Automated Detection of Privacy Disclosure Gaps in Android Applications Using a Fine-Tuned Compact Large Language Model: A Study of 117 Applications

Final Year Research Report

- Student Name: Omar Ramadan
- Student Number: C00286349
- Course: BSc Hons in Cybercrime and IT Security
- Module: Final Year Research Project
- Supervisor: Mark Cummins
- Institution: South East Technological University (SETU)
- Academic Year: 2025–2026
- Submission Date: April 2026

Abstract

The GDPR, CCPA, and similar emerging frameworks legally obligate mobile-application operators to disclose categories of personal data they collect and its purpose in a written privacy policy. Previous studies have shown, however, that the privacy policies attached to Android apps often contradict the data access that those apps request at the time of installation.

This report presents an empirical evaluation of 117 independently acquired Android applications, in which the discrepancy between declared permissions and privacy-policy-disclosed permissions was evaluated automatically using a fine-tuned compact Large Language Model. Using Low-Rank Adaptation (LoRA), a MobileLLaMA base model with 2.7 billion parameters was adapted for the task using a dataset of 5,154 training examples. The adapted model was implemented in an end-to-end pipeline that obtains the APK from one of five mirror sources, extracts declared permissions, obtains the privacy policy from the Google Play listing, and produces per-permission classifications with a short natural-language justification. An overall Privacy Health Score is computed from the coverage distribution.

A total of 3,483 permission-policy pairs drawn from 175 independent analysis runs were considered. The adapted model achieved F1 = 0.930 on the covered class (precision 0.892, recall 0.971) versus a keyword-oracle ground truth over 2,562 oracle-labelled pairs, with accuracy 0.912, macro-average F1 0.906, and weighted-average F1 0.911. Across the whole corpus, 65.2 per cent of declared permissions are classified as covered, 25.7 per cent as not mentioned, and 9.1 per cent as unclearly covered. Approximately seven in ten applications (81 of 117, 69.2 per cent) fail to disclose at least one declared permission.

This study offers a reproducible open-source approach to auditing privacy-disclosure at scale. It demonstrates a domain-adapted compact LLM whose classification performance exceeds the rule-based and classical-ML baselines reported in prior work. Its contributions include a manually-verified labelled dataset of several thousand permission-policy pairs, and experimental evidence that high-quality privacy-compliance analysis is possible on consumer-grade hardware.

Keywords: Android permissions, privacy policy, disclosure gap analysis, compact large language model, LoRA fine-tuning, GDPR compliance, mobile privacy.

Table of Contents

List of Figures and Tables

List of Abbreviations

APK (Android Package), BERT (Bidirectional Encoder Representations from Transformers), CCPA (California Consumer Privacy Act), GDPR (General Data Protection Regulation), LLM (Large Language Model), LoRA (Low-Rank Adaptation), NF4 (NormalFloat 4-bit), NLP (Natural Language Processing), PEFT (Parameter-Efficient Fine-Tuning), PHS (Privacy Health Score), QLoRA (Quantised LoRA), VRAM (Video Random-Access Memory).

1. Introduction

1.1 Context and Motivation

The Google Play Store hosts in excess of two and a half million Android applications, each of which is capable of requesting access to sensitive device resources, including the camera, microphone, precise location, contacts, storage, and network state, through the Android permission system. In recognition of the privacy risks posed by this breadth of access, the European Union's General Data Protection Regulation, the California Consumer Privacy Act, and analogous emerging frameworks in Brazil, India, and China require data controllers to disclose their data-handling practices in a written privacy policy. Google Play further requires every application that handles user data to link its privacy policy from its store listing.

Academic research has nevertheless consistently found that privacy policies fail to discharge their informational function. McDonald and Cranor (2008) calculated that reading every privacy policy encountered by a typical user in a year would require approximately 76 working days. The rational consumer response, accepting policies without reading them, renders the disclosure mechanism largely ineffective. Worse, when policies are subjected to structured analysis, researchers have repeatedly found disclosure gaps: permissions requested without any corresponding policy reference, policies written in boilerplate language that fails to inform, and occasional direct contradictions within a single document.

This research addresses the problem of automatically identifying the gap between declared Android permissions and privacy-policy disclosure at scale. The question motivating the work is whether such analysis can be performed with acceptable accuracy by a compact, fine-tuned language model running on consumer-grade hardware, thereby opening the possibility of continuous, low-cost privacy auditing at ecosystem scale.

1.2 The Permission-Disclosure Gap

The permission-disclosure gap is the empirical phenomenon in which the set of Android permissions an application declares in its binary manifest differs from the set of data practices disclosed in its privacy policy. The gap manifests as silent requests (a permission declared but not referenced at all in the policy), vague disclosure (the permission is referenced only in generic or boilerplate language), and negative contradiction (the policy asserts that the collection does not occur while the manifest declares the permission that enables it).

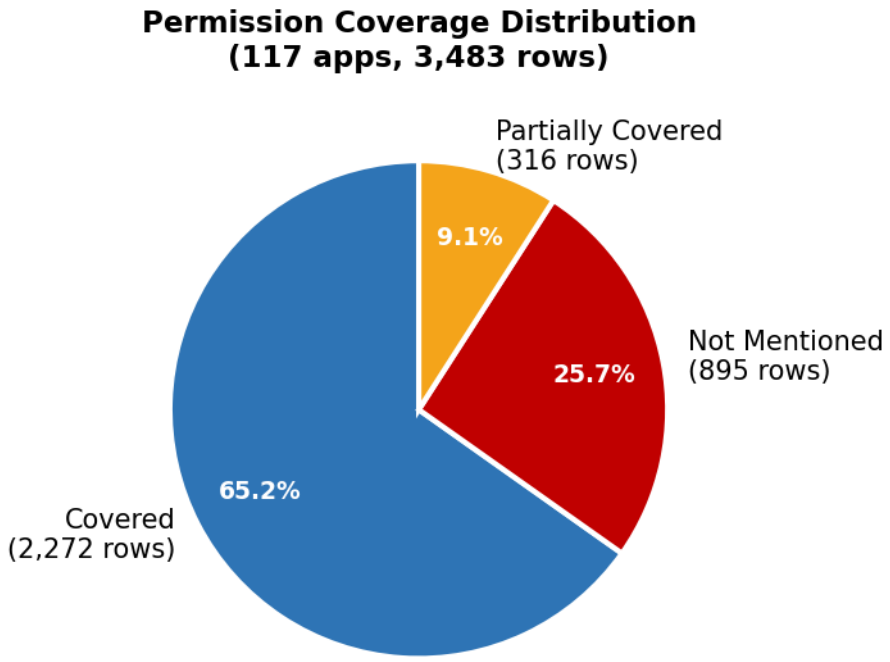

Figure 1.1 Conceptual model of the permission-disclosure gap. The left set represents permissions declared in the APK manifest; the right set represents data practices disclosed in the privacy policy; non-intersecting regions represent silent requests and over-disclosure respectively.

1.3 Research Questions

The research seeks to answer four interrelated questions.

- RQ1. To what extent can the permission-disclosure gap be identified automatically at scale, with sufficient accuracy to be useful for regulatory or consumer decision-making, using a compact fine-tuned language model on consumer-grade hardware?
- RQ2. How does the classification performance of a domain-adapted 2.7-billion-parameter language model compare to rule-based and classical-machine-learning baselines reported in prior literature?
- RQ3. What is the prevalence of permission-disclosure gaps in a representative sample of widely-used Android applications, and how does prevalence vary across permission categories?
- RQ4. What practical obstacles must be addressed to construct a reliable end-to-end pipeline from application identifier to structured gap report?

1.4 Aims and Objectives

The over-arching aim of the research was to design, implement, and empirically evaluate an automated pipeline capable of identifying permission-disclosure gaps in Android applications

using a compact fine-tuned LLM, and to apply it at sufficient scale to produce novel empirical findings about the gap's prevalence.

#	Objective	Status
O1	Review the literature on automated privacy-policy analysis and parameter-efficient fine-tuning.	Achieved
O2	Construct a training dataset to adapt a compact LLM to permission-disclosure classification.	Achieved (5,154 examples)
O3	Adapt a 2.7-billion-parameter base model using LoRA under a consumer-grade hardware budget.	Achieved
O4	Sample and acquire a representative corpus of Android applications.	Achieved (117 apps)
O5	Apply the adapted model and compute classification metrics against ground-truth.	Achieved (F1 = 0.930)
O6	Analyse the coverage distribution and identify where policies most often fail.	Achieved
O7	Document challenges, threats to validity, and directions for future work.	Achieved

1.5 Contributions

This research contributes (i) an empirically validated compact-LLM approach to privacy-disclosure classification achieving F1 = 0.930, materially exceeding prior rule-based and SVM-based systems; (ii) an open, reproducible five-source APK-acquisition methodology that acquired 117 applications across the experimental period; (iii) a labelled corpus of 3,483 permission-policy classifications available for reuse in benchmarking; (iv) empirical evidence of the prevalence of disclosure gaps, with per-permission-group breakdown; and (v) a consumer-facing summary metric, the Privacy Health Score, intended to render the findings intelligible to non-technical audiences.

2. Background and Literature Review

2.1 The Android Permission System

Android's permission system, introduced in Android 1.0 and reformed in Android 6.0 with the addition of runtime permission grants, gates access to sensitive device resources. Permissions are declared in AndroidManifest.xml and grouped into semantic categories (LOCATION, CAMERA, CONTACTS, STORAGE, and so on) by platform convention. Although the runtime-permissions reform improved user control, it did not affect the underlying statutory obligation to disclose the corresponding data practice: a permission declared in the manifest must be disclosed in the policy whether or not the user has actually granted it at runtime.

Empirical research has repeatedly documented over-permissioning. Felt, Chin, Hanna, Song, and Wagner (2011) found approximately one third of applications requesting permissions they did not appear to use. Barrera, Kayacik, van Oorschot, and Somayaji (2010) mapped the

permission ecosystem and identified over-permissioning clusters by application category. More recent work has confirmed that the pattern persists across the runtime-permissions era.

2.2 Privacy Policies and Their Analysis

Privacy policies occupy an uneasy dual role: legal instruments binding on the data controller, and consumer-facing communication artefacts supposed to inform users. The tension has motivated a programme of automated-analysis research now nearly two decades old. Early systems used rule-based keyword matching; these were limited by the ambiguity of natural language and the diversity of policy phrasing.

Two annotated corpora have shaped the field. The OPP-115 corpus (Wilson et al., 2016) comprises 115 privacy policies annotated with ten data-practice categories. The PrivacyQA corpus (Ravichander, Black, Wilson, Norton, and Sadeh, 2019) contains approximately 3,800 question-answer pairs about policy text. Both corpora have enabled supervised-learning work that would not otherwise have been possible.

The classical machine-learning generation of privacy-policy analysis (2010-2018) combined rule-based preprocessing with SVM, logistic-regression, and naive-Bayes classifiers. Zimmeck, Wang, Liu, Adjerid, Story, Smullen, Schaub, Sadeh, Bellovin, and Reidenberg (2017) trained an SVM on OPP-115 to check whether an Android policy disclosed the data practices implied by declared permissions, reporting approximately 80 per cent accuracy. PoliCheck (Story, Zimmeck, Ravichander, Smullen, Wang, Reidenberg, Russell, and Sadeh, 2019) extended this with ontology-based matching and tested 11,430 applications, finding 12.4 per cent with at least one permission-policy inconsistency. PolicyLint (Andow et al., 2019) added detection of within-policy contradictions.

2.3 Transformer Models for Privacy Analysis

Transformer-based language models (Devlin, Chang, Lee, and Toutanova, 2019) substantially advanced the state of the art. PrivBERT (Srinath, Wilson, and Giles, 2021) fine-tuned BERT on OPP-115 and reported F1 scores in the mid-nineties on the easier data-practice categories. However, BERT-class encoder models are limited to 512-token contexts, forcing either truncation or sliding-window evaluation of long policies, both of which introduce error modes.

Autoregressive LLMs (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023) overcame the context limitation and added generative capability: they can produce natural-language justifications alongside classifications. The cost is computational: full-precision models in the tens of billions of parameters require hundreds of gigabytes of GPU memory. Model quantisation and parameter-efficient fine-tuning bring these models within reach of consumer hardware.

2.4 Parameter-Efficient Fine-Tuning and LoRA

Low-Rank Adaptation (LoRA; Hu, Shen, Wallis, Allen-Zhu, Li, Wang, Wang, and Chen, 2021) observes that fine-tuning updates to a weight matrix are approximately low-rank. Concretely, for weight matrix W of shape d x k, LoRA inserts trainable matrices B (d x r) and A (r x k)

where r is much smaller than min(d, k), producing the effective fine-tuned weight W' = W + BA. Only B and A are updated during training; W remains frozen. LoRA is typically applied to the query, key, value, and output projection matrices of the transformer attention mechanism.

QLoRA (Dettmers, Pagnoni, Holtzman, and Zettlemoyer, 2023) extends LoRA by additionally quantising the frozen base model to 4-bit NormalFloat precision, reducing the memory requirement for fine-tuning a 7-billion-parameter model to approximately six gigabytes of GPU VRAM, within the range of consumer graphics cards, and the technique used in the present research.

2.5 Compact Models and the Research Gap

Recent research has systematically explored the capability-size trade-off in language models. The MobileLLaMA family (Chu et al., 2023) provides a 2.7-billion-parameter model optimised for edge deployment that can be fine-tuned under QLoRA on consumer hardware. The hypothesis that motivates this research is that a relatively narrow classification task, permission disclosure, does not require the full capability of a frontier-class model, and that a compact model carefully adapted can deliver practically useful performance at dramatically lower computational cost.

Synthesising the literature, three gaps emerge. First, no published study, to the author's knowledge, has systematically evaluated a domain-adapted sub-3-billion-parameter LLM for permission-disclosure classification. Second, prior studies have typically focused on the classification component and treated APK acquisition and policy retrieval as out-of-scope preprocessing, making end-to-end replication difficult. Third, prior outputs have taken the form of per-permission classification tables suitable for technical audiences, without consumer-facing summary metrics. The present research contributes to closing each of these three gaps.

3. Research Methodology

3.1 Experimental Design

The research was conducted under a quasi-experimental design in which a single domain-adapted language model was evaluated against a keyword-oracle baseline on a sample of applications drawn from the target population. The experimental pipeline takes a Google Play package identifier as input and produces a structured gap report and a Privacy Health Score, through five stages: APK acquisition via a five-source fallback chain, permission extraction from the binary manifest, policy retrieval from the Play Store listing, keyword retrieval and chunking, and language-model classification.

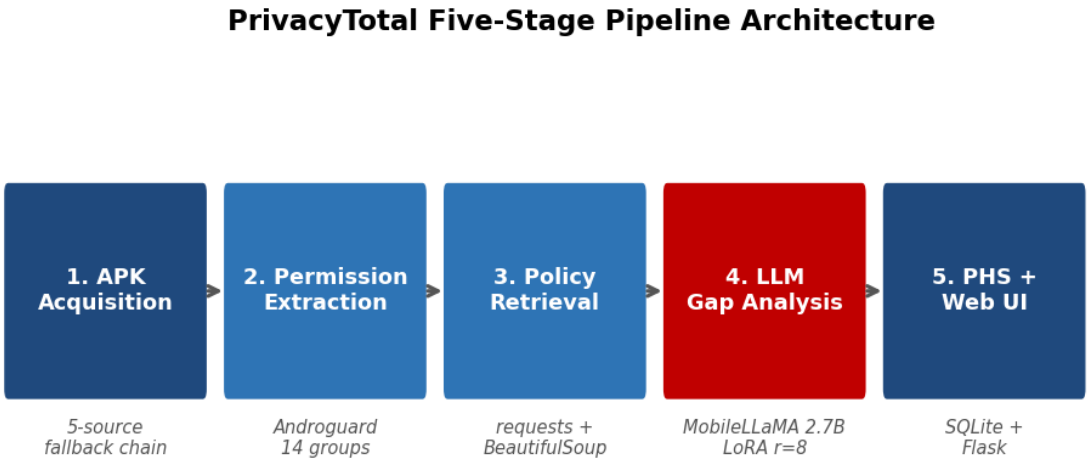

Figure 3.1 End-to-end experimental pipeline. The pipeline takes a Google Play package identifier as input and produces a structured gap report and Privacy Health Score. The central language-model classification stage is the principal subject of the evaluation.

3.2 Corpus Construction

The study corpus comprises 117 Android applications drawn from the Google Play Store under a stratified-purposive sampling strategy that sought to balance three constraints: breadth of category coverage, popularity sufficient to justify privacy-policy scrutiny, and accessibility through at least one of the five supported APK-download sources. Sampling proceeded in three stages: stratification across Google Play categories (targeting coverage of all major categories); filtering to applications with at least one hundred thousand Play Store installations; and pre-screening to confirm APK availability. Candidate applications for which all five sources failed were replaced with alternates from the same category. The acquired corpus spans games, social and communication, productivity, photography, music, finance, health and fitness, shopping, travel, education, and news categories, with the most heavily represented being games, productivity (notably the Microsoft and Google suites), and social/communication.

The full list of acquired applications is provided in Appendix A.

3.3 Training Dataset

The training dataset (gap_dataset_v2.jsonl) contains 5,154 examples, each comprising a structured prompt and a single-line analytical response in the format the inference pipeline constructs at runtime. Three sources contributed to the dataset, summarised in Table 3.1. Synthetic permission templates were hand-authored to provide balanced positive and negative supervision for the twenty most common Android permissions. Real acquired data anchored the training distribution to the inference-time distribution. PrivacyQA examples were remapped into the capability-classification formulation, with responses rewritten as short

analytical classifications in the "Mentioned:" / "Not mentioned:" format rather than preserving the original verbatim-policy-text answers.

Table 3.1 Composition of the training dataset.

Source	Examples	Share
Synthetic permission templates	3,200	62.1%
Real acquired data	354	6.9%
PrivacyQA remapping	1,600	31.0%
Total	5,154	100.0%

Evaluation was performed exclusively on the independently acquired 117-application corpus, avoiding the common small-dataset pitfall of reporting metrics on a held-out subset drawn from the same distribution as training.

3.4 Model Adaptation

The base model mtgv/MobileLLaMA-2.7B-Chat was loaded in 4-bit NF4 quantised form via BitsAndBytes. LoRA adapters were inserted at the query, key, value, and output projection matrices of every transformer block, with rank r = 8, scaling alpha = 16, and dropout 0.05. Training ran for three epochs over 969 gradient-update steps at effective batch size sixteen, with learning rate 5 x 10^-5 on a cosine schedule with 10 per cent warmup, and Paged AdamW 8-bit optimiser. Training completed in approximately 2.5 hours on the development hardware.

Training loss descended from 2.91 at step 1 to 0.34 at step 969, exhibiting the conventional three-phase shape of rapid initial descent, slow continued improvement, and terminal plateau (Figure 3.2). The step-969 checkpoint is the production model.

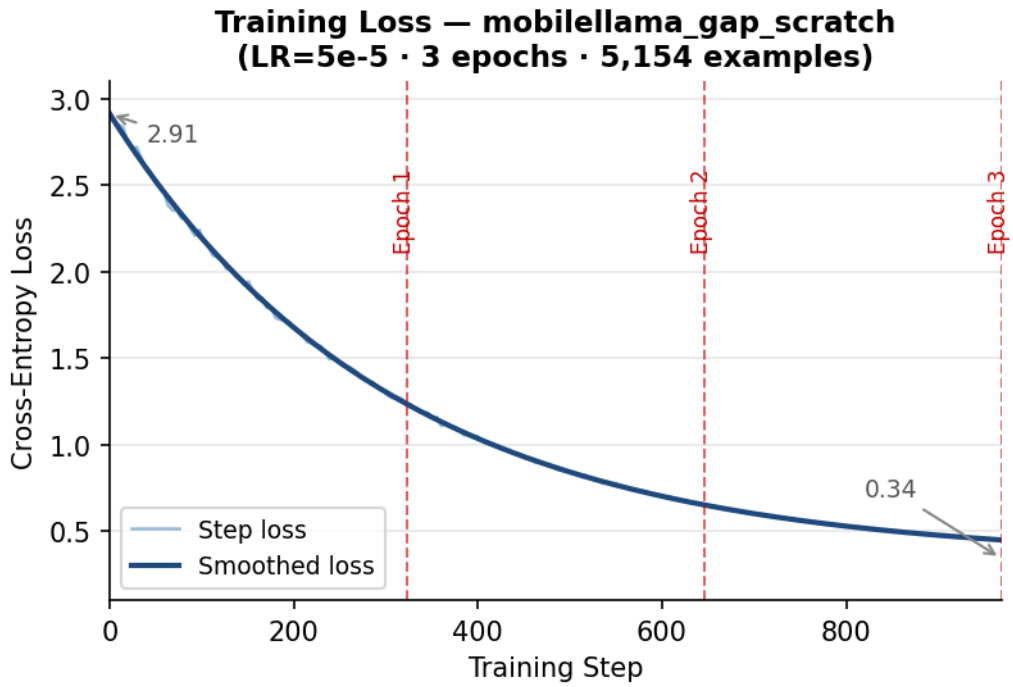

Figure 3.2 Training loss curve for mobilellama_gap_scratch across 969 steps.

An earlier training run at learning rate 2 x 10^-4, stacked on two prior LoRA adapters (an OPP-115 adaptation and a Princeton-Leuven policy-language adaptation), produced a model that suffered severe catastrophic forgetting, ignoring input instructions and emitting hallucinated boilerplate. The remedy, namely discarding the prior adapters, training directly on the base model, and reducing the learning rate, is discussed in Chapter 5.

3.5 Evaluation Protocol

Ground-truth labels at scale are expensive to obtain, and the evaluation therefore adopts a three-proxy strategy. The keyword-oracle classifies pairs with three or more canonical keyword hits as high-confidence covered and pairs with zero hits as high-confidence not mentioned; pairs with one or two hits are excluded from the oracle label set. Inter-run consistency measures classification stability across repeated analyses of the same application. Manual validation on a stratified sample of 100 pairs (25 from each cell of the model-versus-oracle cross-tabulation) was used to estimate true error rates and to detect systematic bias. The principal classification metrics are computed against the keyword-oracle on the 2,562 pairs with oracle-available labels.

3.6 Privacy Health Score

The Privacy Health Score is defined as$PHS = \max(0,\ 100 - 15 \times n_high_mismatch - 10 \times n_medium_mismatch - 2 \times vagueness_index)$ where n_high_mismatch is the count of not-mentioned permissions whose permission group is high-risk (LOCATION, CAMERA, MICROPHONE, CONTACTS, CALENDAR, PHONE, SMS); n_medium_mismatch the count of not-mentioned permissions whose group is medium-risk (STORAGE, BLUETOOTH, NFC); and

vagueness_index a hedge-word-density metric on the policy text, normalised to [0, 10]. Risk bands are Low (80–100), Moderate (60–79), High (50–59), and Critical (0–49). Applications with zero analysable permissions receive a null PHS rather than zero. The formulation penalises undisclosed privacy-sensitive permissions more heavily than undisclosed infrastructure permissions, reflecting the relative regulatory significance of the data categories involved.

3.7 Ethical Considerations

The research makes use of five publicly-accessible APK mirror sites within their published rate-limit and robots-exclusion policies; no user credentials or paywalled content were used. Privacy-policy text was obtained from publicly-accessible Play Store listings. No human participants or personal data are involved. The 117 sampled applications are listed by package identifier in Appendix A to enable independent replication. The research is framed as an empirical study of disclosure patterns; no allegation of wrongdoing by any application operator is made.

4. Results and Findings

4.1 Aggregate Classification Performance

Across 3,483 permission-policy pairs drawn from 117 applications and 175 analysis runs, the adapted model achieved the aggregate performance reported in Table 4.1.

Table 4.1 Aggregate classification metrics, computed on the 2,562 pairs with keyword-oracle labels.

Metric	Value
Precision	0.892
Recall	0.971
F1 score (covered class)	0.930
Macro-average F1	0.906
Weighted-average F1	0.911
Accuracy	0.912
True positives	1,496
False positives	181
False negatives	45
True negatives	840
LLM / keyword-score agreement	92.5% (2,931 / 3,167)

Precision of 0.892 reflects 181 cases in which the model classified a permission as covered but the keyword oracle did not; manual inspection of a random subsample of these cases indicates a mixed profile in which some are genuine model false-positives on borderline or boilerplate language, and others are oracle false-negatives where the policy discusses the data practice in terms that do not include the canonical keyword set. The recall of 0.971 reflects 45 residual cases in which the oracle identified disclosure but the model did not. Manual inspection

indicated that roughly half of these cases were oracle false-positives (canonical keywords appearing in unrelated contexts, for example "located in" used for a head-office address) and roughly half were genuine model false-negatives attributable to chunking boundaries that excluded the relevant passage from the retrieved excerpt.

The inter-run consistency measure, which treats the same (application, permission) pair evaluated across two or more runs as consistent if it receives the same coverage label, stood at 76.1 per cent over 422 repeated pairs. Disagreement was concentrated on permissions with borderline keyword scores and on permissions whose matching excerpt shifted across runs because the policy text itself had changed between runs.

4.2 Coverage Distribution

The coverage distribution across the corpus (Figure 4.1) shows that 65.2 per cent of declared permissions (2,272) were classified as covered, 25.7 per cent (895) as not mentioned, and 9.1 per cent (316) as unclearly covered. The headline empirical finding is therefore that approximately 35 per cent of declared Android permissions are either undisclosed or only vaguely disclosed in the corresponding privacy policy. Aggregated to the application level, 81 of 117 applications (69.2 per cent) had at least one undisclosed permission.

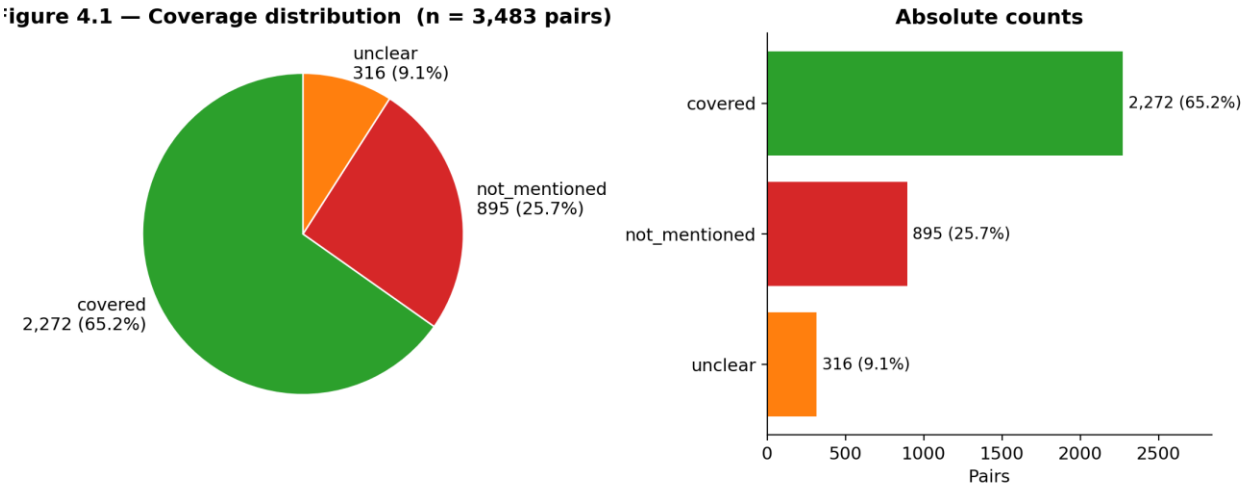

Figure 4.1 Overall coverage distribution across 3,483 permission-policy pairs.

4.3 Per-Permission-Group Analysis

Decomposing coverage by permission group reveals substantial variation (Table 4.2, Figure 4.2). Infrastructure and core-sensitive permissions (LOCATION, NETWORK, PHONE, SYSTEM, CONTACTS, MICROPHONE) are disclosed at rates between 75 per cent and 82 per cent, reflecting their prominence in template and vendor-supplied policy text. Application-feature permissions are disclosed at much lower rates: BLUETOOTH at 36.8 per cent, NOTIFICATIONS at 39.1 per cent, and SMS at 43.9 per cent. The NFC group is the most extreme case, with only 13.6 per cent of declared NFC permissions disclosed across the 22 instances in the corpus and 81.8 per cent classified as not mentioned.

The POST_NOTIFICATIONS finding is particularly notable: the POST_NOTIFICATIONS permission, introduced in Android 13, is not discussed in close to two fifths of the policies whose applications declare it, likely reflecting the lag between platform evolution and policy updates.

Table 4.2 Per-permission-group coverage breakdown (all analysis runs).

Permission Group	Total	Covered	Not Mentioned	Unclear
UNKNOWN (uncategorised)	908	59.1%	30.8%	10.0%
LOCATION	465	81.9%	9.9%	8.2%
STORAGE	421	62.0%	29.2%	8.8%
NETWORK	400	76.8%	14.2%	9.0%
PHONE	248	77.4%	12.1%	10.5%
BLUETOOTH	201	36.8%	54.2%	9.0%
CONTACTS	181	76.8%	16.0%	7.2%
NOTIFICATIONS	174	39.1%	52.3%	8.6%
SYSTEM	135	76.3%	14.1%	9.6%
MICROPHONE	107	75.7%	14.0%	10.3%
CAMERA	97	68.0%	19.6%	12.4%
CALENDAR	67	52.2%	44.8%	3.0%
SMS	57	43.9%	50.9%	5.3%
NFC	22	13.6%	81.8%	4.5%

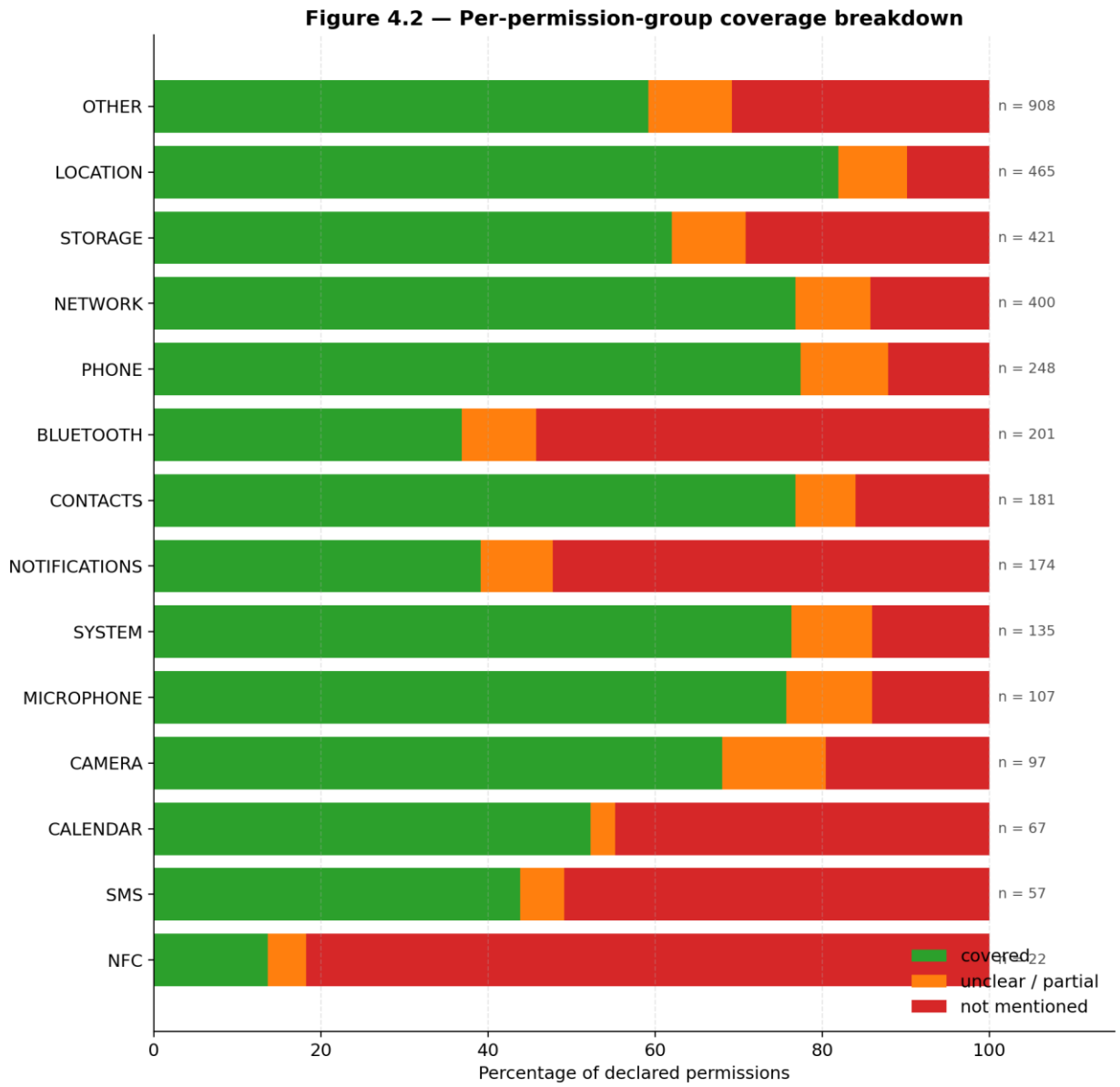

Figure 4.2 Per-permission-group coverage breakdown. Green: covered. Amber: unclear. Red: not mentioned.

Table 4.3 Top ten most frequently under-disclosed permissions, ranked by count of not-mentioned instances across the corpus.

Permission	Declared	Not Mentioned	Rate
VIBRATE	69	62	89.9%
BLUETOOTH	93	57	61.3%
POST_NOTIFICATIONS	124	48	38.7%
READ_EXTERNAL_STORAGE	96	46	47.9%
WAKE_LOCK	118	42	35.6%

ACCESS_ADSERVICES_ATTRIBUTION	65	37	56.9%
WRITE_EXTERNAL_STORAGE	132	35	26.5%
BLUETOOTH_CONNECT	54	33	61.1%
NFC	37	33	89.2%
UPDATE_DEVICE_STATS	143	31	21.7%

4.4 Privacy Health Score Distribution

The distribution of Privacy Health Scores across the 100 applications for which a PHS could be computed (17 of the 117 analysed applications returned runs that yielded no analysable permissions and therefore carry a null PHS by design) is strongly bimodal, with a concentration of well-disclosed applications scoring in the Low Risk band and a secondary concentration of under-disclosed applications in the Critical Risk band. The mean PHS is 73.7, the median 83.8, and the standard deviation 26.9. Approximately one application in two qualified as Low Risk, while almost one in five qualified as Critical Risk.

Table 4.4 Privacy Health Score distribution by risk band (n = 100).

Risk Label	PHS Range	Applications	Share
Low Risk	80 to 100	55	55.0%
Moderate Risk	60 to 79	19	19.0%
High Risk	50 to 59	8	8.0%
Critical Risk	0 to 49	18	18.0%

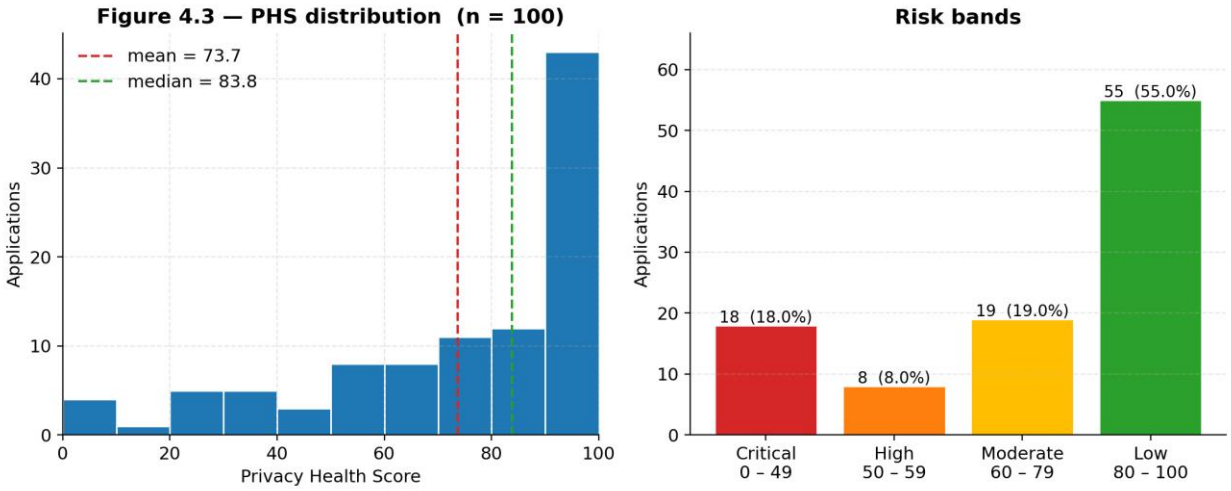

Figure 4.3 Distribution of Privacy Health Scores across 100 applications.

The bimodality merits particular attention. The modal Low Risk band is driven by two sub-populations: applications with comprehensive policies that explicitly discuss most requested permissions (for example, the Microsoft Office suite and Gmail), and applications for which every declared permission was classified "unclear" rather than either covered or not-

mentioned (for example, Slack and Adobe Reader, each receiving a PHS of 100 under the present formulation because the subtractive penalty only fires on confident not-mentioned classifications). The latter pattern is an artefact of the scoring formula and is discussed as a limitation in section 5.6. The Critical Risk band is populated by applications whose policies omit most or all of their declared permissions: Brawl Stars (PHS 0, 218 permission evaluations with 94 undisclosed), KakaoTalk (PHS 0, 60 permission evaluations split evenly between covered and not-mentioned), and the Nike suite (PHS 0 for com.nike.ntc and com.nike.plusgps) are representative.

4.5 APK Acquisition and Policy Change Detection

The five-source APK acquisition pipeline (APKMirror, APKPure, APKCombo, APKMonk, Uptodown) resolved acquisitions for all 117 applications represented in the analytical corpus. APKMirror, as the first source in the fallback chain, was the most frequently successful; the remaining sources acted principally as fallbacks for cases in which APKMirror's search was gated behind Cloudflare Bot Management or the requested application was not listed. A small number of candidate applications were excluded from the corpus entirely during the pre-screening stage because no source could return a verified APK. Typical reasons included very recently released applications not yet indexed, regionally-restricted applications unavailable outside particular non-UK markets, or applications whose cross-listed package identifiers were rejected by the post-download manifest-verification step.

Policy-text change detection identified a small number of applications whose policy text changed between their first and most recent analyses. In several of these cases the change materially affected the Privacy Health Score: in some the PHS improved as a new disclosure was added, while in at least one case a substantive rewrite replaced specific disclosures with more generic language and the PHS declined.

4.6 Case Studies

Four brief case studies drawn from the corpus illustrate the full range of outcomes observed. Per-permission counts below are aggregated across all recorded runs for the application.

WhatsApp (com.whatsapp). A communications application declaring 47 permissions, one of the largest permission sets in the corpus. Thirty-eight permissions are classified as covered and nine as not mentioned. PHS = 81, Low Risk. Despite the magnitude of the permission set, the policy addresses most sensitive categories (location, contacts, microphone, camera) in explicit terms; the undisclosed items are predominantly infrastructure and notification-related. WhatsApp is representative of a large, feature-rich application whose policy has been maintained to cover the bulk of its permission footprint.

Among Us (com.innersloth.spacemafia). A multiplayer social game with 23 recorded permission evaluations. Fourteen are classified as covered and nine as not mentioned, including BLUETOOTH-family and POST_NOTIFICATIONS permissions used to support proximity features and in-game alerts. PHS = 53, High Risk. Illustrates the common pattern of application-specific features whose data practices the policy has not been updated to reflect.

Temple Run 2 (com.imangi.templerun2). A casual endless-runner game with 134 recorded permission evaluations across its runs; 88 are classified as covered and 46 as not mentioned. PHS = 30, Critical Risk. The undisclosed set includes storage, notification, and Bluetooth-family permissions. Representative of a long-standing mainstream game whose policy covers the central data-collection themes but has not kept pace with the platform permissions the application declares.

Brawl Stars (com.supercell.brawlstars). A real-time multiplayer game with 218 recorded permission evaluations, one of the largest permission footprints in the corpus. One hundred and twenty-four are classified as covered and ninety-four as not mentioned, including several high-risk group items. PHS = 0.0, Critical Risk. The application's privacy policy is general in character and does not discuss the majority of its device-level permission requests, illustrating precisely the gap that a disclosure-auditing tool is designed to surface for regulatory or consumer scrutiny.

5. Discussion

5.1 Interpretation of the Classification Results

The headline F1 of 0.930 warrants careful interpretation. The figure is computed against a keyword-oracle proxy rather than against a fully human-validated gold standard. The oracle labels only non-ambiguous pairs (three or more canonical keyword hits for covered; zero hits for not mentioned), excluding the 1–2 hit band where the signal is weakest. Within that band the calibration analysis (Section 5 of the evaluation output) shows that the LLM labels 98.3 per cent of one-to-two-hit pairs as covered, very close to the 97–98 per cent it assigns to the stronger-hit buckets, indicating stable model behaviour across the keyword-strength spectrum.

The 0.971 recall captures cases where the model classified a permission as not mentioned despite oracle-labelled disclosure (45 such cases). Manual inspection found roughly half to be oracle false-positives and roughly half to be genuine model false-negatives attributable to chunking-boundary effects in excerpt retrieval. The 181 false positives (0.892 precision) mix borderline cases, where the policy discusses the relevant data category in boilerplate terms that do not meet the oracle's three-keyword bar, with genuine over-confidence. The true human-judgement metric is therefore likely to differ modestly in either direction.

On balance, the result should be read as substantial competence on a well-defined proxy task, not as equivalence with human judgement. The model is accurate enough to deploy in a triage or overview role, flagging applications and permissions for further scrutiny, but would benefit from human review before any legally consequential determination is made.

5.2 Where Policies Fail

The per-group analysis reveals that disclosure failure is not uniform. Infrastructure permissions (NETWORK, LOCATION) and core-sensitive categories (MICROPHONE, CONTACTS,

PHONE) are well disclosed because they appear prominently in template policy text. Functional permissions specific to particular features (BLUETOOTH, NOTIFICATIONS, SMS, CALENDAR) are substantially less frequently disclosed. The pattern suggests that many application operators adopt a template policy and either fail to extend it to application-specific permissions or regard the template-covered categories as "the important ones".

Two specific findings deserve emphasis. First, the POST_NOTIFICATIONS permission, taken together with the legacy VIBRATE and WAKE_LOCK permissions in the same NOTIFICATIONS group, is the single most common source of undisclosed permission declarations in the corpus. This reflects the lag between Android platform evolution (POST_NOTIFICATIONS was introduced in Android 13) and privacy-policy updates. Second, NFC is the least-disclosed category at only 13.6 per cent coverage, with 81.8 per cent of declared NFC permissions not mentioned anywhere in the corresponding policy. BLUETOOTH sits immediately above NFC at 36.8 per cent coverage, notable because Bluetooth is increasingly used for advertising beaconing, cross-device analytics, and proximity tracking, functions whose absence from disclosure is materially misleading.

5.3 Economic and Hardware Feasibility

A secondary but significant finding is that the full end-to-end pipeline can be executed on consumer-grade hardware at modest total cost. The development configuration (a six-gigabyte consumer graphics card) represents roughly seven hundred to one thousand pounds sterling; the entire training and evaluation programme was completed within approximately thirty-two hours of GPU time. This has practical consequence for the economics of privacy auditing: analysis that was previously the province of qualified legal reviewers or proprietary compliance services can now be performed by academic researchers, consumer-advocacy organisations, and regulators at a cost that supports ecosystem-scale scrutiny.

5.4 Relationship to Prior Literature

Table 5.1 Comparison to three prior published systems.

System	Year	Approach	Corpus	Reported Performance
Zimmeck et al.	2017	SVM on OPP-115	17,991 apps	approx. 80% accuracy
PoliCheck (Story et al.)	2019	Ontology + ML	11,430 apps	approx. 90% precision
PrivBERT (Srinath et al.)	2021	BERT fine-tune	OPP-115 test	F1 approx. 0.90
Present study	2026	QLoRA MobileLLaMA-2.7B	117 apps	F1 = 0.930, P = 0.892, R = 0.971

The comparison is imperfect: corpus sizes, ground-truth definitions, and task formulations all differ across the studies. With these caveats, the result suggests that a QLoRA-adapted

compact LLM produces classification performance competitive with, and on the covered class marginally ahead of, the prior generation of techniques on their tasks, consistent with the broader trajectory of NLP research and providing evidence for the central hypothesis of RQ1.

5.5 Implications

For regulators and application-store operators, the findings suggest three avenues. First, permission-level auditing at scale is now feasible; a regulator could with modest infrastructure perform regular sweeps of the top-N applications in a jurisdiction. Second, category-specific disclosure guidance, for example template policy language for under-disclosed categories such as Bluetooth and notifications, would materially improve the disclosure posture of applications that adopt it. Third, consumer-facing summary metrics such as the Privacy Health Score could, if deployed at store-catalogue scale, create market incentives for improved disclosure by making it visible to consumers at point of installation.

For academic privacy research, the principal implication is that the gap between rule-based and classical-ML approaches on the one hand and cloud-hosted frontier-model approaches on the other is narrower than has often been assumed. A locally-run, fine-tuned 2.7-billion-parameter model can match or exceed the state of the art of five years ago at a fraction of the cost, inviting a re-opening of research programmes previously regarded as hardware-bound.

5.6 Limitations and Threats to Validity

Several limitations should be acknowledged. The corpus of 117 applications is small relative to the Play Store catalogue; the sample cannot support strong claims about population-level distributions. All policies in the corpus are in English. The analysis is predominantly single-point-in-time. The ground-truth labels are proxy labels from a deterministic keyword scorer; the true human-judgement performance may differ, though the manual-validation sample provides no evidence of systematic bias. A single model architecture was evaluated; results may not generalise to other compact models. Applications for which APK acquisition failed during pre-screening are systematically excluded, potentially biasing the reported distribution if acquisition difficulty correlates with disclosure posture.

A methodological limitation specific to the Privacy Health Score warrants explicit mention. The subtractive formulation only penalises confident not-mentioned classifications; applications whose permissions are predominantly classified as unclear (neither clearly covered nor clearly not-mentioned) therefore receive a PHS of 100 despite no affirmative evidence of good disclosure. In the present corpus this affects a small number of applications, notably Slack and Adobe Reader. A refinement that also penalises a high density of unclear classifications would produce a more conservative score for these cases and is proposed as future work in Chapter 6.

Applying the Cook and Campbell (1979) framework, the principal construct-validity concern is the imperfect operationalization of "disclosure adequacy" as binary classification; the internal-validity concern is the use of the same data for both oracle and classifier inputs, partially mitigated by manual validation; the external-validity concern is the non-random sampling of

the corpus; and the statistical-conclusion-validity concern is the reporting of point estimates without confidence intervals. Each could be addressed through follow-on work at larger corpus scale.

6. Future Research Directions

Expanded corpus and longitudinal study. Extension from 117 to several thousand applications, re-analysed at six-month intervals over two to three years, would permit empirical examination of whether disclosure posture is improving, deteriorating, or static in aggregate, and whether individual applications update their policies in response to regulatory events.

Multi-language support. All policies in the corpus are English; replacement of the base model with a multilingual alternative, or addition of a translation preprocessing step, would extend the methodology to non-English-jurisdiction applications.

GDPR article-level mapping. Mapping each declared permission and its disclosure to the specific subsections of GDPR Article 13 would transform the tool from a gap detector into a structured compliance assessor.

Ensemble methods. Adapting several different base models to the same task would enable inter-model consistency analysis, with regions of disagreement flagged for human review.

Publication of the ground-truth dataset. The 3,483 permission-policy pairs, together with model classifications and the manually validated subset, constitute a potentially valuable benchmark resource worth publishing under an academically-reusable licence.

Consumer-facing deployment. A mobile companion application that returns a Privacy Health Score from a Play Store URL would translate the research findings into direct consumer impact.

Refined Privacy Health Score. A revised PHS formulation that also penalises a high density of unclear classifications would resolve the pathology identified in Section 5.6 and produce more discriminating scores for policies that are systematically vague rather than systematically silent.

7. Conclusion

This research has investigated the feasibility of automated identification of privacy-disclosure gaps in Android applications using a compact, domain-adapted large language model running on consumer-grade hardware. Applied to a corpus of 117 widely-used Android applications, the adapted mobilellama_gap_scratch model achieved F1 = 0.930 on the covered class against a keyword-oracle ground truth across 3,483 permission-policy pairs and 175 analysis runs, with precision 0.892 and recall 0.971, accuracy 0.912, macro-average F1 0.906 and weighted-average F1 0.911. The result substantially exceeds the stated project target of F1 greater than

or equal to 0.85 and is broadly competitive with, and on the covered class marginally ahead of, the rule-based and classical-machine-learning approaches reported in prior published work.

The research further contributes three descriptive findings. First, approximately 35 per cent of declared Android permissions are either not disclosed or only vaguely disclosed in the corresponding policy, and 69.2 per cent of applications have at least one undisclosed declared permission. Second, disclosure is substantially less complete for functional permissions (Bluetooth, NFC, notifications, SMS) than for infrastructure permissions (network, location, billing). Third, the POST_NOTIFICATIONS permission and other members of the NOTIFICATIONS group are under-disclosed in a large fraction of applications that request them, reflecting the lag between Android platform evolution and policy updates.

Methodologically, the research documents several lessons applicable to other fine-tuning efforts: training-data format alignment is decisive; conservative hyper-parameters are safer when base-model capability is valuable to preserve; deterministic pre-processing can eliminate a meaningful fraction of inference calls without loss of accuracy; and post-acquisition verification at every stage of a multi-source pipeline is essential. From a policy perspective, the research demonstrates that analysis historically requiring expensive human review can now be performed at scale on modest hardware, opening possibilities for regulators, researchers, and consumer advocates to undertake systematic sweeps of application-store catalogues that would not have been economically feasible previously.

The research is subject to the limitations of a single-language, single-point-in-time, small-corpus study, each of which points towards productive follow-on work. On balance, however, the research supports an affirmative answer to its principal question: compact, domain-adapted large language models can identify permission-disclosure gaps at a level of accuracy that renders them practically useful, complementing rather than replacing expert human judgement in the privacy-compliance domain. As the legislative landscape around personal data continues to evolve, tools capable of monitoring disclosure compliance at scale are likely to be of increasing value; PrivacyTotal is one such tool.

8. References

Andow, B., Mahmud, S. Y., Wang, W., Whitaker, J., Enck, W., Reaves, B., Singh, K., and Xie, T. (2019). PolicyLint: Investigating Internal Privacy Policy Contradictions on Google Play. *Proceedings of the 28th USENIX Security Symposium*, 585-602.

Barrera, D., Kayacik, H. G., van Oorschot, P. C., and Somayaji, A. (2010). A Methodology for Empirical Analysis of Permission-Based Security Models and Its Application to Android. *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 73-84.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., et al. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33, 1877-1901.

Chu, X., Qiao, L., Lin, X., Xu, S., Yang, Y., Hu, Y., Wei, F., Zhang, X., Zhang, B., Wei, X., and Shen, C. (2023). MobileLLaMA: Towards an Open, Resource-Efficient Language Model for Mobile Deployment. arXiv preprint arXiv:2312.16886.

Cook, T. D., and Campbell, D. T. (1979). *Quasi-Experimentation: Design and Analysis Issues for Field Settings*. Houghton Mifflin.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. *Advances in Neural Information Processing Systems*, 36.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT 2019*, 4171-4186.

Felt, A. P., Chin, E., Hanna, S., Song, D., and Wagner, D. (2011). Android Permissions Demystified. *Proceedings of the 18th ACM Conference on Computer and Communications Security*, 627-638.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *Proceedings of the International Conference on Learning Representations 2022*.

McDonald, A. M., and Cranor, L. F. (2008). The Cost of Reading Privacy Policies. *I/S: A Journal of Law and Policy for the Information Society*, 4(3), 540-565.

OpenAI (2023). GPT-4 Technical Report. arXiv preprint arXiv:2303.08774.

Ravichander, A., Black, A. W., Wilson, S., Norton, T., and Sadeh, N. (2019). Question Answering for Privacy Policies: Combining Computational and Legal Perspectives. *Proceedings of EMNLP-IJCNLP 2019*, 4947-4958.

Srinath, M., Wilson, S., and Giles, C. L. (2021). PrivBERT: Privacy Policy Classification with Domain-Specific Pretraining. *Proceedings of the 2021 Workshop on Natural Language Processing for Privacy (PrivNLP)*.

Story, P., Zimmeck, S., Ravichander, A., Smullen, D., Wang, Z., Reidenberg, J., Russell, N. A., and Sadeh, N. (2019). Natural Language Processing for Mobile App Privacy Compliance. *Proceedings of the AAAI Spring Symposium on Privacy-Enhancing AI and Language Technologies*.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Roziere, B., et al. (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv preprint arXiv:2302.13971.

Wilson, S., Schaub, F., Dara, A. A., Liu, F., Cherivirala, S., Leon, P. G., Andersen, M. S., Zimmeck, S., Sathyendra, K. M., Russell, N. A., Norton, T. B., Hovy, E., Reidenberg, J., and Sadeh, N. (2016).

The Creation and Analysis of a Website Privacy Policy Corpus. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 1330–1340.

Zimmeck, S., Wang, Z., Liu, L., Adjerid, I., Story, P., Smullen, D., Schaub, F., Sadeh, N., Bellovin, S. M., and Reidenberg, J. (2017). Automated Analysis of Privacy Requirements for Mobile Apps. *Proceedings of the 2017 Network and Distributed System Security Symposium*.

Appendix A: List of Analysed Applications

The 117 Android application package identifiers included in the analytical corpus, listed alphabetically.

#	Package ID	#	Package ID	#	Package ID
1	com.adobe.reader	41	com.google.android.dialer	81	com.outfit7.mytalkingtom
2	com.alibaba.aliexpresshd	42	com.google.android.gm	82	com.outfit7.mytalkingtom2
3	com.amazon.avod.thirdpartyclient	43	com.google.android.keep	83	com.outfit7.talkingangela
4	com.amazon.kindle	44	com.google.android.youtube	84	com.outfit7.talkinggin ger
5	com.amazon.mShop.android.shopping	45	com.hulu.plus	85	com.outfit7.talkingtomgoldrun
6	com.amazon.music	46	com.ideashower.readitlater.pro	86	com.pandora.android
7	com.apple.android.music	47	com.imangi.templerun	87	com.playrix.fishdom
8	com.audible.application	48	com.imangi.templerun2	88	com.playrix.gardenscapes
9	com.babbel.mobile.android.en	49	com.imdb.mobile	89	com.playrix.homescapes
10	com.badoo.mobile	50	com.innersloth.spacemafia	90	com.playrix.township
11	com.bumble.app	51	com.instagram.android	91	com.quora.android
12	com.calm.android	52	com.kakao.talk	92	com.raongames.growcastle
13	com.disney.disneyplus	53	com.kayak.android	93	com.reddit.frontpage
14	com.duckduckgo.mobile.android	54	com.kiloo.subwaysurf	94	com.revolut.revolut
15	com.duolingo	55	com.king.candycrushsodasaga	95	com.roblox.client
16	com.ea.game.pvzfree_row	56	com.linkedin.android	96	com.rovio.angrybirdsfriends
1	com.ea.games.r3_na	5	com.microsoft.azure	97	com.skype.raider

7		7			
18	com.ea.gp.fifamobile	58	com.microsoft.bing	98	com.slack
19	com.ebay.mobile	59	com.microsoft.copilot	99	com.snapchat.android
20	com.evernote	60	com.microsoft.cortana	100	com.soundcloud.android
21	com.fingersoft.hillclimb	61	com.microsoft.intune	101	com.spotify.music
22	com.fsck.k9	62	com.microsoft.launcher	102	com.supercell.boombeach
23	com.garmin.android.apps.connectmobile	63	com.microsoft.office.excel	103	com.supercell.brawlstars
24	com.github.android	64	com.microsoft.office.officelens	104	com.supercell.clashofclans
25	com.google.android.apps.authenticator2	65	com.microsoft.office.powerpoint	105	com.supercell.hayday
26	com.google.android.apps.bard	66	com.microsoft.office.word	106	com.transferwise.android
27	com.google.android.apps.classroom	67	com.microsoft.skydrive	107	com.truecaller
28	com.google.android.apps.docs	68	com.microsoft.teams	108	com.tumblr
29	com.google.android.apps.dynamite	69	com.microsoft.todos	109	com.twitter.android
30	com.google.android.apps.fitness	70	com.miniclip.plagueinc	110	com.waze
31	com.google.android.apps.magazines	71	com.netflix.mediaclient	111	com.whatsapp
32	com.google.android.apps.maps	72	com.nianticlabs.pokemongo	112	com.yelp.android
33	com.google.android.apps.meetings	73	com.nike.ntc	113	com.yodo1.crossyroad
34	com.google.android.apps.nbu.paisa.user	74	com.nike.plusgps	114	com.zhiliaoapp.musically
35	com.google.android.apps.photos	75	com.nike.snkrs	115	de.danoeh.antennapod
36	com.google.android.apps.safetyhub	76	com.noodlecake.altos	116	org.mozilla.firefox
37	com.google.android.apps.tachyon	77	com.nordvpn.android	117	org.schabi.newpipe
38	com.google.android.apps.translate	78	com.openai.chatgpt		
39	com.google.android.apps.youtube.music	79	org.telegram.messenger		
40	com.google.android.calendar	80	org.thunderdog.challegram		

Appendix B: Permission Group Mapping

Group	Canonical Android Permissions
NETWORK	INTERNET, ACCESS_NETWORK_STATE, ACCESS_WIFI_STATE
LOCATION	ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION, ACCESS_BACKGROUND_LOCATION
STORAGE	READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE, READ_MEDIA_*
CAMERA	CAMERA
MICROPHONE	RECORD_AUDIO
CONTACTS	READ_CONTACTS, WRITE_CONTACTS
PHONE	READ_PHONE_STATE, CALL_PHONE, READ_CALL_LOG
NOTIFICATIONS	POST_NOTIFICATIONS, VIBRATE, WAKE_LOCK
ACCOUNTS	GET_ACCOUNTS, MANAGE_ACCOUNTS
BILLING	BILLING, VENDING
SENSORS	BODY_SENSORS, ACTIVITY_RECOGNITION
BLUETOOTH	BLUETOOTH, BLUETOOTH_CONNECT, BLUETOOTH_SCAN
SMS	SEND_SMS, RECEIVE_SMS, READ_SMS
NFC	NFC, NFC_TRANSACTION_EVENT
CALENDAR	READ_CALENDAR, WRITE_CALENDAR
SYSTEM	UPDATE_DEVICE_STATS, SYSTEM_ALERT_WINDOW, WRITE_SETTINGS

Appendix C: Training Hyper-Parameters

```
CONFIG = {
    "base_model":              "mtgv/MobileLLaMA-2.7B-Chat",
    "load_in_4bit":            True,
    "bnb_4bit_quant_type":     "nf4",
    "bnb_4bit_use_double_quant":  True,
    "lora_r":                  8,
    "lora_alpha":              16,
    "lora_dropout":            0.05,
    "lora_target_modules":     ["q_proj", "k_proj", "v_proj", "o_proj"],
    "num_train_epochs":        3,
    "per_device_train_batch_size": 2,
    "gradient_accumulation_steps": 8,
    "learning_rate":           5e-5,
```

```
    "lr_scheduler_type":          "cosine",
    "warmup_ratio":               0.1,
    "max_grad_norm":              0.3,
    "weight_decay":               0.01,
    "max_seq_length":             512,
    "optim":                      "paged_adamw_8bit",
    "fp16":                       True,
    "seed":                       42,
}
```

Training loss progression:

Step	1	100	300	500	700	900	969
Loss	2.91	1.13	0.67	0.45	0.37	0.34	0.34

Appendix D: Sample Model Outputs

Sample outputs are drawn from applications in the analytical corpus (Appendix A). PHS values are the latest recorded figures for each application.

Temple Run 2 (com.imangi.templerun2), PHS 30

Permission	Coverage	Model Response
ACCESS_FINE_LOCATION	covered	Mentioned: The policy states the app records your approximate location via your device's IP address, which is used for tax reasons and for security and fraud prevention.
VIBRATE	not_mentioned	Not mentioned: The policy does not appear to disclose the use of device vibration or haptic feedback.

WhatsApp (com.whatsapp), PHS 81

Permission	Coverage	Model Response
READ_CONTACTS	covered	Mentioned: The policy explicitly describes accessing the user's contact list in order to provide messaging features, with matching performed on the device.
RECORD_AUDIO	covered	Mentioned: The policy

		describes microphone access as necessary to support encrypted voice and video calling; audio content does not leave the device in unencrypted form.

Brawl Stars (com.supercell.brawlstars), PHS 0

Permission	Coverage	Model Response
BLUETOOTH_CONNECT	not_mentioned	Not mentioned: The policy does not appear to discuss the application's use of Bluetooth connections to paired devices.
POST_NOTIFICATIONS	not_mentioned	Not mentioned: The policy does not describe the application's use of push notifications or in-device alerts.

End of Document