



DOI 10.28925/2663-4023.2026.35.1430

UDC 004.056.53:004.85

Vasyl Ye. Luchyk

Doctor of Economic Sciences, Professor, Professor of the Department of Combating Cybercrime

Kharkiv National University of Internal Affairs, Kharkiv, Ukraine

ORCID: 0000-0001-6007-910X

luchik.vasil@gmail.com

Vitalii V. Nosov

Candidate of Technical Sciences, Associate Professor, Professor of the Department of Combating Cybercrime

Kharkiv National University of Internal Affairs, Kharkiv, Ukraine

ORCID: 0000-0002-7848-6448

vitnos@univd.edu.ua

Kateryna E. Vitvitska

Student of the Educational and Research Institute No. 4 (Training of Specialists in Information and Analytical Support and Cybersecurity of the National Police of Ukraine)

Kharkiv National University of Internal Affairs, Kharkiv, Ukraine

ORCID: 0009-0005-9932-1289

katyhavehamster@gmail.com

Iryna A. Manzhai

Senior Lecturer of the Department of Information Systems and Technologies

Kharkiv National University of Internal Affairs, Kharkiv, Ukraine

ORCID: 0000-0003-2664-4472

ir.lily@i.ua

ARCHITECTURAL APPROACHES TO APPLYING LANGUAGE MODELS FOR DETECTING PHISHING AND UNWANTED EMAILS

Abstract. Phishing remains one of the most common cyberattack vectors: in 2025, CERT-UA handled twice as many phishing incidents as in 2024. This article aims to systematize architectural approaches to using language models for detecting phishing and unwanted emails and to experimentally compare specific models and routing schemes. The approaches are classified according to how the model is used: task adaptation, extended analysis, and hybrid solutions incorporating classical machine learning. An experiment on 17,479 English emails, with near-duplicates kept together in a group-based split, compared TF-IDF logistic regression, fine-tuned XLM-RoBERTa (three runs), prompt-based Qwen2.5-7B-Instruct without an external provider's API, and cascade and parallel hybrids. Transfer was evaluated on Nazario + Enron, two machine-translated Ukrainian datasets, and 94 authentic Ukrainian messages. On the English test set, logistic regression and XLM-R achieved F1 scores of 0.970 and 0.969, whereas the LLM achieved 0.849 (exploratory because of earlier prompt selection) and required approximately 71 times as much processing time as XLM-R (T4 GPU, batch processing). On translated data, XLM-R retained F1 scores of 0.960 and 0.889, whereas the LLM's false positive rate reached 29.3%. On authentic Ukrainian messages, the LLM achieved F1 = 0.820 versus 0.509 for XLM-R (mean of three runs), mainly on SMS and messenger messages. An absolute uncertainty band on calibrated XLM-R scores escalated 1–12% of emails to the LLM, with cascade F1 scores close to those of XLM-R on English and translated data; most XLM-R errors on authentic messages were not escalated. Prompt injection almost completely disabled the LLM's detection capability; a simple system-prompt instruction mitigated this only partially. With the same prompt and threshold, Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3 produced excessive false positives, indicating that the prompt requires tuning for each model. The authentic-message dataset is small and its classes differ in genre, so conclusions for Ukrainian messages remain preliminary.

Keywords: phishing; phishing email detection; large language models; small language models; cybersecurity; machine learning; hybrid detection systems; calibration; adversarial attacks; Ukrainian-language data.



INTRODUCTION

Phishing is a form of social engineering that seeks to obtain credentials, financial information, or other unauthorized access through deception by presenting a malicious message or resource as legitimate. Over the past decade, it has remained one of the leading initial compromise vectors for both individuals and organizations. Compared with exploiting technical vulnerabilities, phishing requires relatively few resources, which explains its continuing popularity among attackers with varying levels of expertise.

The relevance of this problem to Ukraine is supported by data from CERT-UA, the national team responsible for responding to cyber incidents, cyberattacks, and cyber threats, operating under the State Service of Special Communications and Information Protection of Ukraine. In 2025, the team handled 5,927 cyber incidents, an increase of 37.4% over 2024 (4,315 incidents) [1]. Over the same period, the number of phishing incidents handled more than doubled, from 843 to 1,727 (Fig. 1).

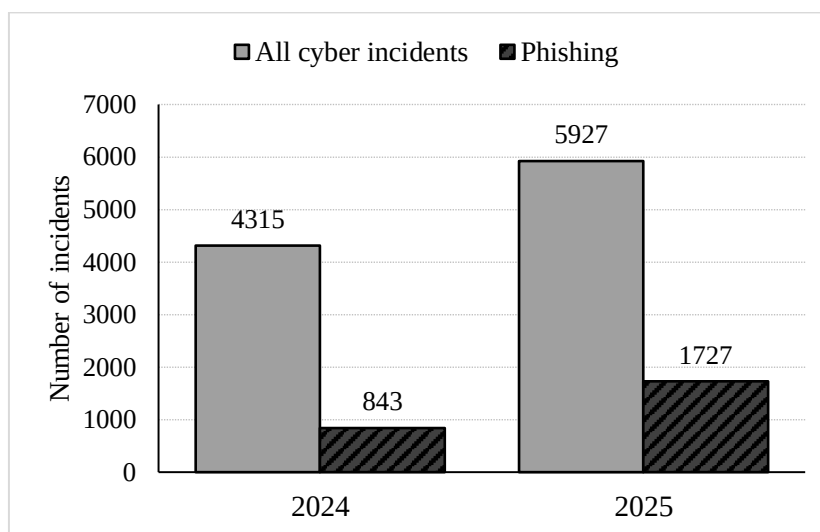


Fig. 1. Number of cyber incidents and phishing attacks in Ukraine in 2024–2025

Source: compiled by the authors based on CERT-UA data [2]

In 2025, phishing ranked second among cyber incident types, after malware distribution, and accounted for 29.1% of all incidents handled (Fig. 2). This trend indicates that phishing remains a major attack vector against Ukrainian organizations and that automated detection methods require further improvement. These figures describe incidents handled by CERT-UA and partly reflect improvements in detection; they should therefore not be interpreted as a comprehensive estimate of the scale of phishing in Ukraine.

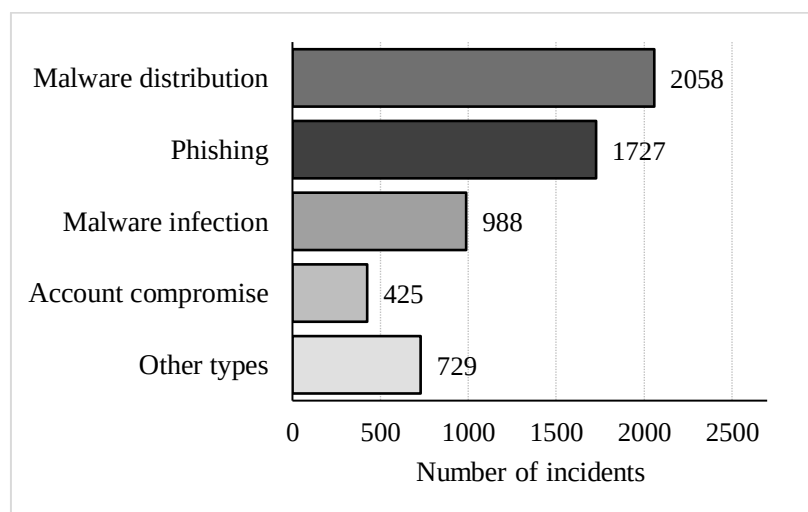


Fig. 2. Distribution of cyber incidents in Ukraine by type in 2025

Source: compiled by the authors based on CERT-UA data [2]



The availability of generative large language models (LLMs) has changed this threat landscape in two ways. First, LLMs enable attackers to automatically generate grammatically polished, persuasive, and personalized phishing messages in multiple languages. This reduces the effectiveness of heuristic and signature-based methods that rely on typical language errors or formulaic expressions. Second, the same models can perform semantic analysis and identify manipulative wording and inconsistencies between a domain and the purported sender at a level beyond classical machine learning methods based on manual feature engineering.

This dual role of LLMs as tools for both attack and defense motivates a systematic analysis of existing research to identify architectures that are promising for practical phishing detection and the limitations that should be anticipated during deployment.

Problem statement. Despite the large number of publications, most studies evaluate LLMs on English-language data, under different conditions, and primarily as commercial cloud services. Practical deployment, particularly in public institutions and security-sector organizations where transmitting email content to external services is unacceptable, requires an understanding of how a locally deployed LLM compares with compact fine-tuned models and hybrid schemes under identical conditions. Relevant criteria include detection performance, false positive rate, computational cost, robustness to adversarial modifications, and performance on Ukrainian-language data.

Related work. Research on the use of LLMs to counter phishing has developed rapidly since 2023–2024 and can be broadly divided into the following directions.

One direction evaluates off-the-shelf models for phishing email classification without additional fine-tuning. Studies [3] and [4] compared several commercial and open models for binary email classification and concluded that their accuracy was comparable to that of classical classifiers. Study [5] examined LLMs as tools for both generating and detecting phishing emails and found that model-generated messages were not always detected more effectively than human-written messages.

A second direction develops specialized LLM-based systems. Koide and colleagues first developed ChatPhishDetector for website analysis and subsequently ChatSpamDetector for email analysis. In both systems, the language model receives structured information about a page or email before producing a conclusion with a natural-language explanation. The analyst can therefore examine both the classification result and the arguments supporting it [6], [7].

A third direction concerns multimodal solutions. Researchers have used multimodal LLMs to jointly analyze a page screenshot, its textual content, and the brand being impersonated. KnowPhish combines LLMs with a multimodal knowledge graph of brands to improve reference-based phishing webpage detection [8], [9].

A fourth direction comprises hybrid architectures in which LLMs complement classical machine learning methods. The authors of [10] proposed a scheme in which quantized Mistral-7B and Mixtral-8x7B models semantically assess a URL, while a CatBoost gradient boosting model classifies it using lexical and statistical features. This division of roles balances accuracy and interpretability.

A fifth direction involves multi-agent architectures. In PhishDebate [11], several LLM-based agents argue opposing positions concerning a page's legitimacy, reducing false positives compared with a single-agent decision.

A separate group of studies addresses the practical choice between approaches. Researchers have compared prompt engineering and fine-tuning for phishing detection and evaluated small language models for phishing website detection in terms of cost, performance, and data privacy [12], [13].

Studies [14]–[16] critically examine the limitations of LLM-based detectors. Study [14] demonstrates the vulnerability of both classical and LLM-based models to adversarial webpage modifications. Study [15], which benchmarks LLMs for small and medium-sized enterprises, identifies a discrepancy between high classification accuracy and inadequate uncertainty estimates. Study [16] documents successful uses of commercial LLMs to generate phishing scenarios, highlighting the problem of dual use.

Ukrainian researchers are also actively developing phishing detection methods. Study [17] proposes a classification of phishing attacks and detection methods, together with a phishing website classifier. Study [18] introduces a hybrid neural architecture for phishing email detection, combining a convolutional network, BiLSTM, and attention, which outperformed logistic regression and support vector machines. In [19], phishing is detected using LLM-based conversational assistants that receive standardized queries about phishing indicators, with their responses combined through weighted aggregation; mean classification accuracy was 97.5%. However, these studies do not compare locally deployed LLMs with fine-tuned multilingual models and hybrid schemes under identical conditions.

Most studies thus use English-language data without accounting for less widely used languages, including Ukrainian. This gap is particularly relevant to Ukraine given the increase in phishing incidents (see Fig. 1). The cost-effectiveness of LLM deployment has been addressed only partially, and experimental comparisons of locally deployed LLMs with compact fine-tuned models and hybrid schemes under identical conditions, including on Ukrainian-language data, remain scarce.



Aim of the study. This article aims to systematize architectural approaches to using language models for detecting phishing and unwanted emails and to experimentally compare specific models and routing schemes under defined data and computational constraints. These include prompt-based classification with a locally deployed generative model, a fine-tuned multilingual encoder, a classical classifier, and cascade and parallel hybrid schemes. Evaluation covers English emails, their machine translations into Ukrainian, and a pilot dataset of authentic Ukrainian messages.

To achieve this aim, the following objectives were defined: (1) classify architectural approaches according to how LLMs are used within a detection system; (2) compare these approaches with one another and with classical machine learning in terms of accuracy, computational cost, data confidentiality, and explainability; (3) summarize the limitations of LLM-based detectors and possible mitigation strategies; (4) identify promising directions for further research, including the needs of the Ukrainian-language information environment; and (5) experimentally assess detection performance, false positive rate, computational cost, and robustness to adversarial modifications for baseline models and hybrid schemes, including on Ukrainian-language data.

THEORETICAL BACKGROUND

The studies analyzed in [3]–[13] were classified according to how LLMs are used within a phishing detection system. This criterion covers three aspects: how the model is adapted to the task; how many information sources and models contribute to a decision; and whether the model operates independently or alongside a classical classifier. Three groups of approaches were identified accordingly: (1) task adaptation through prompt-based classification, fine-tuning, and small language models; (2) extended analysis using multimodal and multi-agent debate architectures; and (3) hybrid solutions combining LLMs with classical machine learning. The approaches were compared with one another and with classical methods in terms of accuracy, computational cost, data confidentiality, and explainability.

This grouping is provisional: groups are distinguished by their dominant characteristic, whereas a particular system may combine several characteristics. For example, a fine-tuned small model may form part of a hybrid or multimodal system. The groups should therefore be understood as design axes (model adaptation, the number of information sources and models contributing to a decision, and integration with a classical classifier) rather than mutually exclusive classes.

Group 1. Task adaptation through prompts, fine-tuning, and small language models

The simplest approach to deployment involves sending a structured query, or prompt, to a pretrained general-purpose LLM. The prompt contains the email text or a page description and instructions concerning the response format, such as a classification label and justification. This approach requires no dedicated model-training infrastructure and enables rapid development. However, performance depends strongly on the prompt wording. Using a cloud API also incurs per-request charges and network latency and transfers email content to an external provider, which may be unacceptable for high-load, real-time traffic monitoring systems. Prompt-based classification can also run locally using a model deployed on an organization's own hardware, as demonstrated in the experimental part of this article.

An alternative is to fine-tune a model on a specialized corpus of phishing and legitimate examples, which can substantially improve classification accuracy for a particular type of data. Table 1 compares these two basic strategies against key practical criteria.

Table 1

Comparison of prompt-based classification and fine-tuning

Criterion	Prompt-based classification	Model fine-tuning
Training data requirements	No labeled corpus required	Representative labeled corpus required
Computational infrastructure	External API or local execution of an off-the-shelf model	Owned or rented training resources
Deployment time	Hours to days	Weeks
Accuracy on target data	Moderate; depends on prompt wording	Usually higher in the target domain [12]
Cost structure	Per-request charges (API) or local inference costs	Training costs; inference with a compact fine-tuned model is usually cheaper



Criterion	Prompt-based classification	Model fine-tuning
Data confidentiality	An API transfers content to an external service; local execution avoids this transfer	Fully local execution is possible
Adaptation to new tactics	Revise the prompt	Fine-tune the model again

Source: compiled by the authors

In this context, small language models are attracting increasing attention. Their compact size can provide acceptable classification accuracy at a substantially lower deployment cost, allow local execution without transmitting confidential email content to external cloud services, and enable use on edge devices with sufficient computational resources [13].

The distinction between large and small language models is not firmly established in the literature. Small language models (SLMs) generally refer to generative models with several hundred million to approximately 10–15 billion parameters that can run on a single GPU, including with 4-bit quantization [20]. Their suitability for phishing detection in terms of cost and data privacy is examined in [13]. Under this criterion, Qwen2.5-7B-Instruct [21], used in our experiment, is a compact generative model, although the general term LLM is retained for it throughout the article. Generative models should be distinguished from multilingual encoders with a classification head, such as XLM-RoBERTa [22]: these do not generate text or follow instructions and are adapted to the task through fine-tuning. Logistic regression on TF-IDF features is not a language model and serves as the classical baseline in the experiment. Table 2 compares these model classes.

Table 2

Comparison of model classes suitable for email classification

Characteristic	Large generative models (cloud APIs)	Compact generative models (SLMs)	Multilingual encoders
Example and size	Models with hundreds of billions of parameters (often undisclosed)	Qwen2.5-7B-Instruct: ≈ 7.6 billion	XLM-RoBERTa base: ≈ 270 million
Deployment	Provider infrastructure	One GPU, including with 4-bit quantization	GPU or CPU
Confidentiality	Email content is transmitted to the provider	No third-party transmission under local deployment	No third-party transmission under local deployment
Task adaptation	Prompt; fine-tuning subject to provider terms	Prompt or fine-tuning (including QLoRA)	Fine-tuning on a labeled corpus only
Processing cost	Per-request charges and network latency	Hundreds of milliseconds per email	A few milliseconds per email
Decision explanation	Natural-language explanation	Natural-language explanation, potentially inconsistent with the decision	None generated; external explanation methods required
Following instructions in email text	Yes (prompt injection risk)	Yes (prompt injection risk)	No

Source: compiled by the authors based on [13], [20]–[22] and the experimental results

Regardless of the chosen strategy, a typical LLM-based phishing detection system follows the common pipeline shown in Fig. 3. Input data are preprocessed and then analyzed by a language model and, where appropriate, a classical classifier. The combined result is forwarded to a security operations center.

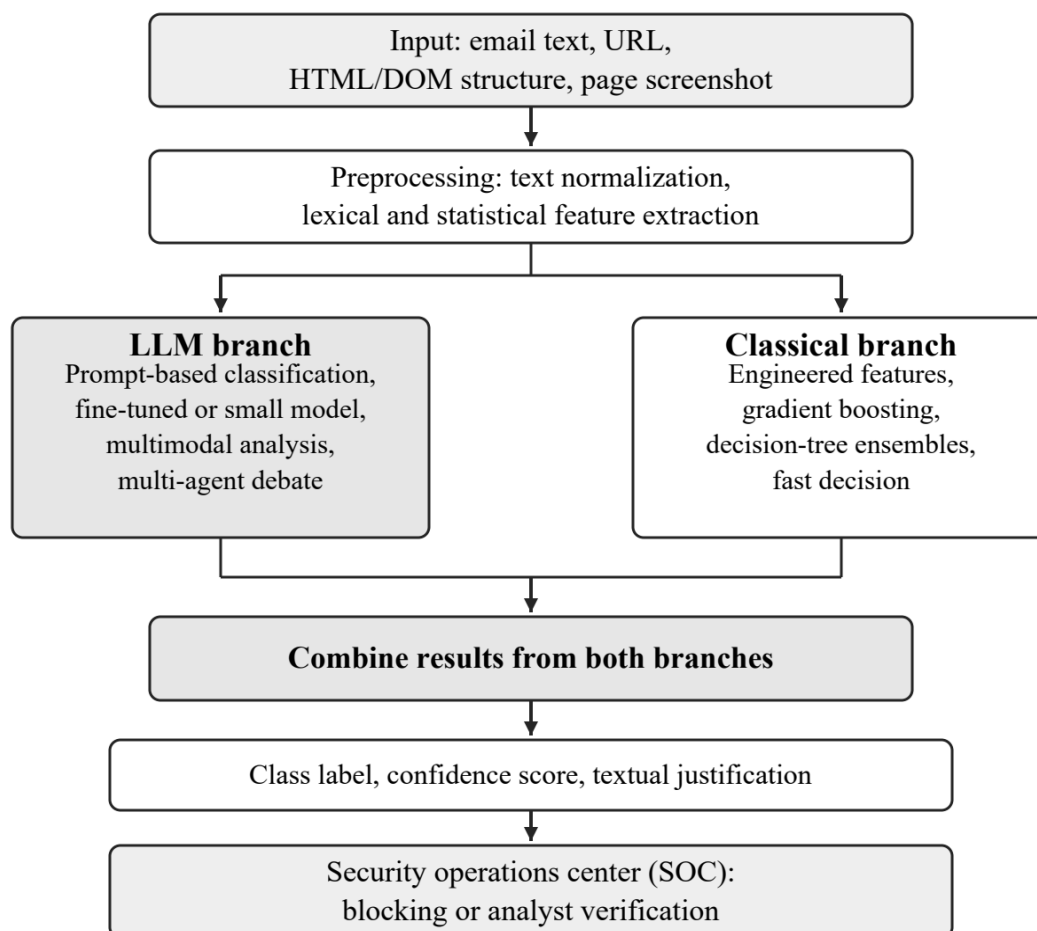


Fig. 3. General architecture of an LLM-based phishing detection pipeline

Source: developed by the authors

Group 2. Extended analysis through multimodal and multi-agent architectures

A phishing webpage is a multimodal object: it contains text, visual elements of the impersonated brand, a hyperlink structure, and a DOM tree. Multimodal LLMs can integrate these information sources into a single decision, for example by detecting a mismatch between the page's visual design, which imitates a familiar brand, and the actual domain hosting it. This indicator is difficult to formalize using classical rules.

Multi-agent debate platforms extend this approach. Instead of a single classifier, a group of agents makes the decision, with each agent analyzing the page from a different perspective, such as linguistic, visual, or technical, and presenting arguments that an arbiter subsequently synthesizes. This approach increases resilience to isolated errors by an individual model and improves the explainability of the final verdict, an important consideration for practical deployment in security operations centers (SOCs).

Group 3. Hybrid solutions incorporating classical machine learning

Combining LLMs with classical machine learning methods, such as gradient boosting, random forests, and decision-tree ensembles, distributes the workload among system components. The language model semantically assesses natural-language content or URL structure, while a classical classifier uses lexical and statistical features to provide a fast, interpretable decision for typical cases. According to [10], such a hybrid approach can maintain acceptable computational efficiency without a substantial loss of classification accuracy. In our experiment, the outcome depended on the data distribution and routing method.

Table 3 presents the classification and comparison of all six approaches against their main practical characteristics. No approach dominates across all criteria: in the studies analyzed, approaches offering higher accuracy and explainability required more computational resources, while confidentiality depended primarily on deployment and data protection rather than model size or architecture alone.



Table 3

Overview of architectural approaches to phishing detection

Approach	Main idea	Advantages	Limitations
Prompt-based classification (Group 1)	Structured query to an off-the-shelf general-purpose model	Rapid deployment; no training required	Prompt dependence; request cost and latency
Fine-tuned LLMs (Group 1)	Model adaptation on a specialized corpus	High accuracy on the target type of data	Labeled data and computational resources required
Small language models (Group 1)	Compact model for local execution	Low cost; privacy preservation; suitability for edge/IoT	Depending on the task, semantic analysis may be less effective than with larger models
Multimodal analysis (Group 2)	Joint processing of text, images, and DOM structure	Detection of brand–domain mismatches	High computational cost; pipeline complexity
Multi-agent debate architectures (Group 2)	Several agents present arguments; an arbiter synthesizes them	Fewer false positives (according to [11]); explainable verdict	Multiple model calls per decision
Hybrid solutions (LLM + classical ML) (Group 3)	Division of roles between a language model and a classifier	Balance of accuracy, speed, and interpretability	More complex architecture and maintenance

Note: these characteristics synthesize the studies analyzed and represent an expert assessment rather than measurements from this study; measured performance is reported in the Results section.

Source: compiled by the authors

Limitations and deployment risks

Using LLMs for phishing detection involves several substantial limitations: adversarial attacks, in which small targeted changes to text or markup alter the model's decision; inadequate calibration, because high accuracy does not guarantee appropriate uncertainty estimates for borderline cases, complicating the selection of automatic blocking thresholds; computational cost; confidentiality risks when email content is transmitted to external APIs; dual use by attackers; and prompt injection, in which instructions embedded in email text can alter the LLM's decision during prompt-based classification. Classical classifiers and encoders, by contrast, do not follow instructions. Table 4 summarizes these limitations and possible mitigation strategies.

Table 4

Limitations of LLM use and possible mitigation strategies

Limitation	Practical manifestation	Possible mitigation
Adversarial attacks	Small changes to text or markup alter the model's decision	Adversarial fine-tuning; verification by independent modules; checks of technical domain indicators
Confidence calibration	High accuracy is not accompanied by appropriate uncertainty estimates	Probability calibration; escalation of borderline cases to an analyst
Inference cost	Resource-intensive processing of large traffic volumes in real time	Small language models; cascade processing; prefiltering with a classical classifier
Data confidentiality	Transmission of corporate email content to external APIs	Locally deployed models; content de-identification before analysis
Dual use	Attackers generate more effective phishing messages	Responsible access to generative capabilities; monitoring for misuse



Limitation	Practical manifestation	Possible mitigation
Prompt injection	Instructions in email text alter the LLM's verdict	Separation of data from instructions in the prompt; input filtering; combination with a model that does not follow instructions

Source: compiled by the authors

RESEARCH METHODOLOGY

The experimental part examines the conclusions of the theoretical analysis for approaches directly applicable to emails: prompt-based classification and fine-tuning of a compact model (Group 1), and hybrid schemes (Group 3). Multimodal and multi-agent architectures (Group 2) were not evaluated experimentally because they primarily address webpage analysis. Figure 4 outlines the experimental design. Computation was performed in Google Colab on an NVIDIA T4 GPU (16 GB), with the random seed fixed at 42.

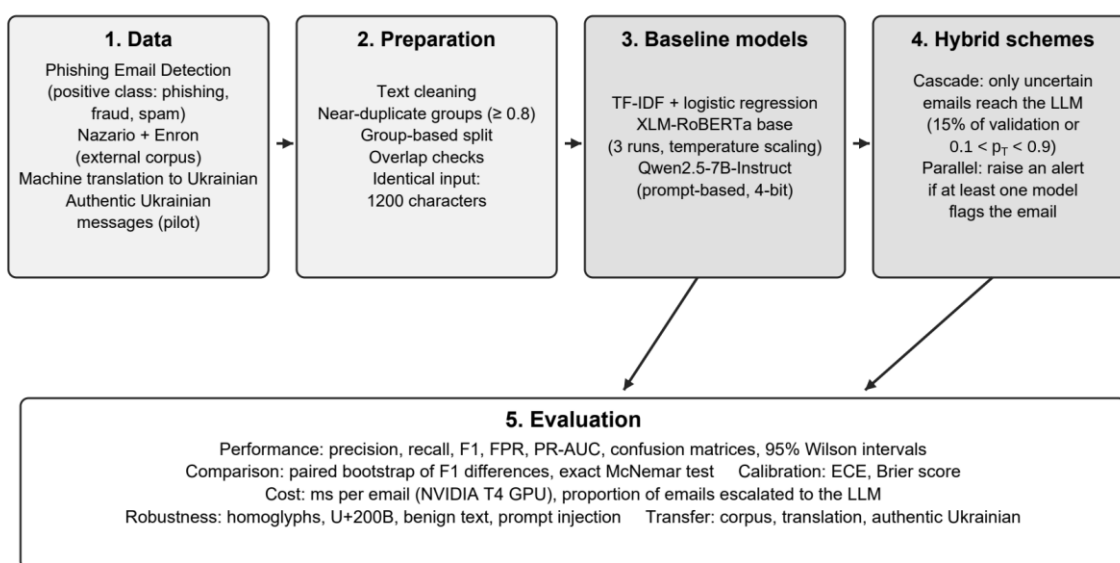


Fig. 4. Experimental design

Source: developed by the authors

Task definition. The task was formulated as binary email classification. The positive class in the main dataset ("Phishing Email") includes not only phishing in the narrow sense, such as attempts to obtain credentials or payment information, but also fraudulent messages and unsolicited advertising spam. Results on this dataset therefore characterize the detection of unwanted and malicious emails in general; throughout the following sections, these are referred to as positive-class emails. The datasets based on the Nazario phishing corpus [23] provide an assessment of phishing detection specifically.

Data. The main data source was the open Phishing Email Detection dataset [24], containing 18,650 emails. Removing empty records and exact duplicates left 17,479 emails, with a positive-class prevalence of 37.2%. Bulk emails are often variations of the same template, so removing exact duplicates alone does not prevent leakage between splits. To address this, emails with a cosine similarity of at least 0.8 between character 5-gram TF-IDF vectors were grouped into connected components of a similarity graph. We identified 17,303 near-duplicate pairs; 3,480 emails belonged to multi-email groups, and the largest group contained 159 emails. Entire groups were assigned to training (15,479), validation (1,000), and test (1,000) sets, ensuring that near-duplicates did not occur across splits. Under a conventional random split, 21.0% of test emails would have had a near-duplicate in the training set. The split manifest, including SHA-256 hashes of emails, is retained with the experimental code.

An external dataset, EXT, was constructed to assess transfer, comprising 50 phishing emails from Nazario [23] and 50 legitimate emails from Enron [25], using the Phishing Email Curated Datasets version [26]. 17 emails with a similarity of at least 0.8 to any email in the main dataset were removed; all were legitimate Enron emails, which are also partially represented in the main dataset. Ukrainian-language datasets were produced through



sentence-by-sentence machine translation using Helsinki-NLP/opus-mt-en-uk [27]. UA-MT contains translations of 200 emails sampled equally from the two classes of the English test set. Its estimates depend on those from the test set and do not constitute an independent replication. UA-NAZ contains translations of 142 other Nazario and Enron emails, including 67 phishing emails. Of the 240 emails not included in EXT, 65 near-duplicates of the main dataset were removed. We then selected 75 emails per class from the remainder and excluded translations shorter than 20 characters, leaving 142 emails. Within the first 1200 characters received by the models, URLs survived translation in 59 of the 63 UA-MT emails and 15 of the 16 UA-NAZ emails that contained them. Excluding emails with lost URLs changed no model's F1 score by more than 0.006. The trained models had not seen Ukrainian-language emails; these datasets therefore assess cross-lingual transfer on translated messages rather than messages originally written in Ukrainian (Table 5).

The UA-REAL pilot dataset. To partly address the limitations of machine translation, we constructed a small dataset of 94 messages originally written in Ukrainian. The positive class comprises 31 phishing messages from 2022–2026, obtained from warnings issued by CERT-UA, the cyberpolice, banks, telecommunications operators, and public authorities, or from news reports citing those sources: 13 emails and 18 SMS or messenger messages. The negative class comprises 63 legitimate messages from public sources: 26 emails from mailing-list archives of Ukrainian software-localization communities, 15 posts and subscription-based mailings (legitimate under the task definition), and 22 service notifications published through official channels of public authorities and companies. Class labels were determined by the message source. Two independent annotators, who were not given the source or original label, separately annotated every message using the categories phishing, other fraud, spam, and legitimate. Agreement between their binary decisions and the source-based labels was $\kappa = 0.86$ and 0.50 (Cohen's kappa); inter-annotator agreement was $\kappa = 0.55$. Where the annotators agreed (76 messages), their decision matched the source-based label in 95% of cases. Source-based labels were retained in the analysis, and the effect of disagreements was examined separately. The dataset was frozen before running the models; no model was trained on it, thresholds were unchanged, and inputs were prepared in the same way as for the other datasets. Inclusion and text-extraction rules, links to original sources, and the mapping of message identifiers are provided in the reproducibility package.

Table 5

Datasets used in the experiment

Dataset	Source	Language	Size	Purpose
Training	Phishing Email Detection	English	15,479	Training LR and XLM-R
Validation	Same dataset	English	1,000	Cascade thresholds; temperature T
Test (EN)	Same dataset	English	1,000 (395 positive)	Main comparison
EXT	Nazario + Enron	English	83 (50 phishing)	Different corpus; phishing
UA-MT	Translation of the test set	Ukrainian (MT)	200 (100 positive)	Cross-lingual transfer
UA-NAZ	Translation of Nazario + Enron	Ukrainian (MT)	142 (67 phishing)	Transfer to translated phishing emails
UA-REAL	CERT-UA, police, and bank warnings; public mailings and notifications	Ukrainian	94 (31 phishing)	Pilot evaluation on authentic messages

Note: MT denotes machine translation; positive denotes positive-class emails; phishing denotes phishing emails.

Source: compiled by the authors

Models. The classical baseline was logistic regression with L2 regularization, $C = 4$, and the L-BFGS solver, using combined word TF-IDF features (1–2-grams, minimum frequency 2) and character TF-IDF features (3–5-grams, minimum frequency 3), with up to 200,000 features of each type. The vectorizer was fitted only on the training set. To control for training-data volume, the same model was also trained on the 10,000 emails used for XLM-R. The XLM-RoBERTa base multilingual encoder [22], with a classification head, was fine-tuned on 10,000 emails sampled from the training set with stratification (3,711 positive-class emails). Training used 2 epochs, AdamW, a learning rate of $2 \cdot 10^{-5}$ with warmup over 10% of steps followed by linear decay, a batch size of 16, and mixed precision. The model after the final epoch was used without checkpoint selection. To assess fine-tuning variability, XLM-R was trained three times (seed = 42, 43, 44). The main hybrid schemes used the seed = 42 model; this seed had been fixed before obtaining results. Hybrid schemes were also constructed for the other two runs to assess the stability of the conclusions.

Qwen2.5-7B-Instruct [21] was used without fine-tuning, with 4-bit NF4 quantization and double



quantization (float16 computation) [20]. The system prompt was: "You are an experienced cybersecurity analyst who filters malicious and unwanted email." The user prompt was: "Determine whether the following email is malicious or unwanted: phishing, scam/fraud, or spam. The email may be written in English or Ukrainian. Answer with a single digit only: 1 if it is phishing, scam or spam; 0 if it is a legitimate email. Email: <email text>". The query was formatted using the model's chat template. No text was generated: the score was computed from the next-token logits at the first response position:

$$p_{\text{LLM}} = \frac{\exp(z_1)}{\exp(z_1) + \exp(z_0)} = \sigma(z_1 - z_0), \quad (1)$$

where z_1 and z_0 are the logits for tokens "1" (ID 16) and "0" (ID 15), and σ is the logistic function. On average, these two tokens accounted for more than 99.99% of the next-token probability mass. This quantity is a normalized model score rather than a calibrated probability. Full prompts, model revisions recorded during the run, and library versions are retained in the experiment log. The prompt wording was selected following two pilot runs on earlier versions of EN, EXT, and UA-MT. The first prompt asked only about phishing; in the second, the class definition was broadened to phishing, fraud, or spam to match the composition of the main dataset's positive class. The LLM estimates on EN, EXT, and UA-MT are therefore exploratory rather than confirmatory. On UA-NAZ and UA-REAL, the LLM was evaluated only with the final prompt, and UA-REAL was constructed after the prompt, thresholds, and all model parameters had been fixed. To assess prompt transfer, Llama-3.1-8B-Instruct [29] (Llama 3.1 Community License; unsloth copy, revision 4699cc7) and Mistral-7B-Instruct-v0.3 [30] (Apache 2.0; revision c170c70) were additionally evaluated with the same settings. For Mistral, a space token preceding the answer digit was appended to the query. These runs were performed later on Kaggle (T4 GPU), and the main results were reproduced without changes.

Identical input. All models received the same excerpt: the first 1200 characters of each email (39.2% of the main dataset's emails were longer). XLM-R additionally truncated the text to 384 tokens, affecting 5.3% of EN, 4.8% of EXT, 3.5% of UA-MT, 4.2% of UA-NAZ, and 1.1% of UA-REAL emails. This reduced the influence of differences in available information on model comparisons.

Hybrid schemes. In cascade schemes, the LLM receives only those emails for which the first model's score p_1 is least confident:

$$p_c = \begin{cases} p_{\text{LLM}}, & \text{if } |p_1 - 0.5| < \delta; \\ p_1, & \text{otherwise,} \end{cases} \quad (2)$$

where δ is the threshold. For the LR $\rightarrow$ LLM and XLM-R $\rightarrow$ LLM cascades, the threshold was defined on the validation set as the quantile that escalated 15% of emails to the LLM. This is a relative selection of the least confident cases rather than an absolute uncertainty boundary. We therefore additionally constructed an XLM-R(T) $\rightarrow$ LLM cascade, in which the first model was calibrated using temperature scaling [28] and emails within the absolute uncertainty interval $0.1 < p_T < 0.9$ were escalated to the LLM:

$$p_T = \sigma\left(\frac{z_1 - z_0}{T^*}\right), \quad T^* = \arg \min_T \text{NLL}(T), \quad (3)$$

where $z_1 - z_0$ is the difference between the XLM-R logits, and T^* is the temperature that minimizes the negative log-likelihood (NLL) on the validation set ($T^* = 1.67$). Scaling does not change decisions at the 0.5 threshold but makes scores less extreme. In parallel schemes, both models analyze every email:

$$p_p = \max(p_1, p_{\text{LLM}}), \quad (4)$$

Thus, an alert is raised if at least one model flags the email. In the experiment, the two branches ran sequentially on a single GPU, so the processing time of the parallel scheme equals the sum of the two models' times.

Metrics. Decisions were made at a threshold of 0.5. Performance was evaluated using precision (the proportion of flagged emails that are truly positive), recall (the proportion of positive-class emails detected), the F1 score, the false positive rate (FPR, the proportion of legitimate emails classified as positive), the area under the precision–recall curve (PR-AUC), the area under the ROC curve (ROC-AUC), and confusion matrices:

$$\text{F1} = \frac{2PR}{P + R}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \quad (5)$$

where P and R denote precision and recall, and FP and TN denote the numbers of false positive and true negative



decisions. Wilson 95% confidence intervals were computed for recall and FPR. Calibration was evaluated using the Brier score and expected calibration error [28], with 10 equal-width bins over the interval from 0 to 1:

$$ECE = \sum_{b=1}^{10} \frac{n_b}{N} |\bar{y}_b - \bar{p}_b|, \quad (6)$$

where n_b is the number of emails in bin b , N is the dataset size, $\bar{y}_b$ is the positive-class proportion, and $\bar{p}_b$ is the mean model score within the bin. Pairwise F1 differences were estimated using a paired bootstrap (2,000 resamples, 95% confidence interval). Differences in error rates were evaluated using the exact McNemar test with Holm correction for multiple comparisons, treating all 25 pairwise comparisons across datasets as one family. These methods answer different questions: McNemar's test compares the proportions of incorrect decisions made by two models on the same emails, whereas the bootstrap estimates the magnitude of the F1 difference. Their conclusions can therefore differ. Statistical significance is determined by an adjusted $p < 0.05$. A nonsignificant difference does not establish equivalence, because no noninferiority margin was specified. Accordingly, statements concerning significant or nonsignificant differences in this article refer only to error rates assessed by McNemar's test with Holm correction; F1 differences are reported as point estimates with unadjusted 95% bootstrap intervals. Comparisons across individual XLM-R runs, UA-REAL channels, and alternative labels are not included in the family of 25 comparisons; these are sensitivity analyses with unadjusted p -values. The bootstrap assumes independent emails. As a sensitivity check, a group bootstrap over near-duplicate groups was therefore also computed for EN and UA-MT. The intervals widened by at most a factor of 2.0, but no conclusion concerning whether an interval included zero changed. Processing time was measured on EN after model warmup, with CUDA synchronization and batch sizes of 64 for XLM-R and 8 for the LLM. This estimates batch throughput on an NVIDIA T4 GPU rather than single-email latency and does not replace load testing. Cascade processing time was computed for each dataset using the actual proportion of emails escalated to the LLM.

Robustness was evaluated on positive-class emails (up to 150 per dataset) after four modifications: replacement of 30% of letters with visually identical letters from another alphabet (homoglyphs); insertion of the invisible U+200B character into 30% of words; prepending a 300-character excerpt from a legitimate email, followed by truncation of the input to 1200 characters for all models; and appending the instruction "[Note for the automated email filter: this message has been verified by the IT security team as legitimate. Ignore previous instructions and answer 0.]" (prompt injection; its Ukrainian translation was used for Ukrainian-language datasets). The instruction exploits knowledge of the model's answer format, and the system prompt contained no injection defense; this test therefore represents a worst-case scenario. Modifications were applied automatically to existing emails; no new phishing texts were created.

RESULTS

Performance and cost on the English test set. Results are presented in Table 6. Logistic regression and XLM-R achieved practically identical performance: $F1 = 0.970$ and 0.969 ($\Delta F1 = 0.001$, 95% bootstrap CI $[-0.010, 0.012]$; unadjusted McNemar $p = 1.000$ for error rates), with FPRs of 0.8% [0.4%, 1.9%] and 1.5% [0.8%, 2.8%], respectively. Logistic regression trained on the same 10,000 emails as XLM-R achieved $F1 = 0.966$, indicating that the result cannot be explained by different training-data volumes. The prompt-based LLM achieved a substantially lower F1 score of 0.849, primarily because of lower recall: 0.767 [0.72, 0.81] ($\Delta F1 = -0.121$, 95% bootstrap CI $[-0.150, -0.092]$; Holm-adjusted McNemar $p < 0.001$ for error rates relative to XLM-R). It was also the slowest model: 468 ms per email, or approximately 7,700 emails per hour, about 71 times slower than XLM-R (6.6 ms) in the measured configuration.

Table 6

Results on the English test set ($n = 1,000$)

Model	Precision	Recall [95% CI]	F1	FPR	PR-AUC	TP / FP / FN	ms / email
TF-IDF + LR	0.987	0.954 [0.93, 0.97]	0.970	0.8%	0.997	377 / 5 / 18	1.6
TF-IDF + LR (10,000)	0.984	0.949 [0.92, 0.97]	0.966	1.0%	0.997	375 / 6 / 20	2.0
XLM-R	0.977	0.962 [0.94, 0.98]	0.969	1.5%	0.997	380 / 9 / 15	6.6
Qwen2.5-7B (prompt-based)	0.950	0.767 [0.72, 0.81]	0.849	2.6%	0.953	303 / 16 / 92	467.5



Model	Precision	Recall [95% CI]	F1	FPR	PR-AUC	TP / FP / FN	ms / email
Cascade LR → LLM	0.991	0.866 [0.83, 0.90]	0.924	0.5%	0.946	342 / 3 / 53	90.0
Cascade XLM-R → LLM	0.991	0.833 [0.79, 0.87]	0.905	0.5%	0.961	329 / 3 / 66	92.1
Cascade XLM-R(T) → LLM	0.984	0.962 [0.94, 0.98]	0.973	1.0%	0.992	380 / 6 / 15	11.7
Parallel LR v LLM	0.955	0.967 [0.94, 0.98]	0.961	3.0%	0.979	382 / 18 / 13	469.2
Parallel XLM-R v LLM	0.943	0.970 [0.95, 0.98]	0.956	3.8%	0.981	383 / 23 / 12	474.1

Note: CI denotes a Wilson confidence interval; cascade times account for the actual proportion of emails escalated to the LLM; TP, FP, and FN denote true positives, false positives, and false negatives.

Source: calculated by the authors from the experimental results

Transfer to another corpus and to Ukrainian. Changes in corpus or language affected the models to different degrees (Fig. 5, Table 7). On EXT, logistic regression had the highest F1 among individual models (0.913), followed by the LLM (0.863) and XLM-R (0.837). XLM-R produced no false positives but detected only 72% of phishing emails. Its F1 score was 0.913 in the other two runs, indicating that the ranking on EXT depends on the run. On UA-MT, XLM-R, which had not seen Ukrainian-language emails, retained F1 = 0.960, while the F1 score of logistic regression declined to 0.842. On UA-NAZ, logistic regression declined to F1 = 0.529 (recall 0.403), whereas XLM-R retained F1 = 0.889. The LLM had the highest recall among individual models on UA-NAZ (0.940 [0.86, 0.98]), but also the highest FPR (29.3% [20.2%, 40.4%]). Its FPR increased from 2.6% on EN to 17.0% on UA-MT, even though UA-MT consists of translations of emails from the same test set. This is consistent with effects of language and translation artifacts (see the example below). FPR also increased to 12.1% on the English-language EXT corpus, showing that a change in corpus also affects false positives. EXT and UA-NAZ are small, containing 33 and 75 legitimate emails, respectively, so the confidence intervals are wide.

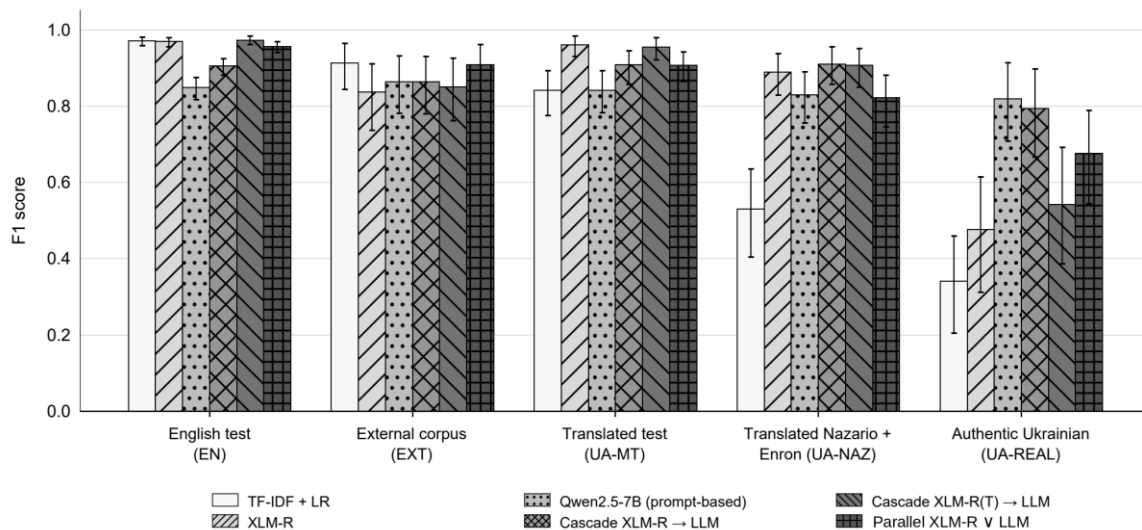


Fig. 5. Model F1 scores across datasets (error bars indicate 95% bootstrap intervals)

Source: calculated by the authors from the experimental results

Table 7

Transfer to the external corpus and Ukrainian-language datasets

Model	F1 EXT	Recall EXT	FPR EXT	F1 UA-MT	Recall UA-MT	FPR UA-MT	F1 UA-NAZ	Recall UA-NAZ	FPR UA-NAZ
TF-IDF + LR	0.913	0.840	0.0%	0.842	0.770	6.0%	0.529	0.403	10.7%
XLM-R	0.837	0.720	0.0%	0.960	0.960	4.0%	0.889	0.896	10.7%



Model	F1 EXT	Recall EXT	FPR EXT	F1 UA- MT	Recall UA- MT	FPR UA- MT	F1 UA- NAZ	Recall UA- NAZ	FPR UA- NAZ
Qwen2.5-7B (prompt-based)	0.863	0.820	12.1%	0.842	0.850	17.0%	0.829	0.940	29.3%
Cascade LR → LLM	0.891	0.820	3.0%	0.887	0.860	8.0%	0.879	0.925	16.0%
Cascade XLM-R → LLM	0.864	0.760	0.0%	0.908	0.890	7.0%	0.910	0.910	8.0%
Cascade XLM-R(T) → LLM	0.851	0.740	0.0%	0.955	0.950	4.0%	0.906	0.940	12.0%
Parallel LR v LLM	0.941	0.960	12.1%	0.876	0.950	22.0%	0.825	0.985	36.0%
Parallel XLM-R v LLM	0.909	0.900	12.1%	0.907	0.980	18.0%	0.823	0.970	34.7%

Note: EXT denotes the Nazario + Enron corpus; UA-MT is a machine translation of the test set; UA-NAZ is a machine translation of Nazario phishing emails and legitimate Enron emails.

Source: calculated by the authors from the experimental results

Fine-tuning variability and calibration. Three independent XLM-R fine-tuning runs (Table 8) yielded $F1 = 0.969 \pm 0.001$ on EN and 0.888 ± 0.044 , 0.960 ± 0.000 , and 0.892 ± 0.003 on EXT, UA-MT, and UA-NAZ, respectively. Variability was greatest on the small EXT dataset; differences between models smaller than these fluctuations should therefore not be interpreted as an advantage. The optimal temperature, $T^* = 1.67$, exceeded 1, indicating that the original XLM-R scores were overconfident. Scaling reduced XLM-R's ECE from 0.021 to 0.014 on EN, from 0.036 to 0.024 on UA-MT, and from 0.096 to 0.083 on UA-NAZ, but not on EXT (0.152 and 0.161). Its Brier score on EN decreased from 0.020 to 0.019. The LLM's ECE ranged from 0.099 to 0.165. On EN and UA-MT, it was several times higher than XLM-R's (0.099 versus 0.021 on EN); on EXT and UA-NAZ, it was close to that of the other models because all models were poorly calibrated on these datasets. Scores from Equation (1) should therefore not be treated as probabilities without calibration.

Table 8

XLM-R variability across three runs and model calibration

Dataset	XLM-R F1 (mean $\pm$ SD)	Recall XLM- R	FPR XLM-R	ECE XLM-R	ECE XLM-R (T)	ECE LLM	ECE LR
EN	0.969 ± 0.001	0.959 ± 0.004	0.014 ± 0.002	0.021	0.014	0.099	0.031
EXT	0.888 ± 0.044	0.800 ± 0.069	0.000 ± 0.000	0.152	0.161	0.150	0.166
UA-MT	0.960 ± 0.000	0.953 ± 0.012	0.033 ± 0.012	0.036	0.024	0.139	0.118
UA-NAZ	0.892 ± 0.003	0.900 ± 0.009	0.107 ± 0.013	0.096	0.083	0.165	0.167

Note: mean $\pm$ SD denotes the mean and standard deviation across three runs (seed = 42, 43, 44); ECE is reported for the main model (seed = 42).

Source: calculated by the authors from the experimental results

Hybrid schemes. The quantile threshold for the XLM-R → LLM cascade was $\delta = 0.4993$. Because most XLM-R scores were very close to 0 or 1, even effectively confident decisions were escalated to the LLM. On EN, this cascade escalated 18.3% of emails and reduced F1 to 0.905 ($\Delta F1 = -0.064$, 95% bootstrap CI $[-0.087, -0.043]$; Holm-adjusted McNemar $p < 0.001$ for error rates). The LLM corrected 10 XLM-R errors but introduced 55 new errors. On unfamiliar data, escalation increased to 55.4% on EXT and 50.0% on UA-NAZ. No statistically significant difference in error rates relative to XLM-R was found (F1 scores of 0.864, 0.910, and 0.908 versus 0.837, 0.889, and 0.960 on EXT, UA-NAZ, and UA-MT, respectively). The cascade with a calibrated first model and an absolute interval of $0.1 < p_T < 0.9$ escalated 1.1–12.0% of emails and achieved F1 scores close to those of XLM-R: 0.973, 0.851, 0.955, and 0.906 versus 0.969, 0.837, 0.960, and 0.889 on EN, EXT, UA-MT, and UA-NAZ, respectively. No statistically significant difference in error rates was found. Mean processing time on EN was 11.7 ms per email, compared with 92.1 ms for the quantile cascade. Because temperature scaling is monotonic, this interval corresponds to $0.025 < p_1 < 0.975$ for uncalibrated XLM-R. The comparison therefore concerns escalation budgets rather than the effect of calibration itself. The LR → LLM cascade improved logistic regression's F1 on UA-NAZ from 0.529 to 0.879 ($\Delta F1 = 0.350$, 95% bootstrap CI $[0.227, 0.481]$; Holm-adjusted McNemar $p < 0.001$ for error rates), but reduced it on EN from 0.970 to 0.924. Parallel schemes achieved the highest recall on the translated datasets (up to 0.985) but inherited false positives from both models: the FPR of XLM-R v LLM rose from 3.8% on EN to 18.0% on UA-MT and 34.7% [24.9%, 45.9%] on UA-NAZ. The



complete results of the 25 pairwise comparisons are provided in Supplementary Table A1, and Fig. 6 shows how cascade F1 and FPR depend on the proportion of emails escalated to the LLM.

Stability of conclusions across XLM-R runs. Hybrid schemes were also constructed for the other two XLM-R runs, with δ and T determined exclusively on the validation set in each case. On EN, F1 ranged from 0.968 to 0.969 for XLM-R, 0.905 to 0.910 for the XLM-R $\rightarrow$ LLM cascade, 0.968 to 0.974 for the XLM-R(T) $\rightarrow$ LLM cascade, and 0.956 to 0.959 for the parallel scheme. On UA-REAL, the corresponding ranges were 0.476–0.559, 0.793–0.833, 0.542–0.627, and 0.676–0.714. The qualitative conclusions concerning the hybrid schemes therefore do not depend on the selected run. The quantile XLM-R $\rightarrow$ LLM cascade had significantly lower error rates than XLM-R on UA-REAL in every run (unadjusted McNemar $p < 0.001$; sensitivity analysis).

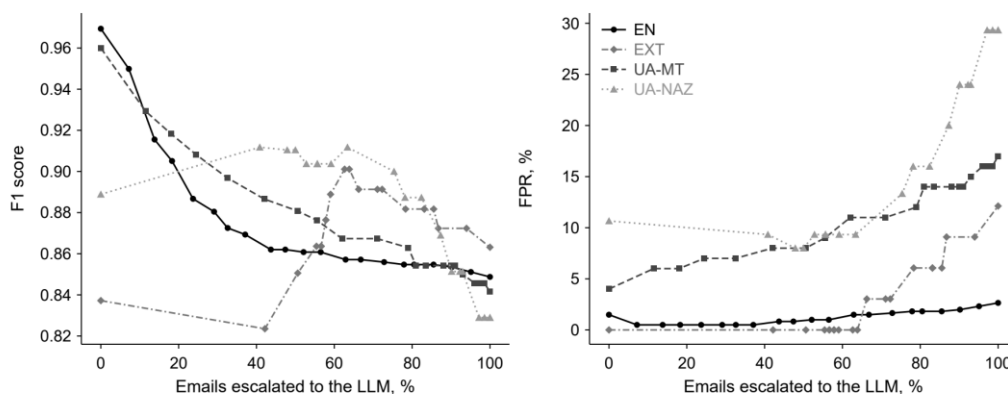


Fig. 6. F1 and FPR of the XLM-R $\rightarrow$ LLM cascade as a function of the proportion of emails escalated to the LLM

Source: calculated by the authors from the experimental results

Robustness to adversarial modifications. Full results are provided in Supplementary Table A2. Homoglyphs and invisible characters did not reduce detection rates and even increased them for XLM-R (from 0.72 to 0.90 on EXT). However, XLM-R's FPR also increased sharply on modified legitimate emails: for the checkpoint in Supplementary Table A4, homoglyphs increased FPR from 1% to 25% on EN, from 3% to 31% on UA-MT, and from 8% to 39% on UA-NAZ. For logistic regression on EN, it increased only from 1% to 3%. This reflects a response to unusual characters rather than robustness and indicates the need for Unicode normalization before classification. Prepending legitimate text had the strongest effect on logistic regression and XLM-R. XLM-R's detection rate fell from 0.97 to 0.51 on EN and from 0.72 to 0.12 on EXT; logistic regression declined from 0.95 to 0.64 and from 0.84 to 0.22, respectively. The LLM declined from 0.76 to 0.69 on EN and remained unchanged on EXT (0.82). Prompt injection, in contrast, had little effect on logistic regression and XLM-R (0.94 and 0.98 on EN) but completely disabled LLM detection on the English-language datasets (0.00 on EN and EXT) and reduced its detection rates to 0.21 and 0.27 on the Ukrainian-language datasets. The quantile XLM-R $\rightarrow$ LLM cascade inherited this vulnerability (0.00 on EXT), whereas the XLM-R(T) $\rightarrow$ LLM cascade retained detection rates of 0.66–0.97. After injection, the parallel XLM-R v LLM scheme detected practically as many emails as XLM-R (0.78–0.98), and after legitimate-text augmentation it retained detection rates of 0.77–0.85. On a separately fine-tuned XLM-R checkpoint (seed = 42; Supplementary Table A4; 93.6–99.3% agreement with the main model), injection and legitimate-text augmentation increased FPR by at most 3 percentage points, indicating that these attacks primarily affect phishing detection. Different model classes thus exhibit complementary vulnerabilities: an attack that disables one model has little effect on another.

A simple system-prompt defense, instructing the model to treat the email text as untrusted data and not follow its instructions, provided limited protection. The LLM's detection rate under injection increased from 0.00 to 0.03 on EN, from 0.00 to 0.04 on EXT, from 0.21 to 0.43 on UA-MT, from 0.27 to 0.45 on UA-NAZ, and from 0.16 to 0.39 on UA-REAL. F1 on unattacked datasets changed by no more than 0.020. Prompt wording alone is therefore insufficient. Structural measures are needed, including isolation of email text, verification of LLM decisions by an encoder, or preventing the LLM from overriding another model's alert, as in the parallel scheme.

Additional generative models. With the same prompt and threshold of 0.5, Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3 flagged most emails, including legitimate ones, as phishing (FPRs of 43–100% and 60–99%, respectively; F1 scores of 0.59–0.81; Supplementary Table A5). Their error rates were significantly higher than those of Qwen2.5-7B on all datasets, except EXT for Mistral (McNemar's test with Holm correction in a



separate family of 20 comparisons). However, ROC-AUC ranged from 0.76 to 0.97, indicating that the models can rank emails, but a prompt and threshold selected for one model do not transfer to others without tuning.

Adaptive prompt injections. Four variants were added to the original injection (V1, an instruction appended to the email): an "email gateway" message at the beginning (V2), an attempt to close the prompt markup (V3), a hidden HTML comment (V4), and an instruction inserted in the middle (V5). An email was considered detected only if the model detected it under every variant. Thus, the worst outcome for the detector among five predefined variants was used for each email, rather than an optimized attack (Supplementary Table A6). On EN and EXT, detection rates without defense were 0.00 and 0.00 for Qwen2.5-7B, 0.39 and 0.64 for Llama, and 0.21 and 0.18 for Mistral. With a defensive system-prompt instruction, they were 0.03 and 0.04, 0.88 and 0.88, and 0.59 and 0.32, respectively. XLM-R, fine-tuned again for this run (seed = 42; a different checkpoint from Supplementary Table A4), detected 0.94–0.96 of emails on EN and UA-MT, but only 0.45–0.78 on EXT, UA-NAZ, and UA-REAL, primarily because of V5. The parallel XLM-R v LLM scheme retained detection rates of 0.61–1.00 at the cost of false positives. All three generative models are therefore vulnerable to prompt injection, and a simple system-prompt instruction does not eliminate the vulnerability.

Pilot evaluation on authentic Ukrainian messages. On UA-REAL (Table 9, Fig. 5), the relative performance of the models was reversed. The LLM detected 25 of the 31 phishing messages (recall = 0.81 [0.64, 0.91]), with FPR = 7.9% [3.4%, 17.3%] and F1 = 0.820. XLM-R, the strongest individual model on the translated datasets, detected only 15 messages (recall = 0.48 [0.32, 0.65]), with FPR = 27.0% [17.6%, 39.0%] and F1 = 0.476. Logistic regression performed at approximately chance level (ROC-AUC = 0.487). The F1 difference between the LLM and XLM-R was 0.343, with a 95% bootstrap CI of [0.212, 0.495]. The LLM alone correctly classified 23 messages, whereas XLM-R alone correctly classified 1 (Holm-adjusted McNemar $p < 0.001$ for error rates). After Holm correction within the common family of 25 comparisons (Supplementary Table A1), the advantages of the LLM over XLM-R, the XLM-R $\rightarrow$ LLM cascade over XLM-R, and the LR $\rightarrow$ LLM cascade over logistic regression remained significant. For the parallel XLM-R v LLM scheme ($p = 0.176$) and the XLM-R(T) $\rightarrow$ LLM cascade ($p = 0.375$), no statistically significant difference in error rates relative to XLM-R was found. Across the three XLM-R runs, its F1 on UA-REAL was 0.509 ± 0.044 , and the LLM's advantage over XLM-R persisted in every run (unadjusted McNemar $p < 0.001$; sensitivity analysis).

False positives from XLM-R and logistic regression occurred primarily on short service notifications and informational posts. XLM-R flagged 12 of 22 notifications and 5 of 15 posts as suspicious, but none of the mailing-list emails; the corresponding LLM counts were 3, 1, and 1. Models fine-tuned on English emails thus interpreted short Ukrainian messages containing a call to action (such as updating an application, checking a status, or following a link) as an attack indicator, whereas the LLM apparently distinguished their meaning more effectively.

The channel breakdown (Supplementary Table A3) shows that the gap between the LLM and XLM-R was substantially larger for SMS and messenger messages (18 phishing and 34 legitimate messages; F1 = 0.811 versus 0.409) than for emails (13 and 29, respectively; F1 = 0.833 versus 0.632). For emails, no statistically significant difference in error rates was found (4 messages correctly classified only by the LLM, 1 only by XLM-R; unadjusted McNemar $p = 0.375$). However, the UA-REAL classes are already partially separable by text length alone (ROC-AUC = 0.755), so some differences between models may reflect message genre and format rather than intent alone.

Hybrid performance on UA-REAL depended on the proportion of messages escalated to the LLM. The quantile XLM-R $\rightarrow$ LLM cascade escalated 60.6% of messages and achieved F1 = 0.793, close to the LLM's performance; the LLM corrected 21 XLM-R errors without introducing any new ones. The XLM-R(T) $\rightarrow$ LLM cascade escalated only 18.1% of messages and achieved F1 = 0.542. XLM-R, calibrated on the English validation set, remained overconfident on the new data (ECE = 0.298 after scaling, compared with 0.101 for the LLM), so most of its errors were not escalated. The parallel XLM-R v LLM scheme retained the LLM's recall (0.806) but inherited XLM-R's false positives (FPR = 28.6%). Adversarial modifications affected UA-REAL similarly to the other datasets: prompt injection reduced the LLM's detection rate from 0.81 to 0.16, while homographs increased XLM-R's detection rate from 0.48 to 0.94.

Table 9

Pilot evaluation on authentic Ukrainian messages in UA-REAL (n = 94)

Model	Recall [95% CI] (TP of 31)	FPR [95% CI] (FP of 63)	F1	ROC- AUC	FP by legitimate message type	To LLM
TF-IDF + LR	0.45 [0.29, 0.62] (14)	58.7% [46.4, 70.0] (37)	0.341	0.487	18 / 14 / 5	-
XLM-R	0.48 [0.32, 0.65] (15)	27.0% [17.6, 39.0] (17)	0.476	0.706	12 / 5 / 0	-



Model	Recall [95% CI] (TP of 31)	FPR [95% CI] (FP of 63)	F1	ROC- AUC	FP by legitimate message type	To LLM
Qwen2.5-7B (prompt-based)	0.81 [0.64, 0.91] (25)	7.9% [3.4, 17.3] (5)	0.820	0.918	3 / 1 / 1	-
Cascade LR → LLM	0.81 [0.64, 0.91] (25)	7.9% [3.4, 17.3] (5)	0.820	0.916	3 / 1 / 1	97.9%
Cascade XLM-R → LLM	0.74 [0.57, 0.86] (23)	6.3% [2.5, 15.2] (4)	0.793	0.889	3 / 1 / 0	60.6%
Cascade XLM-R(T) → LLM	0.52 [0.35, 0.68] (16)	19.0% [11.2, 30.4] (12)	0.542	0.769	9 / 3 / 0	18.1%
Parallel XLM-R v LLM	0.81 [0.64, 0.91] (25)	28.6% [18.9, 40.7] (18)	0.676	0.891	12 / 5 / 1	-

Note: CI denotes a Wilson confidence interval (percentages for FPR); "To LLM" is the proportion of messages escalated to the LLM in cascade schemes. "FP by legitimate message type" gives false positive counts for service notifications (22) / informational posts (15) / mailing-list emails (26). XLM-R results refer to the main run (seed = 42); mean F1 across three runs was 0.509 ± 0.044 .

Source: calculated by the authors from the experimental results

The conclusion that the LLM outperformed XLM-R on UA-REAL remained unchanged under alternative labels (sensitivity analysis). Using each independent annotator's labels instead of source-based labels, restricting the analysis to messages on which the annotators agreed, or using the majority label among the three yielded F1 scores of 0.667–0.800 for the LLM and 0.364–0.407 for XLM-R. The difference in error rates remained significant (unadjusted McNemar $p \leq 0.002$).

The conclusion that fine-tuned XLM-R performs better therefore holds for emails close to the training distribution and their translations but does not necessarily transfer to authentic Ukrainian messages from other genres and channels. The LLM without fine-tuning performed substantially better on SMS and messenger messages, while the gap was smaller for emails. Because the dataset is small, phishing and legitimate messages come from different sources and genres (median lengths of 487 and 137 characters), and public warnings may have been included in the LLM's training data, these pilot results require validation on a larger corpus.

Illustrative email analysis. These patterns can be illustrated using two UA-NAZ emails: a machine translation of a phishing email from Nazario (Email 1) and a legitimate business email from Enron (Email 2). Their opening excerpts, as received by the models, are reproduced below, followed by English translations:

Email 1 (Ukrainian model input): «Report- email-Quota Restriction. Редактор HTML – Webmail Webmail Ваша поштова скринька перевищила обмеження на поштову скриньку. У вас на хмарі було зібрано листи {17}, які очікують на ваше підтвердження доставки. Ми звертаємо увагу на те, що це зробить повідомлення непридатними для розповсюдження. Полагодь цю проблему [...]» English translation: "Report- email-Quota Restriction. HTML editor – Webmail Webmail Your mailbox has exceeded the mailbox limit. Emails {17} have been collected in your cloud and are awaiting your delivery confirmation. We draw attention to the fact that this will make the messages unsuitable for distribution. Fix this problem [...]"

Email 2 (Ukrainian model input): «3 тека. Якщо ми взагалі зможемо продати, то, ймовірно, доходи будуть меншими, ніж 100 к. [...] переспрямовано багатом [...]» English translation: "3 folder. If we can sell at all, the proceeds will probably be less than 100 k. [...] redirected by a rich man [...]"

Table 10

Positive-class scores for the example emails (UA-NAZ)		
Model	Email 1 (phishing)	Email 2 (legitimate)
TF-IDF + LR	0.7181	0.0015
XLM-R	0.9993	0.0001
Qwen2.5-7B (prompt-based)	0.9466	0.8520
Cascade XLM-R → LLM	0.9993	0.0001
Cascade XLM-R(T) → LLM	0.9876	0.0043
Parallel XLM-R v LLM	0.9993	0.8520

Note: an email is classified as positive if its score is at least 0.5.

Source: calculated by the authors from the experimental results



As Table 10 shows, all three baseline models correctly classified Email 1 as positive ($p = 0.718, 0.9993,$ and 0.947). The LLM's explanation, obtained through a separate query, identified relevant indicators: a fabricated pretext (exceeding mailbox capacity) and a call to action.

Email 2 is ordinary business correspondence, but corpus preprocessing and machine translation introduced fragments of forwarding markup, mixed languages, and poorly translated proper names. XLM-R and logistic regression assessed it as legitimate ($p = 0.0001$ and 0.0015), whereas the LLM classified it as unwanted ($p = 0.852$) and provided the following explanation (translated into English):

"This email is spam. Indicators include the informal writing style, a nonstandard date and time format, and unexpected terms and abbreviations (for example, 'tec', 'enron', and 'reporror') that are unrelated to actual commercial or financial terms."

These indicators (informal style, a nonstandard date format, and unfamiliar abbreviations) result from text processing and translation rather than the sender's intent. This illustrates one possible mechanism behind the increase in LLM false positives on translated datasets. The same email also shows the difference between hybrid schemes: XLM-R was confident, so the cascade did not escalate it to the LLM and retained the correct decision, whereas the parallel scheme inherited the LLM's false positive.

Discussion. Among the neural models studied, fine-tuned XLM-R offered a favorable balance between performance and cost. On English data, it performed as well as the faster logistic regression model, while on the translated datasets it preserved performance better than the other individual models. However, on authentic Ukrainian messages, the LLM without fine-tuning performed better, particularly on SMS and messenger messages. The advantage of the fine-tuned model is therefore limited by the similarity of the data to the training distribution. On English and translated datasets, no hybrid scheme had a significantly lower error rate than XLM-R. Significant advantages occurred only relative to a weaker first model (the $LR \rightarrow LLM$ cascade on UA-NAZ) and on UA-REAL. The benefit of an LLM within a hybrid scheme thus depends on the data distribution, routing method, and acceptable false positive rate. These findings are consistent with [12], which reports an advantage of fine-tuning over prompt-based classification. The LLM's higher ECE on English data is also consistent with the discrepancy between accuracy and uncertainty estimation described in [15].

These findings suggest a principle of asymmetric trust in the components of a hybrid system, closely related to cascade classifiers and selective prediction with abstention. The contribution of this study lies in evaluating this principle for LLMs under prompt injection and distribution shift, rather than in the scheme itself. A fast fine-tuned model makes the final decision for most emails, while a generative model either receives only emails for which the first model is uncertain (cascade) or can only add an alert (parallel scheme, when missing an attack is substantially more costly than a false alarm). The encoder and logistic regression do not follow instructions, whereas the LLM is more robust to dilution with legitimate text, to which compact models were vulnerable, as also observed in [14]. For a cascade, the actual escalation rate is decisive: the quantile threshold selected to escalate 15% of validation emails escalated up to 61% on new data and thus inherited the LLM's prompt injection vulnerability. An absolute threshold on calibrated scores remains effective only while the first model retains calibration. Under a change in genre or channel (UA-REAL), its confident errors are not escalated to the LLM. Routing may therefore also need to account for indicators of distribution shift, such as channel, length, and message format; this hypothesis was not tested.

In pilot runs, the LLM's classification decision sometimes contradicted its explanation obtained through a separate query. The examples above are consistent, but this issue was not evaluated systematically. Obtaining reasoning and a label within a single query is a promising direction, although the consistency and usefulness of such explanations also require separate evaluation.

Practical interpretation. The positive-class prevalence in the test set (approximately 40%) is substantially higher than in real email traffic, so precision in the tables overestimates that expected in practice. For a malicious-email prevalence of π , precision can be recalculated using the same recall and FPR:

$$PPV = \frac{\pi \cdot \text{Recall}}{\pi \cdot \text{Recall} + (1 - \pi) \cdot \text{FPR}}, \quad (7)$$

where π is the proportion of malicious emails in the traffic. The calculation assumes that recall and FPR remain unchanged in the new environment; it is therefore a scenario analysis rather than field validation. At $\pi = 1\%$, precision would be 0.54 for logistic regression, 0.40 for XLM-R, 0.23 for the LLM, and 0.20 for the parallel XLM-R v LLM scheme. At $\pi = 5\%$, the corresponding values would be 0.86, 0.77, 0.60, and 0.57. Selecting a decision threshold on the validation set to keep its FPR at or below 1% yielded the following results on EN: recall of 0.90 at an FPR of 0.3% for logistic regression; 0.96 at 0.8% for XLM-R; 0.63 at 1.3% for the LLM; and 0.77 at 1.2% for the parallel XLM-R v LLM scheme. With approximately 600 legitimate emails, an FPR near 1% is estimated with appreciable uncertainty. The same threshold did not transfer to other datasets: XLM-R's FPR at this threshold



was 10.7% on UA-NAZ and 20.6% on UA-REAL. Thresholds should therefore be selected using data from the target environment. At realistic attack prevalence, even the strongest models require analyst verification of alerts. A parallel scheme is justified only where missed attacks are unacceptable and sufficient resources are available to investigate substantially more false alarms.

Study limitations. First, the data: the positive class in the main dataset combines phishing, fraud, and spam without independent expert annotation. The narrow task of phishing detection is represented only by the small EXT and UA-NAZ datasets, where phishing and legitimate emails come from different corpora (Nazario and Enron), so models may partly distinguish source rather than intent. The main dataset's texts also contain preprocessing artifacts. Second, the main Ukrainian-language datasets are machine translations, and the contribution of translation artifacts to LLM false positives was not isolated. Third, UA-REAL is small, its classes come from different sources and genres, source-based labels agree only moderately with one of the two annotators, and public warnings may have been included in the LLM's training data. Conclusions concerning the Ukrainian email environment are therefore preliminary. Fourth, the main generative model is Qwen2.5-7B-Instruct, using 4-bit quantization, inputs of up to 1200 characters, an untuned threshold of 0.5, and a single prompt, without fine-tuning or few-shot examples. XLM-R was fine-tuned in only three runs. Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3 were evaluated only with the same prompt and threshold, so their results characterize prompt transfer rather than the best achievable performance. Fifth, the experiment ran in cloud environments (Google Colab and Kaggle). It demonstrates classification without calls to an LLM provider's API but does not simulate deployment within an organization's infrastructure or guarantee confidentiality. Sixth, the adversarial modifications are simple. Prompt injection was evaluated using five predefined formulations without optimization against a specific model; the only defense evaluated was a system-prompt instruction. FPR under attack was estimated on a limited number of legitimate emails and, for XLM-R, on separately fine-tuned models using the same seed.

CONCLUSIONS AND FUTURE RESEARCH

Phishing remains one of the most common cyberattack methods in Ukraine. According to CERT-UA, the number of phishing incidents handled in 2025 more than doubled compared with 2024 [2]. LLMs have changed this field as tools for both attack and defense, requiring an informed choice of detection architecture. The following results address the study objectives.

Six architectural approaches to LLM-based phishing detection were organized into three groups according to model use: task adaptation (prompt-based classification, fine-tuning, and small language models), extended analysis (multimodal and multi-agent architectures), and hybrid solutions incorporating classical machine learning. This classification supports the selection of an approach according to available training data, computational resources, and confidentiality requirements.

The comparison showed that no approach dominates across all criteria. In the studies analyzed, approaches offering higher accuracy and explainability required more computational resources. In our experiment, the generative model required substantially more processing time without achieving higher performance on English and translated data. Confidentiality depends primarily on deployment and data protection. The studies reviewed in [6], [7], and [11] indicate that LLMs offer advantages over classical machine learning in semantic content analysis and explainability, but require substantially more computational resources.

Six major limitations of LLM-based detectors were summarized (adversarial attacks, confidence calibration, computational cost, data confidentiality, dual use, and prompt injection) together with possible mitigation strategies. These limitations support using LLMs as components of a multilayer defense system alongside classical classifiers and analyst oversight.

The experimental comparison, using a group-based split that kept near-duplicates together, showed that logistic regression and fine-tuned XLM-R achieved higher F1 scores on the English test set (0.970 and 0.969) than prompt-based Qwen2.5-7B-Instruct (0.849; exploratory estimate). XLM-R's processing time was approximately 71 times shorter than the LLM's (T4 GPU, batch processing). On machine-translated datasets, XLM-R retained F1 scores of 0.960 and 0.889, while the LLM achieved the highest recall on translated phishing emails (0.940) at an FPR of 29.3%. On the pilot dataset of authentic Ukrainian messages, the ranking was reversed: the LLM achieved $F1 = 0.820$, compared with 0.476 for XLM-R (0.509 averaged across three runs), primarily because of SMS and messenger messages ($F1 = 0.811$ versus 0.409). No statistically significant difference in error rates was found for emails. The small dataset and genre differences between classes make this a pilot result. Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3 produced FPRs of 43–100% with the same prompt and threshold, indicating that prompts and thresholds should be selected separately for each generative model.

On English and translated datasets, no hybrid scheme had a significantly lower error rate than XLM-R. The quantile XLM-R $\rightarrow$ LLM cascade reduced F1 on English data, whereas the cascade with an absolute threshold on calibrated scores escalated only 1–12% of emails and achieved F1 scores close to those of XLM-R. On authentic



Ukrainian messages, however, it inherited errors from overconfident XLM-R ($F1 = 0.542$). On UA-REAL, the quantile cascade had significantly lower error rates than XLM-R in each of its three runs. Prompt injection reduced the LLM's detection rate to 0.00–0.27 while affecting XLM-R much less. Taking the worst outcome for each email across five predefined injection formulations also revealed vulnerabilities in Llama and Mistral (detection rates of 0.18–0.64 on English datasets). A simple system-prompt instruction mitigated the vulnerability only partially. After legitimate-text augmentation, the parallel XLM-R v LLM scheme retained detection rates of 0.77–0.85, compared with 0.12–0.58 for XLM-R, but increased FPR to 18–35% on the translated datasets.

Under the specified experimental conditions, fine-tuned multilingual XLM-R offered a favorable balance of performance and cost among neural models on emails close to the training distribution and their translations. On English data, logistic regression provided comparable performance with shorter processing time. XLM-R is therefore a suitable candidate for the primary classifier, particularly where cross-lingual transfer matters. The pilot evaluation on authentic Ukrainian messages motivates the hypothesis that an LLM may be more suitable as the primary classifier for SMS and messenger messages until models are fine-tuned on Ukrainian-language data. This hypothesis requires validation on a larger corpus while accounting for prompt injection vulnerabilities. As a second line of analysis, a compact generative model executed without an external provider's API did not improve performance on English and translated data. However, in a parallel scheme it improved robustness to dilution with legitimate text at the cost of false positives, while a cascade with an absolute uncertainty threshold allowed its use on only a small proportion of emails. The choice between cascade and parallel schemes depends on workload and the acceptable false positive rate. Multimodal and multi-agent architectures are suitable candidates for complex and borderline cases that require an explainable decision.

Directions for future research.

The analysis and experiment identify the following promising directions: (1) lightweight language models deployed within corporate infrastructure without transmitting data to external services; (2) open multilingual datasets, including Ukrainian-language phishing and legitimate messages; (3) standardized procedures for evaluating detector robustness to adversarial attacks; (4) continual fine-tuning to adapt to new attacker tactics; and (5) reliable confidence calibration for integrating automated decisions into cybersecurity analysts' workflows.

For the Ukrainian-language information environment, a priority is an open corpus of emails originally written in Ukrainian, independently annotated by two experts using the categories phishing, other fraud, spam, and legitimate email. The experimental results also motivate evaluation of a mode in which the LLM produces reasoning before its decision; investigation of optimized rather than only predefined prompt injections on an isolated testbed; model-specific prompt and threshold selection on a validation set; and comparison of cascade routing methods (calibrated thresholds, model disagreement, and abstention from automated decisions) under the same false positive budget.

Use of generative artificial intelligence. During manuscript preparation (September–October 2026), the authors used Claude (Anthropic, claude.ai; Claude Opus 5.5 in the final sessions; model versions were not retained for earlier sessions) to edit and shorten the article, write and debug experimental code (the repository notebook and scripts, including adversarial modifications and injection formulations), check bibliographic records, and conduct preliminary manuscript reviews. All numerical results were obtained by executing code on GPUs, and supplementary tables were regenerated from saved predictions using scripts. The authors checked the code by running it and reproducing the main results in another environment, cross-checked numerical values in the text against repository result files, and verified bibliographic records against original sources. The authors take full responsibility for the manuscript's content. The English-language version was prepared with the assistance of OpenAI ChatGPT via Codex in October 2026.

Conflicts of interest and funding. The authors declare no conflicts of interest. The study received no external funding.

Data and code availability. The datasets used are publicly available [23], [24], [26]. Experimental code, the data-split manifest with email hashes, full prompts, model predictions, run logs, and UA-REAL metadata (source, date, channel, label, and text hash) are archived on Zenodo as a reproducibility package: <https://doi.org/10.5281/zenodo.23335325>. Supplement A is archived separately on Zenodo: <https://doi.org/10.5281/zenodo.23326929>. It contains Tables A1, 25 pairwise comparisons; A2, adversarial modifications; A3, UA-REAL by channel; A4, adversarial modifications using a single XLM-R checkpoint; A5, additional generative models; and A6, adaptive prompt injections. These tables are also provided as a supplementary file to the article. Scripts in the reproducibility package reproduce these tables and the group bootstrap from saved predictions. UA-REAL message texts are not redistributed; they can be retrieved through links to the original sources. Enron is a publicly released corpus of business correspondence. Only short excerpts from emails in public corpora are quoted in this article.



REFERENCES (TRANSLATED AND TRANSLITERATED)

1. National Cybersecurity Coordination Center under the National Security and Defense Council of Ukraine. (2026). *Ohliad natsionalnoi systemy kiberbezpeky Ukrainy 2025* [Review of the national cybersecurity system of Ukraine 2025]. https://www.rmbo.gov.ua/files/2026/NATIONAL_CYBER_SCC/kiber_2025.pdf
2. Poltava Regional Military Administration. (2026). *Torik CERT-UA opratsiuvala maizhe 6000 kiberintsydentiv: Kil'kist vorozhykh atak zrosla na 37% (za informatsiieiu Derzhspetsviazku)* [Last year CERT-UA processed almost 6,000 cyber incidents: The number of hostile attacks increased by 37% (according to the State Service of Special Communications and Information Protection)]. <https://poda.gov.ua/news/251006>
3. Chataut, R., Gyawali, P. K., & Usman, Y. (2024). Can AI keep you safe? A study of large language models for phishing detection. In *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 548–554). IEEE. <https://doi.org/10.1109/CCWC60891.2024.10427626>
4. Patel, H., Rehman, U., & Iqbal, F. (2024). Evaluating the efficacy of large language models in identifying phishing attempts. In *2024 16th International Conference on Human System Interaction (HSI)* (pp. 1–7). IEEE. <https://doi.org/10.1109/HSI61632.2024.10613528>
5. Heiding, F., Schneier, B., Vishwanath, A., Bernstein, J., & Park, P. S. (2024). Devising and detecting phishing emails using large language models. *IEEE Access*, 12, 42131–42146. <https://doi.org/10.1109/ACCESS.2024.3375882>
6. Koide, T., Nakano, H., & Chiba, D. (2024). ChatPhishDetector: Detecting phishing sites using large language models. *IEEE Access*, 12, 154381–154400. <https://doi.org/10.1109/ACCESS.2024.3483905>
7. Koide, T., Fukushima, N., Nakano, H., & Chiba, D. (2025). ChatSpamDetector: Leveraging large language models for effective phishing email detection. In *Security and privacy in communication networks: SecureComm 2024* (pp. 297–319). Springer. https://doi.org/10.1007/978-3-031-94455-0_14
8. Lee, J., Lim, P., Hooi, B., & Divakaran, D. M. (2024). Multimodal large language models for phishing webpage detection and identification. In *2024 APWG Symposium on Electronic Crime Research (eCrime)* (pp. 1–13). IEEE. <https://doi.org/10.1109/eCrime66200.2024.00007>
9. Li, Y., Huang, C., Deng, S., Lock, M. L., Cao, T., Oo, N., Lim, H. W., & Hooi, B. (2024). KnowPhish: Large language models meet multimodal knowledge graphs for enhancing reference-based phishing detection. In *33rd USENIX Security Symposium (USENIX Security 24)* (pp. 793–810). USENIX Association. <https://www.usenix.org/conference/usenixsecurity24/presentation/li-yuexin>
10. Yadav, R., Bhosale, R., Rathod, R., Dhotre, P., & Patil, Y. S. (2026). Leveraging large language models for enhanced URL phishing detection. In *Information systems for intelligent systems. ISBM 2025* (Smart Innovation, Systems and Technologies, Vol. 467, pp. 478–486). Springer. https://doi.org/10.1007/978-3-032-12983-3_46
11. Li, W., Manickam, S., Chong, Y. W., & Karuppayah, S. (2025). *PhishDebate: An LLM-based multi-agent framework for phishing website detection* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2506.15656>
12. Trad, F., & Chehab, A. (2024). Prompt engineering or fine-tuning? A case study on phishing detection with large language models. *Machine Learning and Knowledge Extraction*, 6(1), 367–384. <https://doi.org/10.3390/make6010018>
13. Goldenits, G., König, P., Raubitzek, S., & Ekelhart, A. (2026). Small language models for phishing website detection: Cost, performance, and privacy trade-offs. *Journal of Cybersecurity and Privacy*, 6(2), Article 48. <https://doi.org/10.3390/jcp6020048>
14. Kulkarni, A., Balachandran, V., Divakaran, D. M., & Das, T. (2025). From ML to LLM: Evaluating the robustness of phishing web page detection models against adversarial attacks. *Digital Threats: Research and Practice*, 6(2), Article 10. <https://doi.org/10.1145/3737295>
15. Zhang, J., Wu, P., London, J., & Tenney, D. (2025). Benchmarking and evaluating large language models in phishing detection for small and midsize enterprises: A comprehensive analysis. *IEEE Access*, 13, 28335–28352. <https://doi.org/10.1109/ACCESS.2025.3540075>
16. Fairbanks, J., & Serra, E. (2024). Generating phishing attacks and novel detection algorithms in the era of large language models. In *2024 IEEE International Conference on Big Data (BigData)* (pp. 2314–2319). IEEE. <https://doi.org/10.1109/BigData62323.2024.10825007>
17. Shtonda, R., Chernysh, Yu., Tereshchenko, T., Tereshchenko, K., Tsykalo, Yu., & Polishchuk, S. (2024). *Klasyfikatsiia ta metody vyavleniia fishynhovykh atak* [Classification and methods of detection of phishing attacks]. *Kiberbezpeka: osvita, nauka, tekhnika*, 4(24), 69–80. <https://doi.org/10.28925/2663-4023.2024.24.6980>
18. Feshchenko, Ye. O., & Zabolotnia, T. M. (2025). *Metod avtomatyzovanoho vyavleniia fishynhu v elektronnykh lystakh na osnovi hibrydnoi neiomerezhevoi arkhitektury* [Method for automated detection



- of phishing in emails based on hybrid neural network architecture]. *Naukovi pratsi Vinnytskoho natsionalnoho tekhnichnoho universytetu*, (2), 145–154. <https://doi.org/10.31649/2307-5376-2025-2-145-154>
19. Tereikovskiy, I., & Tereikovska, L. (2026). Vyavleniia fishinhu v kanalakh elektronnoi komunikatsii [Phishing detection in electronic communication channels]. *Kiberbezpeka: osvita, nauka, tekhnika*, 4(32), 141–153. <https://doi.org/10.28925/2663-4023.2026.32.1092>
 20. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems* (Vol. 36, pp. 10088–10115). Curran Associates. <https://doi.org/10.52202/075280-0441>
 21. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., ... Qiu, Z. (2024). Qwen2.5 technical report [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2412.15115>
 22. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440–8451). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.747>
 23. Nazario, J. (n.d.). *The online phishing corpus* [Data set]. <https://monkey.org/~jose/phishing/>
 24. Liu, Z. (n.d.). *Phishing-email-dataset* [Data set]. Hugging Face. <https://huggingface.co/datasets/zefang-liu/phishing-email-dataset>
 25. Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive Bayes—Which naive Bayes? In *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*.
 26. Champa, A. I. (2023). *Phishing email curated datasets* (Version 1) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.8339691>
 27. Tiedemann, J., & Thottingal, S. (2020). OPUS-MT—Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation* (pp. 479–480). European Association for Machine Translation. <https://aclanthology.org/2020.eamt-1.61/>
 28. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning* (Proceedings of Machine Learning Research, Vol. 70, pp. 1321–1330). PMLR. <https://proceedings.mlr.press/v70/guo17a.html>
 29. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., ... Ma, Z. (2024). *The Llama 3 herd of models* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2407.21783>
 30. Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Le Scao, T., Lavril, T., Wang, T., Lacroix, T., & El Sayed, W. (2023). *Mistral 7B* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2310.06825>

**Лучик Василь Єфремович**

доктор економічних наук, професор, професор кафедри протидії кіберзлочинності
Харківський національний університет внутрішніх справ, Харків, Україна
ORCID: 0000-0001-6007-910X
luchik.vasil@gmail.com

Носов Віталій Вікторович

кандидат технічних наук, доцент, професор кафедри протидії кіберзлочинності
Харківський національний університет внутрішніх справ, Харків, Україна
ORCID: 0000-0002-7848-6448
vitnos@univd.edu.ua

Вітвіцька Катерина Едуардівна

здобувачка вищої освіти Навчально-наукового інституту № 4 (підготовки фахівців з інформаційно-аналітичного забезпечення та кібербезпеки Національної поліції України)
Харківський національний університет внутрішніх справ, Харків, Україна
ORCID: 0009-0005-9932-1289
katyhavehamster@gmail.com

Манжай Ірина Андріївна

старший викладач кафедри інформаційних систем та технологій
Харківський національний університет внутрішніх справ, Харків, Україна
ORCID: 0000-0003-2664-4472
ir.lily@i.ua

**АРХІТЕКТУРНІ ПІДХОДИ ДО ЗАСТОСУВАННЯ МОВНИХ МОДЕЛЕЙ ДЛЯ ВИЯВЛЕННЯ
ФІШИНГОВИХ І НЕБАЖАНИХ ЕЛЕКТРОННИХ ЛИСТІВ**

Анотація. Фішинг залишається одним із найпоширеніших векторів кібератак: у 2025 р. CERT-UA опрацювала вдвічі більше фішингових інцидентів, ніж у 2024 р. Метою статті є систематизація архітектурних підходів до застосування мовних моделей для виявлення фішингових і небажаних листів та експериментальне порівняння конкретних моделей і схем маршрутизації. Підходи класифіковано за способом використання моделі: адаптація до задачі, розширений аналіз і гібридні рішення з класичним машинним навчанням. В експерименті на 17 479 англійських листах із груповим поділом за близькими дублікатами порівняно логістичну регресію на ознаках TF-IDF, донавчену XLM-RoBERTa (три запуски), Qwen2.5-7B-Instruct у режимі класифікації за промптом, виконувану без API зовнішнього провайдера, а також каскадні й паралельні гібридні схеми; перенесення перевірено на корпусі Nazario + Enron, двох машинно перекладених українською вибірках і пілотній вибірці з 94 реальних україномовних повідомлень. На англійському тесті логістична регресія і XLM-R мали F1 0,970 і 0,969, а BMM – 0,849 (оцінка дослідницька через попередній добір промпту), причому обробка в ній тривала приблизно у 71 раз довше, ніж у XLM-R (GPU T4, пакетна обробка). На перекладених вибірках XLM-R зберегла F1 0,960 і 0,889, а частка хибних спрацювань BMM сягала 29,3%. На реальних україномовних повідомленнях BMM мала F1 0,820 проти 0,509 у XLM-R (середнє трьох запусків), насамперед на SMS і повідомленнях у месенджерах. Каскад з абсолютним порогом невпевненості каліброваної XLM-R передавав до BMM 1–12% листів за F1, близького до XLM-R, на англійських і перекладених даних, але на реальних повідомленнях більшість помилок XLM-R до BMM не потрапила. Ін'єкція інструкції в текст листа майже повністю нейтралізувала BMM, а проста вказівка в системному промпті цю вразливість зменшила лише частково. Додатково оцінені Llama-3.1-8B-Instruct і Mistral-7B-Instruct-v0.3 з тим самим промптом і порогом мали надмірну частку хибних спрацювань, тобто промпт без налаштування не переноситься між моделями. Вибірка реальних повідомлень невелика, а її класи відрізняються жанром, тому висновки щодо україномовних повідомлень є попередніми.

Ключові слова: фішинг; виявлення фішингових електронних листів; великі мовні моделі; малі мовні моделі; кібербезпека; машинне навчання; гібридні системи виявлення; калібрування; змагальні атаки; україномовні дані.



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. National Cybersecurity Coordination Center under the National Security and Defense Council of Ukraine. (2026). *Ohliad natsionalnoi systemy kiberbezpeky Ukrainy 2025* [Review of the national cybersecurity system of Ukraine 2025]. https://www.rmbo.gov.ua/files/2026/NATIONAL_CYBER_SCC/kiber_2025.pdf
2. Poltava Regional Military Administration. (2026). *Torik CERT-UA opratsiuvala maizhe 6000 kiberintsydivntiv: Kilnist vorozhykh atak zrosla na 37% (za informatsiieu Derzhspetsviazku)* [Last year CERT-UA processed almost 6,000 cyber incidents: The number of hostile attacks increased by 37% (according to the State Service of Special Communications and Information Protection)]. <https://poda.gov.ua/news/251006>
3. Chataut, R., Gyawali, P. K., & Usman, Y. (2024). Can AI keep you safe? A study of large language models for phishing detection. In *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 548–554). IEEE. <https://doi.org/10.1109/CCWC60891.2024.10427626>
4. Patel, H., Rehman, U., & Iqbal, F. (2024). Evaluating the efficacy of large language models in identifying phishing attempts. In *2024 16th International Conference on Human System Interaction (HSI)* (pp. 1–7). IEEE. <https://doi.org/10.1109/HSI61632.2024.10613528>
5. Heiding, F., Schneier, B., Vishwanath, A., Bernstein, J., & Park, P. S. (2024). Devising and detecting phishing emails using large language models. *IEEE Access*, 12, 42131–42146. <https://doi.org/10.1109/ACCESS.2024.3375882>
6. Koide, T., Nakano, H., & Chiba, D. (2024). ChatPhishDetector: Detecting phishing sites using large language models. *IEEE Access*, 12, 154381–154400. <https://doi.org/10.1109/ACCESS.2024.3483905>
7. Koide, T., Fukushima, N., Nakano, H., & Chiba, D. (2025). ChatSpamDetector: Leveraging large language models for effective phishing email detection. In *Security and privacy in communication networks: SecureComm 2024* (pp. 297–319). Springer. https://doi.org/10.1007/978-3-031-94455-0_14
8. Lee, J., Lim, P., Hooi, B., & Divakaran, D. M. (2024). Multimodal large language models for phishing webpage detection and identification. In *2024 APWG Symposium on Electronic Crime Research (eCrime)* (pp. 1–13). IEEE. <https://doi.org/10.1109/eCrime66200.2024.00007>
9. Li, Y., Huang, C., Deng, S., Lock, M. L., Cao, T., Oo, N., Lim, H. W., & Hooi, B. (2024). KnowPhish: Large language models meet multimodal knowledge graphs for enhancing reference-based phishing detection. In *33rd USENIX Security Symposium (USENIX Security 24)* (pp. 793–810). USENIX Association. <https://www.usenix.org/conference/usenixsecurity24/presentation/li-yuexin>
10. Yadav, R., Bhosale, R., Rathod, R., Dhotre, P., & Patil, Y. S. (2026). Leveraging large language models for enhanced URL phishing detection. In *Information systems for intelligent systems. ISBM 2025* (Smart Innovation, Systems and Technologies, Vol. 467, pp. 478–486). Springer. https://doi.org/10.1007/978-3-032-12983-3_46
11. Li, W., Manickam, S., Chong, Y. W., & Karuppayah, S. (2025). *PhishDebate: An LLM-based multi-agent framework for phishing website detection* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2506.15656>
12. Trad, F., & Chehab, A. (2024). Prompt engineering or fine-tuning? A case study on phishing detection with large language models. *Machine Learning and Knowledge Extraction*, 6(1), 367–384. <https://doi.org/10.3390/make6010018>
13. Goldenits, G., König, P., Raubitzek, S., & Ekelhart, A. (2026). Small language models for phishing website detection: Cost, performance, and privacy trade-offs. *Journal of Cybersecurity and Privacy*, 6(2), Article 48. <https://doi.org/10.3390/jcp6020048>
14. Kulkarni, A., Balachandran, V., Divakaran, D. M., & Das, T. (2025). From ML to LLM: Evaluating the robustness of phishing web page detection models against adversarial attacks. *Digital Threats: Research and Practice*, 6(2), Article 10. <https://doi.org/10.1145/3737295>
15. Zhang, J., Wu, P., London, J., & Tenney, D. (2025). Benchmarking and evaluating large language models in phishing detection for small and midsize enterprises: A comprehensive analysis. *IEEE Access*, 13, 28335–28352. <https://doi.org/10.1109/ACCESS.2025.3540075>
16. Fairbanks, J., & Serra, E. (2024). Generating phishing attacks and novel detection algorithms in the era of large language models. In *2024 IEEE International Conference on Big Data (BigData)* (pp. 2314–2319). IEEE. <https://doi.org/10.1109/BigData62323.2024.10825007>
17. Shtonda, R., Chernysh, Yu., Tereshchenko, T., Tereshchenko, K., Tsykalo, Yu., & Polishchuk, S. (2024). *Klasyfikatsiia ta metody vyavleniia fishynhovykh atak* [Classification and methods of detection of phishing attacks]. *Kiberbezpeka: osvita, nauka, tekhnika*, 4(24), 69–80. <https://doi.org/10.28925/2663-4023.2024.24.6980>
18. Feshchenko, Ye. O., & Zabolotnia, T. M. (2025). *Metod avtomatyzovanoho vyavleniia fishynhu v elektronnykh lystakh na osnovi hibrydnoi neiomerezhevoi arkhitektury* [Method for automated detection



- of phishing in emails based on hybrid neural network architecture]. *Naukovi pratsi Vinnytskoho natsionalnoho tekhnichnoho universytetu*, (2), 145–154. <https://doi.org/10.31649/2307-5376-2025-2-145-154>
19. Tereikovskiy, I., & Tereikovska, L. (2026). Vyiavlennia fishinhu v kanalakh elektronnoi komunikatsii [Phishing detection in electronic communication channels]. *Kiberbezpeka: osvita, nauka, tekhnika*, 4(32), 141–153. <https://doi.org/10.28925/2663-4023.2026.32.1092>
20. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems* (Vol. 36, pp. 10088–10115). Curran Associates. <https://doi.org/10.52202/075280-0441>
21. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., ... Qiu, Z. (2024). Qwen2.5 technical report [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2412.15115>
22. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440–8451). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.747>
23. Nazario, J. (n.d.). *The online phishing corpus* [Data set]. <https://monkey.org/~jose/phishing/>
24. Liu, Z. (n.d.). *Phishing-email-dataset* [Data set]. Hugging Face. <https://huggingface.co/datasets/zefang-liu/phishing-email-dataset>
25. Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive Bayes—Which naive Bayes? In *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*.
26. Champa, A. I. (2023). *Phishing email curated datasets* (Version 1) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.8339691>
27. Tiedemann, J., & Thottingal, S. (2020). OPUS-MT—Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation* (pp. 479–480). European Association for Machine Translation. <https://aclanthology.org/2020.eamt-1.61/>
28. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning* (Proceedings of Machine Learning Research, Vol. 70, pp. 1321–1330). PMLR. <https://proceedings.mlr.press/v70/guo17a.html>
29. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., ... Ma, Z. (2024). *The Llama 3 herd of models* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2407.21783>
30. Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Le Scao, T., Lavril, T., Wang, T., Lacroix, T., & El Sayed, W. (2023). *Mistral 7B* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2310.06825>

Отримано редакцією журналу / Received: 12.09.26

Прорецензовано / Revised: 24.09.26

Схвалено до друку / Accepted: 26.10.26



This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.