Agentic Knowledge Distillation: Autonomous Training of Small Language Models for SMS Threat Detection

Adel ElZemity
School of Computing
University of Kent
Canterbury, United Kingdom
ae455@kent.ac.uk

Joshua Sylvester
School of Computing
University of Kent
Canterbury, United Kingdom
jrs71@kent.ac.uk

Budi Arief
School of Computing
University of Kent
Canterbury, United Kingdom
b.arief@kent.ac.uk

Rogério De Lemos
School of Computing
University of Kent
Canterbury, United Kingdom
r.delemos@kent.ac.uk

***Abstract*—SMS-based phishing (smishing) attacks have surged, yet training effective on-device detectors requires labelled threat data that quickly becomes outdated. To deal with this issue, we present Agentic Knowledge Distillation, which consists of a powerful LLM acts as an autonomous teacher that fine-tunes a smaller student SLM, deployable for security tasks without human intervention. The teacher LLM autonomously generates synthetic data and iteratively refines a smaller on-device student model until performance plateaus. We compare four LLMs in this teacher role (Claude Opus 4.5, GPT 5.2 Codex, Gemini 3 Pro, and DeepSeek V3.2) on SMS spam/smishing detection with two student SLMs (Qwen2.5-0.5B and SmolLM2-135M). Our results show that performance varies substantially depending on the teacher LLM, with the best configuration achieving 94.31% accuracy and 96.25% recall. We also compare against a Direct Preference Optimisation (DPO) baseline that uses the same synthetic knowledge and LoRA setup but without iterative feedback or targeted refinement; agentic knowledge distillation substantially outperforms it (e.g. 86–94% vs 50–80% accuracy), showing that closed-loop feedback and targeted refinement are critical. These findings demonstrate that agentic knowledge distillation can rapidly yield effective security classifiers for edge deployment, but outcomes depend strongly on which teacher LLM is used.**

***Index Terms*—agentic AI, knowledge distillation, large language models, synthetic data generation, SMS spam detection**

I. Introduction

SMS-based attacks have become a critical cyber security threat. Smishing (SMS phishing) attacks increased by over 300% in recent years, with attackers exploiting mobile messaging to deliver phishing links, fake banking alerts, cryptocurrency scams, and social engineering attempts [1]. Unlike email spam, SMS messages bypass many traditional security filters and benefit from higher user trust, and the recipients are more likely to click on the link contained in the SMS, making them particularly effective attack vectors. The challenge for defenders is that SMS spam evolves rapidly, requiring security models that can adapt quickly to new threat patterns.

Deploying large language models (LLMs) for on-device SMS filtering faces significant challenges: high computational costs, latency constraints, and privacy concerns when processing personal messages. A natural way to address these limitations is to use small language models (SLMs).

Following recent taxonomies in the field [2], [3], we define an SLM as a neural language model with a parameter count typically under 10 billion (e.g., 100M to 7B), characterised by its ability to perform inference on local, resource-constrained consumer hardware without reliance on cloud infrastructure. Conversely, we define LLMs as models exceeding this parameter threshold that typically exhibit emergent generalist capabilities but necessitate massive distributed memory for execution. Within this architectural boundary, we specifically adopt the functional perspective of Belcak et al. [4], identifying an SLM as a model capable of serving a single user's agentic workloads with practically acceptable latency on their local device.

SLMs offer a practical alternative for edge deployment, but they require task-specific fine-tuning, a process that traditionally demands substantial human expertise and labelled data, which may not capture emerging threat patterns. Knowledge distillation [5] addresses this gap. A teacher LLM (typically an LLM with strong task capability) transfers its knowledge to a student SLM, a smaller model that is trained to replicate the teacher LLM's behaviour and can then be deployed where the teacher LLM cannot. In our setting, the student is an SLM for on-device SMS filtering, and the teacher is an LLM that provides the task knowledge. However, conventional distillation requires human engineers to design data pipelines, analyse errors, and iterate on training strategies, a slow process ill-suited to rapidly evolving threats.

As a non-agentic baseline, we also consider Direct Preference Optimisation (DPO), a recently proposed alignment and fine-tuning paradigm that optimises a model directly from preference pairs without an explicit reinforcement learning loop. In our context, DPO provides a useful point of comparison for assessing whether the additional complexity of agentic knowledge distillation is justified. Rather than using an autonomous LLM agent to iteratively generate data and refine the student, DPO fine-tunes the student SLM in a single-stage procedure using synthetic preference data derived from teacher LLM outputs. This allows us to isolate the benefits of iterative, agent-driven adaptation over a simpler, static fine-tuning pipeline.

We propose a novel *Agentic Knowledge Distillation* approach, which is an automated framework where the LLM itself serves as an autonomous machine learning (ML) engineer. Given a task specification and evaluation criteria, the LLM agent generates synthetic training data from its knowledge (including patterns of phishing, smishing, and social engineering attacks), executes fine-tuning on the student SLM, evaluates performance against a separate, teacher-generated synthetic validation set, and iterates until performance plateaus. This ensures the optimisation process remains entirely self-contained and independent of human-labelled data, and enables rapid adaptation to new threat categories.

For our teacher LLM We choose four models ranked within the top 10 on the Artificial Analysis Intelligence Index (as of January 2026) [6]:

- Claude Opus 4.5 from Anthropic [7],
- GPT 5.2 Codex from OpenAI [8],
- Gemini 3 Pro from Google [9],
- DeepSeek V3.2 [10].

We compare these four as teacher LLMs to test whether leading general-purpose LLMs transfer knowledge effectively to SLMs. For student SLMs, we use Qwen2.5-0.5B-Instruct (494M parameters) and SmolLM2-135M-Instruct (135M parameters). These can be deployed on consumer hardware with acceptable latency, span a useful range of capacity to test how student SLM size interacts with teacher LLM quality, and are instruction-tuned for consistent task formatting.

We pair each teacher LLM with each student SLM (eight configurations), using the SMS Spam Collection as a strictly held-out test set [11]. Each teacher LLM generates synthetic training data, fine-tunes the student SLM with Low-Rank Adaptation (LoRA), and iterates based only on aggregate evaluation metrics, never on raw test examples. Our results reveal that teacher LLM capability significantly impacts student SLM performance, with substantial variation in final accuracy and precision-recall balance across teacher LLMs.

Our contributions are:

1) We introduce an agentic knowledge distillation framework for autonomous fine-tuning of deployable student SLMs using a teacher LLM.
2) We provide a study showing that teacher LLM choice is important for the effectiveness of agentic knowledge distillation.

The rest of the paper is organised as follows. Section II reviews related work on SMS spam detection, knowledge distillation, synthetic data generation, and LLM agents. Section III describes our methodology, including the agentic workflow and strict train-test separation. Section IV outlines the experimental setup. The results are discussed in Section V. Section VI discusses implications and limitations. Section VII concludes the paper and presents directions of future research. The system prompt given to each teacher LLM is in the Appendix.

II. Related Work

This section reviews related work on SMS spam detection, knowledge distillation, and LLM-based synthetic data generation, and situates our approach within recent agentic and preference-based training methods.

SMS Spam and Smishing Detection. SMS spam detection has been studied extensively as both a text classification problem and a security challenge [11]. Traditional approaches used Naive Bayes, SVM, and rule-based filters [12]. However, attackers continuously adapt their tactics, using URL shorteners, homoglyphs, and social engineering to evade detection. Recent work has explored deep learning approaches, but these typically require large labelled datasets that become outdated as attack patterns evolve [13]. Our approach addresses this by leveraging LLM knowledge of attack patterns to generate diverse synthetic training data without requiring manual labelling of emerging threats.

Knowledge Distillation. Knowledge distillation, introduced by Hinton et al. [5], transfers knowledge from large models to smaller ones through soft label matching. Recent work has extended this to LLMs, with approaches like DistilBERT [14] and TinyBERT [15] demonstrating effective compression. Our work differs by eliminating the need for a shared training dataset: the teacher LLM generates data directly from its parametric knowledge.

Synthetic Data Generation. LLMs have shown remarkable capability in generating training data for various tasks. Self-Instruct [16] demonstrated that models can generate instruction-following data to improve their own performance, while Stanford Alpaca [17] used this approach to create instruction-tuned models. We extend this concept to cross-model distillation for security applications, where a large model generates synthetic attack and benign examples to train a smaller, deployable detector.

LLM Agents. Recent advances in LLM agents [18], [19] have enabled autonomous task completion through tool use and iterative reasoning. Our work applies this agentic paradigm to the ML engineering process itself, treating model training as a task the LLM can autonomously complete. Combined with parameter-efficient fine-tuning methods like LoRA [20], this enables rapid iteration on consumer hardware.

Preference Optimisation. Preference-based fine-tuning has emerged as an alternative to reinforcement learning from human feedback (RLHF) for aligning language models with desired behaviours [21]. Direct Preference Optimisation (DPO) [21] formulates preference learning as a supervised objective over paired outputs, avoiding the need for an explicit reward model or online policy optimisation. DPO and related methods have been successfully applied to instruction following and helpfulness alignment in general-purpose LLMs [21]. We apply DPO in a security classification context by preference-tuning the student SLM on teacher LLM generated preferences over candidate classifications. This serves as a strong baseline that relies on static synthetic supervision without any failure pattern-driven data generation.

III. Methodology

Given a teacher LLM T, a student SLM S_θ, and a task description D, our goal is to fine-tune the student SLM for detecting SMS spam and smishing using only synthetic data generated by the teacher LLM T. The teacher LLM must autonomously generate all training data and orchestrate the fine-tuning process without access to any test data.

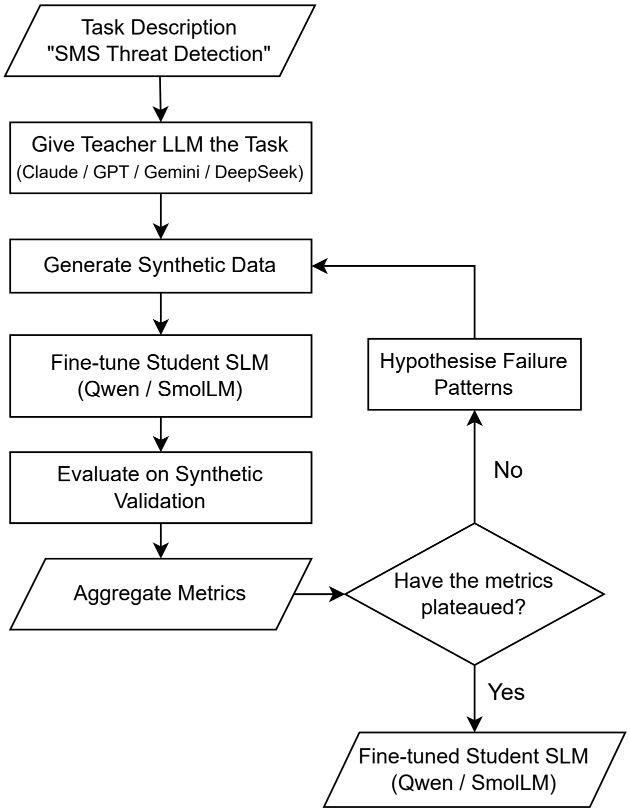

Fig. 1. Agentic knowledge distillation workflow. The teacher LLM generates both synthetic training and synthetic validation sets. The conditional loop uses metrics derived strictly from the synthetic validation set. The final evaluation on the human-labelled test set on the outputted fine-tuned student SLM.

A. Workflow Overview

Figure 1 presents the agentic knowledge distillation workflow used to fine-tune a student SLM for SMS threat detection under the supervision of a teacher LLM. The workflow is formulated as an autonomous, closed-loop optimisation process, where the teacher LLM iteratively generates data, fine-tunes the student SLM, evaluates performance, and refines its strategy until performance converges.

The process begins with a task specification ("SMS Threat Detection") that is provided to the teacher LLM (e.g., Claude, GPT, Gemini, or DeepSeek), as shown at the top of Figure 1. Each teacher LLM is given an identical system prompt defining the mission objective and the structure of the iterative loop (see Appendix). This prompt specifies:

Algorithm 1 Agentic Knowledge Distillation Loop

```
Require: Teacher LLM $T$, Student SLM $S_\theta$, Task $D$
$T$ generates fixed synthetic validation set $\mathcal{V}$ from $D$
repeat
    $T$ generates/refines synthetic training set $\mathcal{F}_t$
    Update student parameters $\theta$ by minimizing $\mathcal{L}(\theta)$ on $\mathcal{F}_t$
    $M_t \leftarrow \text{Evaluate}(S_\theta, \mathcal{V})$
    if $M_t$ indicates failure patterns then
        $T$ hypothesizes weaknesses and generates targeted
        hard negatives for $\mathcal{F}_{t+1}$
    end if
until $M_t$ plateaus or max iterations reached
return Fine-tuned student $S_\theta$
```

1) the objective of fine-tuning the student SLM for SMS threat detection, and
2) an iterative cycle comprising synthetic data generation, fine-tuning, evaluation, error analysis, and refinement.

Given the task, the teacher LLM first generates a synthetic training dataset $\mathcal{F}$, which is used to fine-tune the student SLM (e.g., Qwen2.5-0.5B or SmolLM2-135M). The fine-tuned student is then evaluated on a held-out synthetic validation set $\mathcal{V}$, which is generated once at the start of the workflow and remains fixed across iterations. The resulting performance metrics are aggregated and fed back to the teacher LLM.

The teacher LLM then determines whether performance has plateaued. If the validation metrics are deemed to have converged, the workflow terminates and the final fine-tuned student SLM is produced as the output of the pipeline. Otherwise, the teacher LLM performs error analysis by hypothesising failure patterns (e.g., systematic misclassification of particular phishing styles or obfuscation strategies) and generates additional targeted synthetic data to address these weaknesses. This newly generated data is incorporated into the training set, and the fine-tuning and evaluation loop repeats until the stopping criterion is met.

Throughout this process, the teacher LLM operates as an autonomous agent with full terminal access, enabling it to install libraries, generate datasets, execute fine-tuning scripts, and invoke the external evaluation system. Crucially, the optimisation loop is driven solely by synthetic validation metrics. The real-world SMS Spam Collection dataset is strictly *air-gapped* from the agentic workflow and is used only for final, post-hoc evaluation to assess real-world generalisation. The logical structure of this autonomous, closed-loop optimization is formally defined in Algorithm 1. This algorithm details the transition from initial synthetic data generation to the iterative refinement stages based on aggregate performance feedback.

1) Synthetic Data Generation: The teacher LLM generates a set of samples from its own knowledge of the domain then writes a script that would automate generating a larger synthetic training dataset

$$\mathcal{F} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{N},$$

where $\tilde{x}_i$ is a synthetic SMS message and $\tilde{y}_i \in \{0, 1\}$ denotes a benign or malicious label assigned by construction.

Synthetic messages are generated from the teacher LLM's knowledge of the domain and cover a diverse range of attack categories, including phishing, smishing, fake delivery alerts, cryptocurrency scams, lottery fraud, and premium-rate abuse. Benign messages include casual conversations, workplace communications, and legitimate service notifications.

To avoid class imbalance, the agent is instructed to generate approximately equal numbers of spam and ham messages ($50/50$ split).

2) Student SLM Fine-Tuning with LoRA: The student SLM is fine-tuned using Low-Rank Adaptation (LoRA). Rather than updating all the model parameters, LoRA injects small trainable matrices into selected layers while keeping the base model frozen. This enables efficient training on consumer hardware.

Let S_θ denote the student SLM where θ are the trainable LoRA parameters. Training minimises binary cross-entropy loss over the synthetic dataset:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[\tilde{y}_i \log S_\theta(\tilde{x}_i) + (1-\tilde{y}_i)\log(1 - S_\theta(\tilde{x}_i))\right].$$

3) Internal Synthetic Validation vs. External Testing: **1) Internal Loop:** For the iterative refinement, the Teacher LLM evaluates the Student SLM on the synthetic validation set $\mathcal{V}$. This provides the feedback vector M_t used to hypothesise failure patterns. Since $\mathcal{V}$ is generated by the Teacher LLM, the "ground truth" labels are known without human intervention. At iteration t, the teacher LLM receives a metric vector

$$M_t = (\mathrm{Acc}_t, \mathrm{Prec}_t, \mathrm{Rec}_t, \mathrm{FP}_t, \mathrm{FN}_t).$$

2) External Evaluation: The human-labelled SMS Spam Collection is used solely as a final "sanity check" after the agentic process has terminated. These external metrics are reported in our Results tables I, II, III but are never fed back to the Teacher LLM.

4) Error Analysis and Targeted Refinement: Based solely on aggregate metrics, the teacher LLM hypothesises likely failure patterns as shown in Figure 1. For example, a high false-positive rate suggests that legitimate service messages may be misclassified as spam. In contrast, a high false-negative rate indicates insufficient coverage of subtle or short-form scams.

The teacher LLM then generates additional synthetic examples specifically targeting these hypothesised weaknesses. This process resembles hard negative mining, but differs in that no actual misclassified examples are observed [22]. Instead, the teacher LLM relies on domain knowledge to construct targeted refinements.

The refined dataset is used for further LoRA fine-tuning, and the evaluation-feedback loop repeats until performance plateaus. This refinement cycle, represented by steps 6 through 8 in Algorithm 1, ensures that the student SLM is exposed to increasingly difficult synthetic examples that directly address its current predictive weaknesses.

B. Evaluation of Framework

To evaluate whether the performance gains observed in our experiments arise from the proposed agentic knowledge distillation framework itself, rather than merely from access to high-quality synthetic data, we introduce a teacher LLM-generated Direct Preference Optimisation (DPO) baseline. This baseline is designed to isolate the contribution of closed-loop, agent-driven refinement by providing a strong non-agentic alternative that uses the same teacher models, the same synthetic knowledge source, and the same fine-tuning setup, but without iterative feedback or targeted data generation.

Specifically, we fine-tune the student SLM using Direct Preference Optimisation (DPO) [21] on preference data generated by a teacher LLM and compare the resulting performance against the full agentic knowledge distillation framework. This enables a controlled comparison between static preference-based fine-tuning and the proposed autonomous, closed-loop training process.

For each teacher LLM T, we construct a synthetic preference dataset. Each sample comprises a prompt x (an SMS message) and a pair of candidate responses (y^+, y^-), where y^+ represents the preferred classification output and y^- a less suitable alternative, as determined by the teacher LLM. Preferences are generated using the same domain knowledge, attack categories, and class balance assumptions as in the agentic knowledge distillation setting, ensuring that both approaches are trained under comparable supervision signals.

The student SLM S_θ is then fine-tuned using DPO under an identical LoRA configuration to the agentic method. All architectural choices, LoRA ranks, scaling factors, learning rates, batch sizes, and training budgets are held constant to ensure a fair comparison between training regimes. Crucially, the DPO baseline is trained in a single offline stage and does not receive any iterative metric feedback, perform failure analysis, or generate targeted refinements.

Formally, DPO optimises the student SLM to increase the likelihood of preferred outputs relative to non-preferred ones under the teacher-induced preference distribution. In contrast to the agentic workflow, this approach does not involve hypothesis generation, error analysis, or adaptive hard-negative synthesis. As a result, it provides a controlled non-agentic baseline that isolates the effect of preference-based fine-tuning from the benefits of autonomous closed-loop reasoning.

The DPO-trained models are evaluated using the same strictly held-out test set. This enables direct comparison between (i) zero-shot baselines, (ii) teacher LLM-generated DPO fine-tuning, and (iii) the proposed agentic knowledge distillation framework.

C. Summary

The resulting system forms a closed-loop agentic knowledge distillation workflow in which the teacher LLM autonomously generates data, fine-tunes the student SLM, and iteratively improves performance using only metric-level feedback. Figure 1 summarises this workflow.

The strict separation between data generation and evaluation ensures that observed improvements reflect genuine generalisation rather than overfitting. We additionally compare against a teacher LLM-generated DPO baseline that leverages the same synthetic knowledge and identical LoRA configurations, but lacks iterative feedback and targeted refinement. This comparison highlights the contribution of agentic reasoning and closed-loop adaptation beyond static preference-based fine-tuning.

IV. Experiments

We design our experiments to evaluate (i) the effectiveness of agentic knowledge distillation for SMS threat detection, (ii) the impact of teacher LLM choice, and (iii) the benefit of closed-loop refinement compared to static synthetic fine-tuning.

A. Models

We evaluate four teacher LLMs: Claude Opus 4.5 [7], GPT 5.2 Codex [8], Gemini 3 Pro [9], and DeepSeek V3.2 [10], all accessed via OpenRouter [23]. Each teacher is used to fine-tune two student SLMs: Qwen2.5-0.5B-Instruct [24] and SmolLM2-135M-Instruct [25], using identical LoRA configurations across all runs.

B. Datasets and Evaluation Protocol

For evaluation, we use the SMS Spam Collection [11], a public corpus of 5,574 SMS messages aggregating data from the Grumbletext UK forum (425 spam), NUS SMS Corpus (3,375 ham), Caroline Tag's PhD thesis (450 ham), and SMS Spam Corpus v.0.1. To ensure a fair and interpretable evaluation, we construct a balanced test subset of 1,494 messages (747 spam, 747 ham), and remove the remaining ham messages from the evaluation set. This prevents the reported metrics from being dominated by the majority "ham" class, which would otherwise inflate accuracy and obscure failure modes on the minority spam class. Balancing the evaluation set enables more meaningful comparison across models and training regimes, particularly for recall and F1 score, which are critical for assessing threat detection performance. Crucially, the agentic training workflow never had access to the content of the test data, which remained strictly held out and air-gapped from the synthetic data generation and fine-tuning process.

C. Experimental Methods

We conduct the following experiments:

- **Zero-shot baselines.** Both student SLMs are evaluated without any fine-tuning using zero-shot prompting to establish a lower bound on performance.
- **DPO with synthetic data.** For each teacher LLM, we generate a synthetic preference dataset of 10,000 samples and fine-tune each student SLM using Direct Preference Optimisation (DPO) with LoRA. This setting isolates the effect of synthetic data generation without closed-loop refinement.
- **Agentic knowledge distillation.** For each teacher LLM and student SLM pair, we run the full closed-loop agentic workflow, in which the teacher autonomously generates synthetic training and validation data, analyses errors, and iteratively refines the training set until validation performance plateaus.

This yields a total of $4 \times 2 \times 3$ experimental configurations (four teachers, two students, and three experimental methods).

D. Implementation Details

All experiments are conducted on a Mac Mini M4 with 16GB unified memory. For fairness, we use identical optimisation settings, LoRA ranks, learning rates, and stopping criteria across all experimental conditions. We record token usage for each agentic run to characterise computational cost. For the LoRA configuration we use a rank of 32, scaling factor of 64, learning rate of 5×10^{-5}, and batch size of 8.

V. Results

This section presents results in three stages. We first report zero-shot performance of the student SLMs without any fine-tuning to establish the initial baseline. We then evaluate a teacher LLM-generated Direct Preference Optimisation (DPO) baseline, which uses a single offline round of synthetic preference data without iterative refinement. Finally, we present results for the proposed agentic knowledge distillation framework, analysing performance across different teacher LLMs and student SLMs, and comparing against both the zero-shot and DPO baselines.

TABLE I
Baseline Performance (No Fine-tuning)

Model	Acc.	Prec.	Recall	F1
Qwen2.5-0.5B	49.80%	28.57%	0.27%	0.53%
SmolLM2-135M	46.85%	46.74%	45.11%	46.00%

Both student SLMs were first evaluated without fine-tuning using zero-shot prompting (Table I). Qwen2.5-0.5B exhibits an extreme bias towards predicting "ham", correctly identifying only 2 out of 747 spam messages, resulting in near-zero recall and F1 score. SmolLM2-135M performs close to random chance, indicating limited inherent discriminative capability for the task in the absence of task-specific fine-tuning.

Table II reports results for the teacher LLM-generated DPO baseline. While DPO fine-tuning yields modest improvements over the zero-shot baselines in some settings, performance remains limited and highly variable across teacher–student pairs. In particular, Qwen2.5-0.5B remains close to chance performance under DPO across all teachers, whereas SmolLM2-135M benefits more substantially, achieving up to 80.25% accuracy when trained on Claude-generated preference data. These results highlight the limitations of single-stage, static preference fine-tuning in the absence of iterative error-driven refinement.

TABLE II
Final Performance by Data-Generating Teacher LLM (DPO)

Teacher LLM	Student SLM	Acc.	Prec.	Recall	F1	Tokens
Claude Opus 4.5	Qwen2.5-0.5B SmolLM2-135M	52.74% **80.25%**	72.52% **81.81%**	52.74% **80.25%**	39.45% **80.01%**	2142.48K
GPT 5.2 Codex	Qwen2.5-0.5B SmolLM2-135M	51.34% 52.61%	57.30% 70.57%	51.34% 52.61%	38.86% 39.38%	1721.52K
Gemini 3 Pro	Qwen2.5-0.5B SmolLM2-135M	52.54% 68.27%	68.93% 77.33%	52.54% 68.27%	39.44% 65.41%	1839.37K
DeepSeek V3.2	Qwen2.5-0.5B SmolLM2-135M	52.48% 76.44%	68.09% 78.44%	52.48% 76.44%	39.40% 76.02%	1892.48K

TABLE III
Final Performance by Agentic Knowledge Distillation Teacher LLM

Teacher LLM	Student SLM	Acc.	Prec.	Recall	F1	Tokens	Time
Claude Opus 4.5	Qwen2.5-0.5B SmolLM2-135M	**94.31%** **86.28%**	92.65% 80.38%	**96.25%** 95.98%	**94.42%** **87.00%**	27.91K 30.56K	∼7 min ∼6 min
GPT 5.2 Codex	Qwen2.5-0.5B SmolLM2-135M	71.08% 59.71%	**98.76%** 55.38%	42.70% **99.87%**	59.64% 71.26%	26.37K 24.37K	∼6 min ∼5 min
Gemini 3 Pro	Qwen2.5-0.5B SmolLM2-135M	85.21% 80.46%	79.29% 72.73%	95.31% 97.46%	86.58% 83.31%	14.83K 16.78K	∼9 min ∼8 min
DeepSeek V3.2	Qwen2.5-0.5B SmolLM2-135M	92.10% 86.21%	91.77% **88.70%**	92.50% 83.00%	92.13% 85.75%	31.2K 32.1K	∼8 min ∼7 min

TABLE IV
Accuracy Improvement by Agentic Knowledge Distillation Teacher LLM (Δ from Baseline)

Teacher LLM	Qwen2.5-0.5B	SmolLM2-135M
Claude Opus 4.5	**+44.51 pp**	**+39.43 pp**
DeepSeek V3.2	+42.30 pp	+39.36 pp
Gemini 3 Pro	+35.41 pp	+33.61 pp
GPT 5.2 Codex	+21.28 pp	+12.86 pp

Table III presents the final performance across all teacher LLM and student SLM combinations, and Table IV shows the accuracy improvements from baseline in Table I. The results reveal substantial variation in the effectiveness of agentic knowledge distillation across teacher LLMs.

Comparing Table II and Table III reveals a clear and consistent performance gap between the teacher LLM-generated DPO baseline and the proposed agentic knowledge distillation approach. While DPO preference-tuning yields modest improvements over zero-shot baselines, performance remains limited, with accuracies typically ranging from 50–80% and substantially lower F1 scores, particularly for the Qwen2.5-0.5B student SLM. In contrast, agentic knowledge distillation achieves markedly higher and more stable performance across all teacher LLM and student SLM pairs, reaching up to 94.31% accuracy and 94.42% F1 (Table III).

Claude Opus 4.5 achieved the best overall results, producing well-balanced classifiers with high accuracy (86–94%), strong precision (80–93%), and excellent recall (96%). The hard negative mining phase proved particularly effective, i.e., for Qwen2.5-0.5B, false positives decreased from 207 to 57 after refinement.

GPT 5.2 Codex produced models with severe precision-recall imbalances. For Qwen2.5-0.5B, the fine-tuned model achieved very high precision (98.76%) but poor recall (42.70%), missing more than half of spam messages. For SmolLM2-135M, the opposite occurred: near-perfect recall (99.87%) but low precision (55.38%), flagging nearly half of legitimate messages as spam. Both configurations resulted in lower overall accuracy than Claude-trained models.

DeepSeek V3.2 achieved strong, well-balanced results, with 86–92% accuracy, precision and recall both in the 83–93% range. For Qwen2.5-0.5B, it reached 92.10% accuracy and 92.13% F1, and for SmolLM2-135M, 86.21% accuracy and 85.75% F1. Its balanced precision-recall contrasts with GPT's severe imbalances and places it among the most effective teacher LLMs.

Gemini 3 Pro achieved intermediate results, ranking behind Claude Opus 4.5 and DeepSeek V3.2. It produced reasonably balanced classifiers with 80–85% accuracy, moderate precision (73–79%), and high recall (95–97%). Notably, Gemini 3 pro was the most token-efficient teacher LLM, using only 14–17K tokens compared to 24–33K for the other models. However, its lower precision compared to Claude Opus 4.5 and DeepSeek V3.2 suggests less effective hard negative mining for reducing false positives.

The GPT 5.2 Codex results highlight a critical failure pattern in agentic knowledge distillation: when the teacher

LLM generates imbalanced or insufficiently diverse synthetic data, the student SLM develops biased decision boundaries. The Qwen2.5-0.5B model trained by GPT 5.2 Codex became overly conservative (high precision, low recall), while the SmolLM2-135M model became overly aggressive (high recall, low precision).

VI. Discussion

The comparison between the DPO baseline and agentic knowledge distillation, as observed in Section V, provides insight into the source of the performance gains. Although DPO leverages the same teacher LLMs, synthetic domain knowledge, and identical LoRA configurations, its substantially lower accuracy and F1 scores (Table II) indicate that static preference-based optimisation alone is insufficient to induce robust decision boundaries. In comparison, the agentic method's iterative use of aggregate evaluation feedback to hypothesise failure patterns and generate targeted synthetic refinements consistently produces well-balanced classifiers (Table III). This suggests that the primary benefit does not arise from the data source itself, but from the closed-loop reasoning and adaptive training dynamics enabled by agentic knowledge distillation, which are critical for closing precision–recall gaps and achieving strong generalisation.

Our results demonstrate that teacher LLM selection is a critical factor in agentic knowledge distillation. The performance gap between teacher LLMs is substantial: Claude Opus 4.5 achieved 86–94% accuracy, DeepSeek V3.2 achieved 86–92% with balanced precision-recall, Gemini 3 Pro achieved 80–85%, and GPT 5.2 Codex achieved only 60–71%. This 25+ percentage point spread suggests that not all LLMs are equally effective as autonomous machine learning (ML) engineers.

Several factors may explain these differences. First, the coverage and structure of the synthetic training data appear to vary across teacher LLMs. The stronger performance of Claude Opus 4.5 and DeepSeek V3.2 suggests that their generated datasets more consistently captured a broad range of spam strategies (e.g., phishing, scams, obfuscation) as well as legitimate conversational patterns, leading to more robust decision boundaries. In contrast, Gemini 3 Pro has high recall but lower precision which indicates that its synthetic data may have insufficiently represented challenging ham examples, causing the student models to over-generalise spam patterns and produce more false positives. GPT 5.2 Codex exhibits extreme precision–recall imbalances, pointing to systematic biases in the synthetic data generation process that skewed the student models toward overly conservative or overly aggressive classification regimes. Second, the effectiveness of the teacher LLM's error analysis and hard negative mining differs: Claude Opus 4.5 and DeepSeek V3.2 appear more capable of identifying recurring failure modes and generating targeted counterexamples, leading to more stable and well-calibrated classifiers.

The precision-recall imbalances observed with GPT 5.2 Codex are particularly instructive. A spam detector (Qwen2.5-0.5B trained by GPT 5.2 Codex) with 98.76% precision but 42.70% recall would miss most spam in practice, despite appearing "precise." Conversely, a detector (SmolLM2-135M trained by GPT 5.2 Codex) with 99.87% recall but 55.38% precision would flag too many legitimate messages. These failure patterns highlight that agentic knowledge distillation can fail dramatically if the teacher LLM cannot generate balanced, high-quality synthetic data.

Interestingly, Gemini 3 Pro was the most token-efficient teacher LLM (14–17K tokens vs. 25–33K for others), yet was outperformed by Claude Opus 4.5 and DeepSeek V3.2. This suggests that token count alone does not determine success: the quality of reasoning and data generation matters more than quantity.

Security Implications. From a cyber security perspective, agentic knowledge distillation offers both opportunities and risks. On the defensive side, this approach enables rapid creation of threat detectors without requiring labelled datasets of emerging attack patterns: the LLM's knowledge of phishing tactics, social engineering, and fraud schemes can be distilled into lightweight, on-device models. This is particularly valuable for SMS security, where privacy concerns make cloud-based filtering undesirable. The best-performing configuration (Claude Opus 4.5 + Qwen2.5-0.5B) achieved 96.25% recall, meaning it catches the vast majority of malicious messages while maintaining 92.65% precision to avoid blocking legitimate communications.

However, the same technique could potentially be misused to train models for generating convincing phishing messages or evading detection. The observed variation in teacher LLM effectiveness may also apply to adversarial applications, though we note that generating effective attacks is likely harder than detecting them.

A key advantage of this framework is the resolution of the label scarcity paradox. Because the teacher LLM generates its own validation "ground truth," the iterative feedback loop does not require a human-labelled oracle. The system is autonomous, requiring only a high-level task description to begin self-improvement.

Limitations. Performance is bounded by the teacher LLM's knowledge of the task domain: emerging attack patterns not represented in the LLM's training data may be missed. Additionally, while the agent never accesses test data content, it requires aggregate evaluation metrics on the synthetic validation set to guide iteration. The quality of synthetic data depends on the teacher LLM's knowledge, which may not cover all edge cases in real-world threats.

VII. Conclusions

We introduced agentic knowledge distillation, demonstrating that a large language model (LLM) can autonomously fine-tune a small language model (SLM) for security-critical tasks like SMS spam and smishing detection. Our evaluation of four teacher LLMs (Claude Opus 4.5, DeepSeek V3.2, Gemini 3 Pro, and GPT 5.2 Codex) reveals that teacher LLM selection critically impacts outcomes. Claude achieved the best results (86–94% accuracy with balanced precision-recall), DeepSeek achieved strong results (86–92% with balanced precision-recall), Gemini achieved moderate success (80–85%), while GPT produced models with severe imbalances (60–71%). The best configuration achieved 96.25% recall for spam detection, demonstrating the potential for effective on-device mobile security.

To contextualise this, we also evaluated a teacher LLM-generated Direct Preference Optimisation (DPO) baseline. While DPO fine-tuning improved performance over zero-shot baselines, it remained substantially lower than agentic knowledge distillation across all teacher LLM and student SLM pairs (50–80% accuracy, lower F1 scores), highlighting that iterative feedback, hypothesis generation, and targeted data refinement are critical components of the agentic approach.

Using SLMs as students is motivated by the need for local deployment: filtering runs on the device so that message content is never sent to external APIs (preserving privacy), latency remains acceptable for real-time use, and inference is feasible on consumer hardware where LLMs cannot run. These findings establish that careful teacher LLM selection is essential for successful agentic knowledge distillation, and that LLMs can serve as autonomous ML engineers for rapid development of security classifiers without requiring labelled threat data. The entire process requires only 15–33K tokens and completes in under 10 minutes on consumer hardware.

This work suggests that agentic knowledge distillation is promising for security applications but requires careful teacher LLM selection. Future work should investigate adaptation to new threat categories, ensemble approaches combining multiple teacher LLMs, and longitudinal evaluation against evolving attack patterns.

Acknowledgements

This work was partly supported by the funding received from the UK EPSRC project EP/X036707/1 on Countering HArms caused by Ransomware in the Internet Of Things (CHARIOT). The authors would also like to thank the anonymous reviewers for their constructive feedback.

References

[1] Proofpoint, "2024 state of the phish report," Proofpoint Inc., Tech. Rep., 2024.

[2] F. Wang, S. Wang, L. Zhang *et al.*, "A comprehensive survey of small language models in the era of large language models," *arXiv preprint arXiv:2409.02641*, 2024.

[3] L. Zhang, H. Zhang, and et al., "Small language models: Survey, measurements, and insights," *International Conference on Machine Learning (ICML)*, 2024.

[4] P. Belcak, G. Heinrich, S. Diao, Y. Fu, X. Dong, S. Muralidharan, Y. C. Lin, and P. Molchanov, "Small language models are the future of agentic AI," *arXiv preprint arXiv:2506.02153*, 2025.

[5] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[6] Artificial Analysis. (2025) Artificial analysis intelligence index. Accessed: 2026-01-31. [Online]. Available: https://artificialanalysis.ai/

[7] Anthropic. (2025) Claude Opus 4.5. [Online]. Available: https://www.anthropic.com/

[8] OpenAI. (2025) GPT-5.2 Codex. [Online]. Available: https://openai.com/

[9] Google. (2025) Gemini 3 pro. [Online]. Available: https://ai.google.dev/

[10] DeepSeek-AI, "DeepSeek-V3.2: Pushing the frontier of open large language models," *arXiv preprint arXiv:2512.02556*, 2025.

[11] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in *Proceedings of the 11th ACM Symposium on Document Engineering (DocEng '11)*, Mountain View, CA, USA, 2011, pp. 259–262.

[12] S. Mishra and D. Soni, "Sms phishing and mitigation approaches," in *Proceedings of the 12th International Conference on Contemporary Computing (IC3)*, Noida, India, 2019, pp. 1–5.

[13] S. Gadde, A. Lakshmanarao, and S. Satyanarayana, "SMS spam detection using machine learning and deep learning techniques," in *Proceedings of the 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, 2021, pp. 358–362.

[14] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[15] X. Jiao *et al.*, "TinyBERT: Distilling BERT for natural language understanding," in *Findings of EMNLP*, 2020.

[16] Y. Wang *et al.*, "Self-instruct: Aligning language models with self-generated instructions," in *Proceedings of the Association for Computational Linguistics (ACL)*, 2023.

[17] R. Taori *et al.*, "Stanford Alpaca: An instruction-following LLaMA model," GitHub repository, 2023. [Online]. Available: https://github.com/tatsu-lab/stanford_alpaca

[18] Significant Gravitas, "AutoGPT: An autonomous GPT-4 experiment," GitHub repository, 2023.

[19] H. Chase, "LangChain," GitHub repository, 2022.

[20] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022.

[21] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: your language model is secretly a reward model," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2023.

[22] J. D. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=CR1XOQ0UTh-

[23] OpenRouter. (2025) OpenRouter: One API for all LLMs. [Online]. Available: https://openrouter.ai/

[24] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu *et al.*, "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024. [Online]. Available: https://arxiv.org/abs/2412.15115

[25] L. Ben Allal, A. Lozhkov, E. Bakouch, G. Márquez Blázquez, G. Penedo, L. Tunstall *et al.*, "Smollm2: When smol goes big – data-centric training of a small language model," *arXiv preprint arXiv:2502.02737*, 2025. [Online]. Available: https://arxiv.org/abs/2502.02737

Appendix

The following system prompt was provided to all teacher LLM's:

> **Mission Objective:** Fine-tune [Student SLM] into a world-class SMS spam detector on this Mac Mini M4.
> **Constraint:** Use the mlx-lm library to leverage the M4's GPU.
> **Iterative Workflow:** (1) **Baseline Evaluation**: Run the base model and calculate Accuracy, Recall, and Precision. (2) **Agentic Knowledge Distillation**: Using your own knowledge, generate two distinct datasets: (a) a synthetic training set of 2,000+ examples, and (b) a held-out synthetic validation set ($\mathcal{V}$) of 500 examples for internal metrics. Both should be balanced 50/50 Spam/Ham and cover diverse categories: phishing, smishing, fake delivery alerts, crypto scams, and aggressive marketing. (3) **Fine-Tuning**: Execute a LoRA fine-tune on the student SLM using the synthetic data. (4) **Evaluation & Feedback Loop**: Evaluate performance. Analyse aggregate error metrics and hypothesise which patterns may be causing errors. Generate targeted hard negatives based on these hypotheses. Repeat if performance hasn't plateaued. (5) **Final Output**: Provide the final adapter weights and performance report.