

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»
Факультет компьютерных наук
Образовательная программа «Компьютерные науки и анализ данных»

NotiSift: фильтр рекламных уведомлений на Android с ML-классификацией push-уведомлений

Выполнил студент:

группы БКНАД242

Федоров Арсений Викторович

Руководитель проекта:

Куренков Владимир Вячеславович

старший преподаватель

Факультет компьютерных наук

Департамент больших данных и информационного поиска

Актуальность работы

- Критически важные данные (ОТР-пароли, статусы заказов) смешиваются с агрессивной рекламой в едином потоке.
- Когнитивная перегрузка: входящие уведомления вызывают произвольное переключение контекста и снижают концентрацию внимания пользователя.
- Несовершенство системных механизмов Android: встроенные инструменты работают на уровне метаданных приложения или канала, полностью игнорируя семантику текста.
- Проблема «грубой блокировки»: при отключении уведомлений от маркетплейса пользователь избавляется от спама, но рискует пропустить важный статус доставки.

Цель и задачи работы

Главная цель: Разработка и экспериментальная оценка прототипа NotiSift - автономного Android-приложения для локальной классификации и блокировки рекламных push-уведомлений.

Основные задачи:

- Формализация задачи фильтрации и подготовка текстового датасета.
- Выбор компактной ML-архитектуры для работы в условиях ограничений смартфонов.
- Разработка полного пайплайна машинного обучения: от предобработки до тонкой настройки модели.
- Оптимизация весов нейросети: экспорт в формат ONNX и int8-квантизация для сжатия размера.
- Проектирование и разработка Android-клиента с функцией перехвата уведомлений и выполнения инференса без доступа в интернет.

Требования к программному продукту

Функциональные спецификации:

- Асинхронный перехват системных событий и точное извлечение текста уведомлений.
- Выполнение локального инференса нейросети в изолированном потоке без доступа к интернету.
- Расчет вероятности спама и автоматическое скрывание сообщений при превышении заданного порога.
- Ведение подробного локального журнала для аудита заблокированных уведомлений.
- Создание понятного графического интерфейса с возможностью настройки чувствительности и управления белым списком.

Нефункциональные характеристики:

- Приватность: гарантия того, что личные переписки не покинут файловую систему смартфона.
- Автономность: приложение работает сразу после установки без дозагрузки весов модели из сети.
- Компактность: экономия оперативной и дисковой памяти мобильного устройства.
- Воспроизводимость: детерминированный запуск всех скриптов и сборки приложения.

Анализ существующих решений

1. **Системные механизмы Android (каналы уведомлений):** обеспечивают глубокую интеграцию, но работают только на уровне всего приложения или канала, полностью игнорируя смысл конкретного сообщения.
2. **Утилиты на основе правил (BuzzKill, FilterBox):** позволяют создавать фильтры по ключевым словам. Минус в том, что они требуют кропотливой ручной настройки и перестают работать при малейшем изменении маркетинговых шаблонов.
3. **Программы отложенной доставки (Daywise):** собирают несрочные уведомления в пакеты для выдачи в заданное время. Снижают количество отвлечений, но не удаляют рекламный мусор как таковой.
4. **Преимущество NotiSift:** применение алгоритмов машинного обучения для автоматического семантического анализа текста без необходимости писать сложные пользовательские правила.

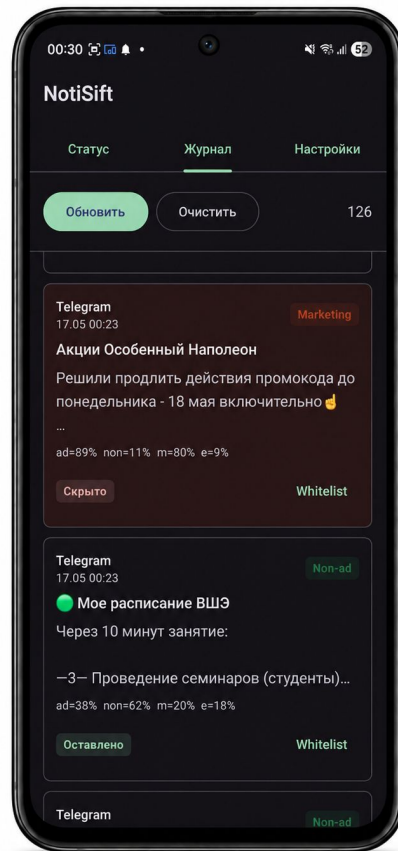
Результат разработки

Готовое нативное Android-приложение, написанное на языке Kotlin с использованием UI-фреймворка Jetpack Compose.

Интеграция сжатой ML-модели через движок ONNX Runtime для быстрого выполнения вычислений прямо на устройстве.

Стабильная работа фоновой службы: система перехватывает и анализирует уведомления автономно, не обращаясь к интернету.

Прозрачный пользовательский интерфейс: наличие визуального журнала аудита, отображение рассчитанных вероятностей и удобное управление белым списком.



Подход к созданию датасета

Этические ограничения: сбор реальных уведомлений невозможен из-за наличия приватной переписки и финансовых данных, поэтому выбран подход генерации синтетических данных.

Использование LLM: корпус из 5000 реалистичных уведомлений в формате JSONL сгенерирован с помощью модели GPT 5.2 Pro.

Имитация мобильного стиля: при генерации задавались жесткие условия на использование эмодзи, англицизмов, сокращений и лаконичность (средняя длина текста около 85 символов).

Защита от запоминания брендов: одни и те же популярные приложения (например, маркетплейсы) принудительно распределялись по разным классам, чтобы нейросеть анализировала смысл, а не имя отправителя.

Структура классов: данные разделены на три категории: non_ad (полезная информация), marketing (агрессивная реклама) и engagement (вовлечение без прямой выгоды).

Как обучалась модель

Базовая архитектура: для проекта выбрана компактная русскоязычная модель RuBERT-tiny2, которая отлично улавливает контекст и помещается в память смартфона.

Формирование признаков: на вход алгоритму подавался только склеенный текст заголовка и тела сообщения. Имя приложения было намеренно удалено для исключения привязки к брендам.

Алгоритм оптимизации: применялся метод AdamW с линейным разогревом и разными скоростями обучения. Предобученные слои трансформера обновлялись осторожно, а новый слой классификации обучался более агрессивно.

Компенсация дисбаланса: так как рекламных сообщений в датасете было меньше, использовалась взвешенная функция потерь. Классу маркетинга был присвоен максимальный штраф за ошибку.

Меры против переобучения: применялся целый комплекс защитных механик, включая dropout, сглаживание меток и раннюю остановку (early stopping) по метрике Macro-F1.

Результаты обучения

Высокая предсказательная способность: на отложенной тестовой выборке общая точность (Accuracy) составила более 92%.

Эффективная блокировка спама: полнота обнаружения нежелательной информации (Recall агрегированного класса) достигла 98.2%.

Безопасность для пользователя: анализ матрицы ошибок показал минимальный риск потери данных. Лишь 12 полезных сообщений из 750 были ошибочно классифицированы как спам.

Успешное сжатие: алгоритм 8-битной квантизации уменьшил размер нейросети в 4 раза (со 117 МБ до 29 МБ) при полном сохранении первоначальной точности.

Превосходство над классикой: сравнение подтвердило, что трансформер справляется с задачей лучше, чем базовые математические фильтры вроде наивного Байеса.

Как разрабатывался механизм управления уведомлениями на Android и inference модели

Служба перехвата: использование системного интерфейса `NotificationListenerService` для мгновенного и асинхронного получения текста входящих уведомлений.

Нативная токенизация: реализация алгоритма разбиения текста `WordPiece` на языке Kotlin для подготовки данных без использования тяжелых Python-библиотек.

Движок ONNX Runtime: выполнение вычислений (инференса) строго в оперативной памяти телефона с помощью оптимизированной C++ библиотеки.

Изолированная обработка: перенос всех вычислений в фоновый поток (`Executor`) для предотвращения зависаний интерфейса операционной системы.

Оптимизация скорости: применение механизма «прогрева» нейросети при запуске и отказ от сжатия файла весов в архиве приложения для быстрого холодного старта.

Логика блокировки: программный вызов метода `cancelNotification` для скрытия спама с учетом белых списков и пользовательских настроек чувствительности.

Результаты курсовой работы

Успешное создание проекта: разработан полнофункциональный прототип нативного Android-приложения NotiSift, готовый к установке и эксплуатации.

Реализация ML-контура: собран репрезентативный датасет на 5000 примеров и выстроен полностью воспроизводимый процесс обучения нейросети.

Высокое качество фильтрации: обученная компактная модель RuBERT-tiny2 показала точность распознавания спама на уровне 97 процентов.

Глубокая оптимизация: алгоритмы квантизации позволили сжать размер нейросети более чем в 4 раза (до 29 мегабайт) без потери предсказательной способности.

Подтверждение концепции: на практике доказана жизнеспособность подхода privacy-first AI, при котором анализ текста выполняется прямо на устройстве без утечки личных данных.

Перспективы проекта

Публикация в Google Play, создание iOS версии приложения;

Выход на международный рынок, монетизация приложения по модели подписки;

Обогащение датасета для адаптации классификатора к работе с другими языками помимо русского и английского;

Список литературы и источников

- [1] Cary Stothart, Ainsley Mitchum и Courtney Yehnert. «The attentional cost of receiving a cell phone notification». В: *Journal of Experimental Psychology: Human Perception and Performance* 41.4 (2015), с. 893—897. DOI: [10.1037/xhp0000100](https://doi.org/10.1037/xhp0000100).
- [2] Android Developers. *Create and manage notification channels*. 2026. URL: <https://developer.android.com/develop/ui/compose/notifications/channels> (дата обр. 17.04.2026).
- [3] Android Developers. *Notification runtime permission*. 2026. URL: <https://developer.android.com/develop/ui/compose/notifications/notification-permission> (дата обр. 15.04.2026).
- [4] Android Developers. *NotificationListenerService*. 2026. URL: <https://developer.android.com/reference/android/service/notification/NotificationListenerService> (дата обр. 20.04.2026).
- [5] Google Play. *BuzzKill Notification Manager*. 2026. URL: <https://play.google.com/store/apps/details?id=com.samruston.buzzkill> (дата обр. 13.05.2026).
- [6] Google Play. *FilterBox Notification Manager*. 2026. URL: <https://play.google.com/store/apps/details?id=com.catchingnow.np> (дата обр. 16.05.2026).
- [7] Daywise. *Daywise: schedule notifications*. 2026. URL: <https://getdaywise.com> (дата обр. 18.05.2026).
- [8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser и Illia Polosukhin. «Attention Is All You Need». В: *Advances in Neural Information Processing Systems*. 2017, с. 5998—6008. URL: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html> (дата обр. 03.03.2026).
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee и Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». В: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1: Long and Short Papers*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, с. 4171—4186. DOI: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423). URL: <https://aclanthology.org/N19-1423/> (дата обр. 07.03.2026).

Список литературы и источников

- [10] Yuri Kuratov и Mikhail Arkhipov. *Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language*. arXiv preprint arXiv:1905.07213. 2019. URL: <https://arxiv.org/abs/1905.07213> (дата обр. 12.03.2026).
- [11] Victor Sanh, Lysandre Debut, Julien Chaumond и Thomas Wolf. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv preprint arXiv:1910.01108. 2019. URL: <https://arxiv.org/abs/1910.01108> (дата обр. 18.03.2026).
- [12] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang и Denny Zhou. «MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices». В: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, с. 2158—2170. DOI: 10.18653/v1/2020.acl-main.195. URL: <https://aclanthology.org/2020.acl-main.195/> (дата обр. 24.03.2026).
- [13] cointegrated. *rubert-tiny2 model card*. 2026. URL: <https://huggingface.co/cointegrated/rubert-tiny2> (дата обр. 09.05.2026).
- [14] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest и Alexander Rush. «Transformers: State-of-the-Art Natural Language Processing». В: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, 2020, с. 38—45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: <https://aclanthology.org/2020.emnlp-demos.6/> (дата обр. 01.04.2026).
- [15] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu и Ming Yin. «Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations». В: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, 2023, с. 10443—10461. DOI: 10.18653/v1/2023.emnlp-main.647. URL: <https://aclanthology.org/2023.emnlp-main.647> (дата обр. 06.04.2026).
- [16] Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu и Shafiq Joty. *Data Augmentation using Large Language Models: Data Perspectives, Learning Paradigms and Challenges*. arXiv preprint arXiv:2403.02990. 2024. URL: <https://arxiv.org/abs/2403.02990> (дата обр. 09.04.2026).
- [17] Microsoft. *How to develop a mobile application with ONNX Runtime*. 2026. URL: <https://onnxruntime.ai/docs/tutorials/mobile/> (дата обр. 26.04.2026).
- [18] TensorFlow. *Post-training quantization — TensorFlow Model Optimization*. 2024. URL: https://www.tensorflow.org/model_optimization/guide/quantization/post_training (дата обр. 05.05.2026).
- [19] Microsoft. *Quantize ONNX models*. 2026. URL: <https://onnxruntime.ai/docs/performance/model-optimizations/quantization.html> (дата обр. 02.05.2026).