top 10 RF Feature Importance

Analysis of feature-importance demonstrates that the risk scores suggested by MCDM take up most of the decision signals in the model: risk score VIKOR and Risk score TOPSIS about 40.1% of the total importance, meaning that objective weighting and ranking are able to capture a strong discriminative risk pattern. The following indicators with the strongest weighted counts are category-level weighted counts, especially Network and System, and then comes Storage and Camera, which is in line with intuition that those capabilities and communication-related behaviors are sensitive and therefore informative in the process of screening malware correctly at its static. Table 7 shows Feature Importance and Interpretability.

4.3 Cross-Validation Stability

To confirm that holdout performance is not an artifact of a single train-test split, additionally report stratified 5-fold cross-validation results for the main Random Forest configurations. As shown in Tables 8, 9, 10, the best-performing hybrid representation B3 achieves 82.55% ± 0.58% accuracy and 91.13% ± 0.38% ROC-AUC, indicating stable performance across folds. The compact hybrid variant B3-lite-100 yields nearly identical stability (82.53% ± 0.58% accuracy, 91.11% ± 0.35% ROC-AUC), supporting the effectiveness of reducing the raw-permission space while retaining semantic-risk features. In contrast, the explainability-oriented setting B2+ attains 73.41% ± 0.40% accuracy and 83.23% ± 0.39% ROC-AUC, reflecting a consistent trade-off: semantic compression improves interpretability but reduces discriminative detail compared to hybrid representations.

Table 6: Summary of predictive performance under holdout and cross-validation.

feature_set	model	tn	fp	fn	tp	test_time
B0	RF	3149	658	661	3145	0.034066677
B1	RF	2796	1011	992	2814	0.029779673
B2	RF	2797	1010	984	2822	0.034452915
B2+	RF	2812	995	991	2815	0.032730579
B3	RF	3145	662	615	3191	0.067857265
B3-lite-100	RF	3152	655	624	3182	0.034764290

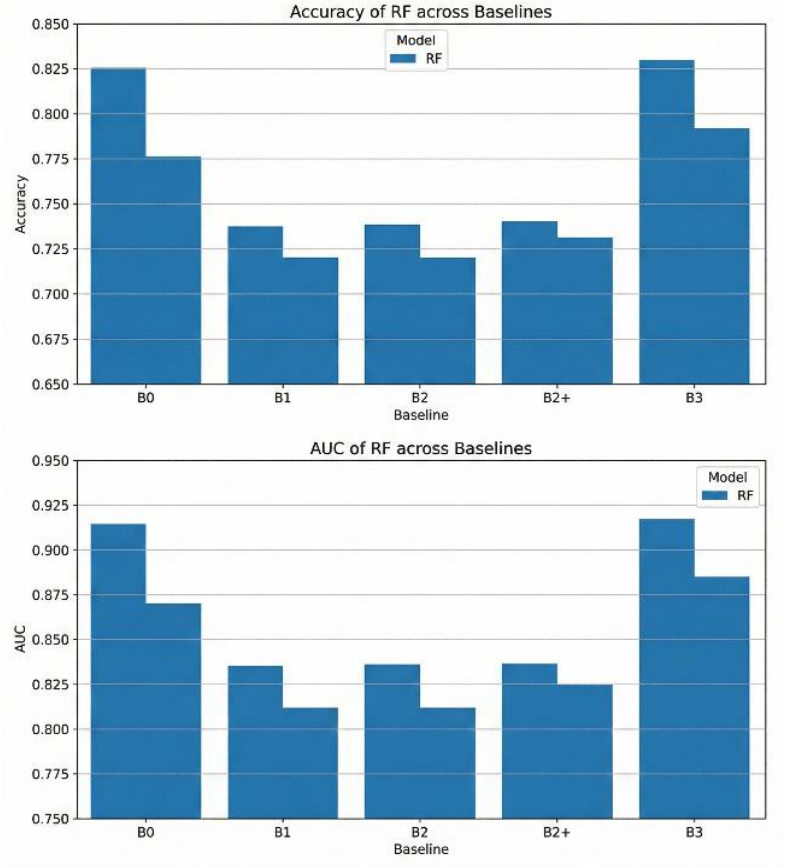

Figure 7: Accuracy and ROC-AUC across ablation baselines.

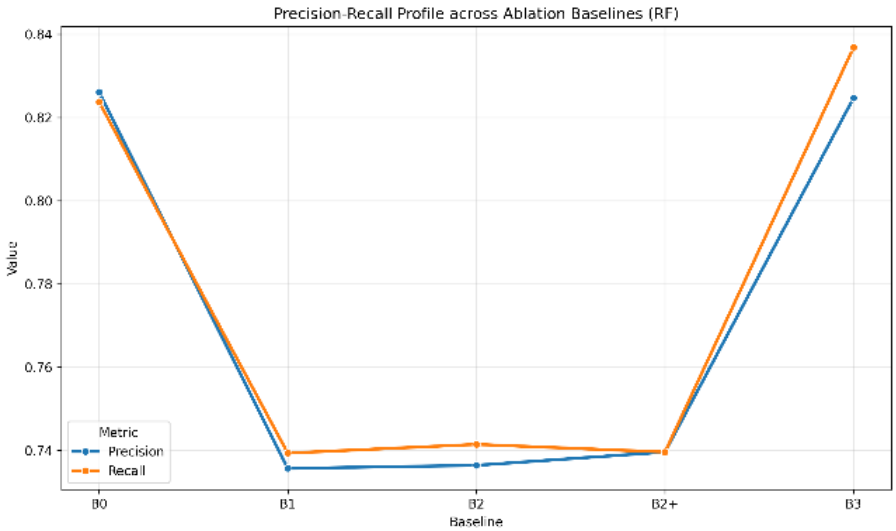

Figure 8: Precision-Recall profile across ablation baselines (RF).

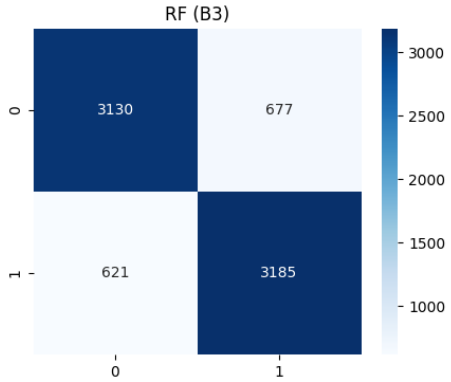

Figure 9: Confusion matrix for the best configuration (B3 + RF) on the holdout set.

Table 7: Top features by random forest importance.

Rank	Feature	Importance
1	risk_score_vikor	0.201744
2	risk_score_topsis	0.199108
3	weighted_cnt_Network	0.141889
4	weighted_cnt_System	0.119517
5	weighted_cnt_Storage	0.088807
6	weighted_cnt_Camera	0.082087
7	weighted_cnt_Location	0.062622
8	weighted_cnt_SMS	0.062184
9	weighted_cnt_Contacts	0.022488
10	weighted_cnt_Phone	0.019554

Table 8: Stratified 5-fold cross-validation result 1.

Feature Set	Model	Acc_Mean	Acc_Std
B3	RF	82.55%	0.58%
B2+	RF	73.41%	0.40%
B3-lite-100	RF	82.53%	0.58%

Table 9: Stratified 5-fold cross-validation result 2.

Feature Set	Model	Auc_Mean	Auc_Std
B3	RF	91.13%	0.38%
B2+	RF	83.23%	0.39%
B3-lite-100	RF	91.11%	0.35%

Table 10: Stratified 5-fold cross-validation result 3.

Feature Set	Model	F1_Mean	Mcc_Mean
B3	RF	82.41%	65.11%
B2+	RF	73.25%	46.83%
B3-lite-100	RF	82.41%	65.08%

4.4 Risk Score Analysis

Further analyze the behavior of the MCDM-derived risk indicators to clarify their practical role within the proposed pipeline. Figure 7 shows that the TOPSIS risk score distributions of benign and malware apps exhibit substantial overlap in the central region: the medians are close (Benign: 0.0225 vs. Malware: 0.0210), while the mean is higher for malware (0.0410 vs. 0.0323) due to a heavier right tail and more extreme outliers. This pattern is consistent with the statistical tests: the Mann-Whitney U test does not indicate a significant difference in central tendency ($p = 1.54e{-}01$), whereas the Kolmogorov-Smirnov test confirms a significant distributional shift ($D = 0.0959$, $p = 1.06e{-}76$), largely driven by tail behavior. Figure 10. (TOPSIS/VIKOR density plots) further illustrates this overlap-plus-tail effect, explaining why risk scores alone are insufficient as standalone decision rules. Consequently, treat TOPSIS/VIKOR as auxiliary continuous signals that complement the learned classifier rather than replacing it. This overlap indicates that the risk score alone is not sufficient as a decision rule, motivating its use as an auxiliary feature within the classifier. Consequently, treat TOPSIS as auxiliary continuous signals that complement the learned classifier rather than replacing it. Table 11 shows the risk score statistics by class.

Malware samples are more risky than benign samples with both MCDM scores (TOPSIS, VIKOR), meaning that semantic criteria based on objectives reflect risk discriminatory information. The close media implies an overlap between benign and borderline applications, which is anticipated with large-scale VirusTotal-derived labeling noise, however, the mean shift indicates that the risk indices can be useful as globally interpretable indicators.

Table 11: Risk-score statistics by class (benign vs. malware).

Label	TOPSIS mean	TOPSIS median	TOPSIS std	VIKOR mean	VIKOR median	VIKOR std
0 (Benign)	0.0323	0.0225	0.0266	0.0611	0.0456	0.0479
1 (Malware)	0.0410	0.0210	0.0529	0.0795	0.0442	0.0978

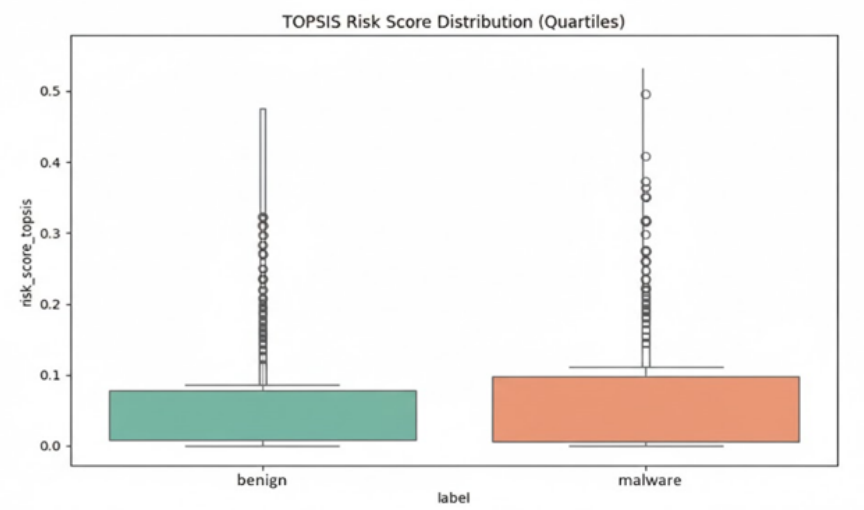

Figure 10: Correlation structure of mcdm risk scores and weighted semantic features.

4.5 Quantitative Comparison with Related Work

In We compare our best configuration (B3 + Random Forest) to the representative studies in Section 2; nevertheless, the comparison is suggestive, since the published results differ in terms of datasets, VirusTotal thresholds, balance of classes, and protocols of evaluation. Our results under the noise reduction AndroZoo labeling and stratified holdout perceptions are 83.16% Accuracy, 91.73% AUC and 83.25% F1-score. A number of previous works confirm greater headline scores in alternative contexts such as permission-based frameworks with aggressive feature removal [5], Bayesian models on AndroZoo/Drebin subsets [6], permission-pattern SVM with weaker VT rules (VT<2 vs. VTgeq2) [7], sensitivity-weighted TextCNN, [8], permission-pattern SVM-based composition schemes, [11] and time-robust static detection with time-shift testing [13]. Another design choice made in our pipeline is that of a high-confidence malware threshold (VT ≥ 10) to cut label noise. Multi-view fusion techniques like LensDroid [14] also report very high performance but use quite different representations. To justify this decision, we reduce the threshold (VT ≥ 1, 2, 5, 10) and we do not train more than eight semantic permission-count features, and the performance of the multiple classifiers saturate at approximately 73.76, which represents an accuracy ceiling at this level of abstraction. This implies that the weaker thresholds give in to noisier positives and strong feature compression may introduce an overlap between benign communication intensive applications and malware that is stealthy. So, it is not so much the maximization of accuracy based on highly abstracted features, but rather we are outputting an interpretable and consistent risk signal through a combination of objective weighting (Entropy + CRITIC) with MCDM scoring (TOPSIS/VIKOR) to give auditable and bias-free prioritization.

5 DISCUSSION

The ablation paper transfigures a distinct discrimination-interpretability trade-off of permission-based malware screening. On the stratified 80/20 holdout split, the hybrid representation (B3 + Random Forest) is best in general (83.16% Accuracy, 91.73% ROC-AUC) and it suggests that a combination of fine-grained permissions and semantic aggregation and continuous MCDM risk indicators will produce increased discrimination. The baseline of raw-permission (B0 + RF) is competitive (82.52% Accuracy, 91.41% ROC-AUC), which confirms that permissions do have a strong predictive signal and the hybrid design is providing complementary information, and enhances auditability. In the semantic track, authorizations are modeled into eight interpretable classes with objective weighting towards the System, Phone, and Contacts; RF model feature attribution is to a large extent influenced by continuous MCDM risk scores (TOPSIS/VIKOR) and by weighted category counts (e.g. Network, System, Storage), justifying the usefulness of

MCDM-rendered summaries of authorization patterns. On the whole, the findings indicate a hybrid feature to have the most accurate and best (AUC) profile, and the reduced dimensionality using compact variants might maintain much predictive potential but enhance the working efficiency. In practice, the framework is to be used on large scale triage, but not final decisions: it can put apps under further analysis (sandboxing/dynamic inspection) and provide clear explanations in terms of semantic categories and risk scores. The thresholds need to be adjusted to the cost of false positives and false negatives in operation and real deployments should be periodically recalibrated and time consciously validated to reduce VirusTotal drift and ecosystem shifts.

6 CONCLUSIONS

This paper came up with a permission-centric static malware detection pipeline, which incorporates semantic permission-category encoding, objective feature weighting, and MCDM-based risk scoring, and goes further to reinforce discrimination with a hybrid representation, retaining informative raw-permission signals. We tested several sets of features using a single labeling policy (vtdetection = 0 benign, vtdetection ≥ 10 malware; unsettled cases were excluded), VirusTotal detection-count labeling policy (vtdetection = 0 benign, vtdetection ≥ 10 malware). There are better results with B3 + Random Forest (83.16% Accuracy, 91.73% ROC-AUC on holdout), but with a much smaller number of raw features, B3-lite-100 also attains similar performance. Moreover, B2 + provides an interpretable categorical level perspective, which facilitates open reporting. Lastly, risk-score analysis indicates that TOPSIS/VIKOR indicators are best applied as auxiliary continuous variables with permission evidence to give them complementary signal to their application in addition to threshold-based decision principles.

ACKNOWLEDGEMENT

The authors would like to thank Mustansiriyah University (https://uomustansiriyah.edu.iq/) in Baghdad-Iraq for its support in the present work.

REFERENCES

[1] Y. Pan, X. Ge, C. Fang, and Y. Fan, “A Systematic Literature Review of Android Malware Detection Using Static Analysis,” IEEE Access, vol. 8, pp. 116363-116379, 2020, [Online]. Available: https://doi.org/10.1109/ACCESS.2020.3002842.

[2] A. Alzubaidi, “Recent Advances in Android Mobile Malware Detection: A Systematic Literature Review,” 2021, Institute of Electrical and Electronics Engineers Inc., [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3123187.

[3] S. F. Ali, M. R. Abdulrazzaq, and M. T. Gaata, “Learning Techniques-Based Malware Detection: A Comprehensive Review,” Mesopotamian Journal of CyberSecurity, vol. 5, no. 1, pp. 273-300, 2025.

[4] R. H. Mahdi and H. Trabelsi, “Detection of Malware by Using YARA Rules,” in 2024 21st International Multi-Conference on Systems, Signals & Devices (SSD), 2024, pp. 1-8, [Online]. Available: https://doi.org/10.1109/SSD61670.2024.10549308.

[5] T. Pathak, T. S. Kumar, and U. Barman, “Static analysis framework for permission-based dataset generation and android malware detection using machine learning,” EURASIP Journal on Information Security, vol. 2024, no. 1, p. 33, 2024.

[6] S. R. T. Mat, M. F. A. Razak, M. N. M. Kahar, J. M. Arif, and A. Firdaus, “A Bayesian probability model for Android malware detection,” ICT Express, vol. 8, no. 3, pp. 424-431, 2022, [Online]. Available: https://doi.org/10.1016/j.icte.2021.09.003.

[7] Z. Namrud, S. Kpodjedo, A. Bali, and C. Talhi, “Deep-Layer Clustering to Identify Permission Usage Patterns of Android App Categories,” IEEE Access, vol. 10, pp. 24240-24254, 2022, [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3156083.

[8] Y. Song, Y. Geng, J. Wang, S. Gao, and W. Shi, “Permission Sensitivity-Based Malicious Application Detection for Android,” Security and Communication Networks, vol. 2021, 2021, [Online]. Available: https://doi.org/10.1155/2021/6689486.

[9] K. Allix, T. F. Bissyandé, J. Klein, and Y. Le Traon, “AndroZoo: collecting millions of Android apps for the research community,” in Proceedings of the 13th International Conference on Mining Software Repositories, in MSR ’16, New York, NY, USA: Association for Computing Machinery, 2016, pp. 468-471, [Online]. Available: https://doi.org/10.1145/2901739.2903508.

[10] J. Wang, X. H. Liu, M. Huang, P. Zhou, and Y. Xu, “Android Malware Detection Method Combining Multi-Frequency Features and Convolutional Neural Networks,” IEEE Access, p. 1, 2025, [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3550124.

[11] H. Kato, T. Sasaki, and I. Sasase, “Android Malware Detection Based on Composition Ratio of Permission Pairs,” IEEE Access, vol. 9, pp. 130006-130019, 2021, [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3113711.

[12] S. Banik and J. P. Singh, "Android Malware Detection by Correlated Real Permission Couples Using FP Growth Algorithm and Neural Networks," IEEE Access, vol. 11, pp. 124996-125010, 2023, [Online]. Available: https://doi.org/10.1109/ACCESS.2023.3323845.

[13] J. Xu, Y. Li, R. H. Deng, and K. Xu, "SDAC: A Slow-Aging Solution for Android Malware Detection Using Semantic Distance Based API Clustering," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 2, pp. 1149-1163, Mar. 2022, [Online]. Available: https://doi.org/10.1109/TDSC.2020.3005088.

[14] Z. Meng et al., "Detecting Android Malware by Visualizing App Behaviors from Multiple Complementary Views," IEEE Transactions on Information Forensics and Security, p. 1, 2025, [Online]. Available: https://doi.org/10.1109/TIFS.2025.3547301.

[15] H. He and E. A. Garcia, "Learning from Imbalanced Data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, Sep. 2009, [Online]. Available: https://doi.org/10.1109/TKDE.2008.239.

[16] C. Drummond, R. C. Holte, and others, "C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in Workshop on Learning from Imbalanced Datasets II, 2003.

[17] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket," in NDSS, 2014, pp. 23-26.

[18] G. S. Tuncay, "Android permissions: Evolution, attacks, and best practices," IEEE Security & Privacy, vol. 22, no. 6, pp. 40-49, 2024.

[19] S. Martinčić-Ipšić, T. Miličić, and L. Todorovski, "The influence of feature representation of text on the performance of document classification," Applied Sciences, vol. 9, no. 4, p. 743, 2019.

[20] A. Sabbah, R. Jarrar, S. Zein, and D. Mohaisen, "Understanding Concept Drift with Deprecated Permissions in Android Malware Detection," arXiv, Jul. 2025, [Online]. Available: https://doi.org/10.48550/arXiv.2507.22231.

[21] O. A. Akanbi, I. S. Amiri, and E. Fazeldehkordi, A Machine-Learning Approach to Phishing Detection and Defense. Syngress, 2014.

[22] C. E. Shannon, "A mathematical theory of communication," The Bell System Technical Journal, vol. 27, no. 3, pp. 379-423, 1948.

[23] Y. Sharma and A. Arora, "IPAnalyzer: A Novel Android Malware Detection System Using Ranked Intents and Permissions," Multimedia Tools and Applications, vol. 83, no. 10, pp. 12345-12362, Mar. 2024, [Online]. Available: https://doi.org/10.1007/s11042-024-18511-6.

[24] A. Alomar, A. A. Aljarullah, and S. Abu-Ghazalah, "Permissions-Based Android Malware Detection Using Machine Learning," Neural Computing and Applications, vol. 37, no. 6, pp. 5255-5270, Dec. 2024, [Online]. Available: https://doi.org/10.1007/s00521-024-10950-4.

[25] Y. Sharma and A. Arora, "A Comprehensive Review on Permissions-Based Android Malware Detection," International Journal of Information Security, vol. 23, no. 3, pp. 1877-1912, Mar. 2024, [Online]. Available: https://doi.org/10.1007/s10207-024-00822-2.

[26] S. Jogsan, "A Survey on Permission-Based Malware Detection in Android Applications," International Journal of Engineering Research and Technology, vol. 9, no. 4, pp. 774-778, May 2020, [Online]. Available: https://doi.org/10.17577/IJERTV9IS040774.