(Online) ISSN 2744-1741
Original Research

Defense and Security Studies
Vol. 7, No. 2, 2026, pp.220-236
https://doi.org/10.37868/dss.v7.id344

StageGuard: A hierarchical BERT-BiLSTM framework with stage transition anomaly scoring for fraudulent conversation detection

Karrar M. Khudhair[1], Bareq M. Khudhair[2*]
[1,2] Department of Computer Techniques Engineering, Imam Al-Kadhim University College (IKC), Iraq

*Corresponding author E-mail: bareq.maher@iku.edu.iq

Received: Jul 14., 2026
Revised: Aug 1., 2026
Accepted: Aug 4., 2026
Online: Aug 5., 2026

Abstract

Fraud conducted through chat applications has grown into a serious global threat affecting millions of people each year. Most detection systems still classify each message in isolation rather than reasoning over the conversation as a whole, which limits their ability to recognise the gradual manipulation that characterises social-engineering fraud. This paper presents StageGuard, a hierarchical deep-learning architecture that models the full trajectory of a fraudulent conversation. The framework builds on the Manipulation Stage Progression Model (MSPM), a five-stage taxonomy grounded in Cialdini's principles of psychological influence [4], in which offenders build trust gradually across turns before exploiting it. On this basis we define the Stage Transition Anomaly Score (STAS), a single-valued, human-readable feature that quantifies how abruptly a conversation moves between stages and serves as an interpretable indicator of manipulative intent. BERT-base-uncased encodes each turn, and the resulting turn embeddings are processed by a bidirectional LSTM [6], [19] with Bahdanau attention [7] that captures temporal dependencies across the conversation. The attention context vector is fused with the STAS and urgency features before a multi-layer classifier produces the fraud decision, and training uses a multi-task objective that jointly optimises fraud detection and per-turn stage classification. Under a leakage-free protocol in which messages are partitioned into disjoint train, validation and test pools before conversations are constructed, and on a realistic task in which both classes contain spam so that content alone is insufficient, StageGuard attains an F1-macro of 0.9919 and an ROC-AUC of 0.9998. The hierarchical design substantially outperforms flat baselines that treat the conversation as a single text (F1 up to 0.952); ablation and McNemar tests indicate that STAS does not change accuracy significantly, and its value is instead the per-turn interpretability it provides for human operators. We further report bootstrap confidence intervals, five-fold cross-validation and an external-domain evaluation that exposes a sharp generalisation gap, which we discuss candidly as a limitation.

Keywords: fraud detection; conversational AI; BERT; BiLSTM; attention mechanism; manipulation stages; STAS; multi-task learning; natural language processing

1. Introduction

Consider a person who receives a message from an unknown contact who appears warm, helpful and attentive. Over several days the exchange becomes increasingly personal, trust develops, and then a request arrives - for money, sensitive details, or an urgent bank transfer. This pattern characterises romance scams, advance-fee fraud and phishing, and prior work has shown that such attacks follow a script that is remarkably consistent across cultures, resembling a predictable narrative [1]. Most automated detectors, however, are not designed to read narratives; they classify individual sentences. Although spam and phishing classifiers have improved markedly over the past decade, reaching F1 scores above 0.97 on standard benchmarks [2, 3], they typically decide from a single message or a short fixed context window. Experienced offenders exploit this: romance fraud opens with ordinary social messages that a single-message classifier cannot distinguish from benign conversation, and the manipulation becomes apparent only when the exchange is examined over time, from the initial greeting to the eventual exploitation. This temporal gap is the focus of our work. We present StageGuard, a system that reasons over the entire conversation and, rather than asking whether an individual message is suspicious, estimates whether the conversation as a whole is progressing toward fraud.

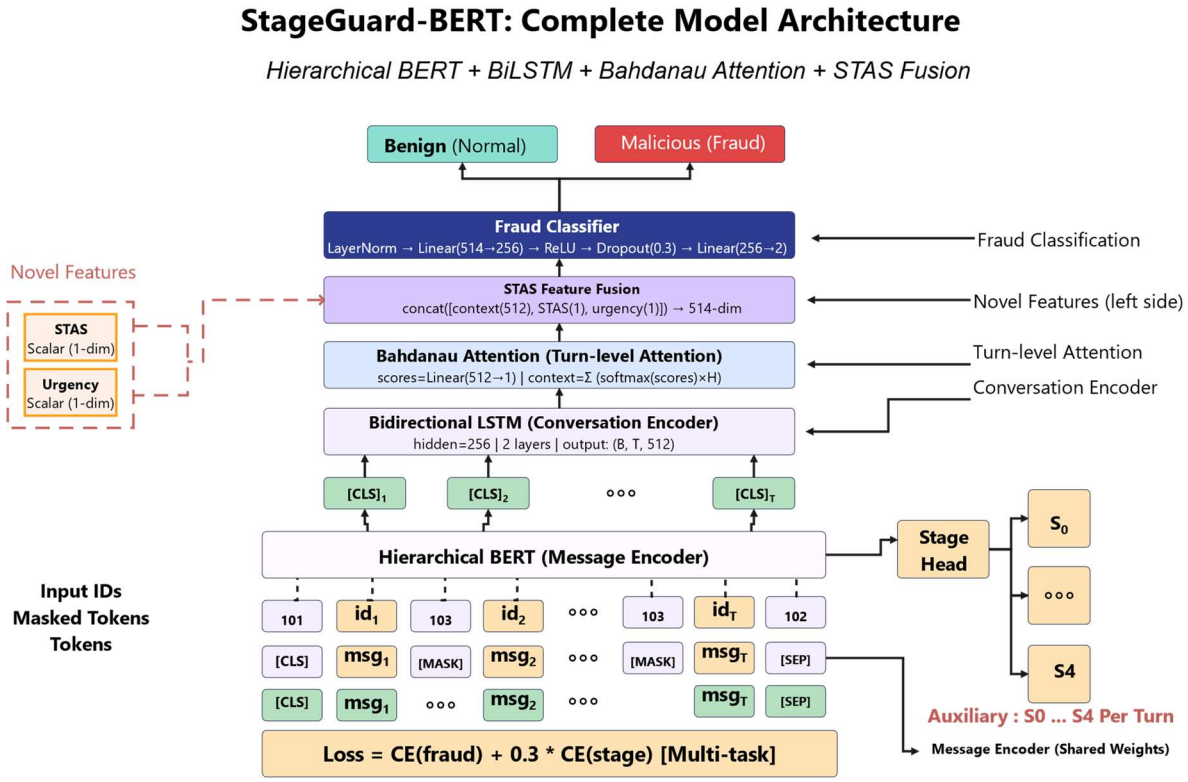

Figure 1. Complete StageGuard-BERT architecture. Each turn of the dialogue is encoded independently by BERT; the resulting sequence is processed by a bidirectional LSTM with Bahdanau attention, after which the STAS feature is fused before the final fraud classifier. An auxiliary stage head provides structured supervision during training.

StageGuard combines two components. First, the Manipulation Stage Progression Model (MSPM) defines five stages - Initial Contact, Trust Building, Opportunity Creation, Pressure, and Execution - that approximate the psychological arc of a scam and draw on Cialdini's six principles of persuasion [4]. Second, the Stage Transition Anomaly Score (STAS) is a single value that captures how abruptly a dialogue moves between these stages; benign conversations tend to progress gradually, whereas fraudulent ones often jump across several stages within a single turn to rush the victim. These components are embedded in a hierarchical neural architecture: BERT [5] encodes each message, a bidirectional LSTM [6] models the sequence of encoded turns, Bahdanau attention [7] highlights the most influential turns, and STAS is fused into the final decision layer. Training uses a multi-task objective that predicts both the conversation-level fraud label and the per-turn manipulation stage.

Evaluated across seven models under a leakage-free protocol on a realistic task, the hierarchical StageGuard architecture reaches an F1-macro of 0.9919 and an ROC-AUC of 0.9998, clearly surpassing flat models that treat the conversation as a single block of text. We find that the hierarchical, turn-level modelling - rather than the STAS feature itself - drives the performance gain; STAS does not significantly change accuracy but provides an interpretable, per-turn signal of when and how a conversation begins to deviate, which the baseline variants cannot produce.

2. Related work

2.1. Spam and phishing message detection

The UCI SMS Spam Collection [8], introduced by Almeida et al., is a widely used benchmark for text spam detection. Standard methods based on TF-IDF representations with Naive Bayes, SVM or Random Forest classifiers report F1 scores in the range of roughly 0.88 to 0.96 on this dataset. Mishra and Soni [9] extended this line of work to phishing SMS and observed that keyword-only signals generalise poorly, because attackers continually change their wording over time. Consequently, research has shifted from hand-crafted features toward semantic encodings. Recurrent architectures such as bidirectional LSTMs [6], with word- or character-level embeddings, outperform bag-of-words baselines by modelling token order and local context within a message. More recent studies apply transformer encoders and large language models to smishing and phishing detection, including multi-class formulations that separate normal, promotional and smishing messages [25] and explainable transformer detectors for phishing email [22]. These approaches nonetheless share the assumption that the unit of detection is a single message, and none model the conversational arc that precedes a fraudulent request.

2.2. BERT and pre-trained language models

Devlin et al. [5] introduced BERT, demonstrating that pre-training a deep bidirectional transformer on large text corpora and fine-tuning on downstream tasks yields state-of-the-art results across eleven NLP benchmarks. Sun et al. [10] studied how to fine-tune BERT for text classification specifically, finding that learning-rate warmup and layer-wise decay are critical for convergence. Subsequent work on fraud and phishing detection adapted BERT to email and SMS corpora and consistently reported gains of around 2–5 percentage points in F1 over recurrent baselines, though results varied by corpus.

The transformer self-attention mechanism [11] has also been applied at the conversation level, where each token in a flattened transcript attends to every other token. CNN-based text classifiers [17] and hierarchical models [18] have been applied to document-level tasks but do not perform turn-by-turn sequence modelling. Flattened transformer approaches also struggle with long conversations, because the quadratic cost of self-attention forces aggressive truncation. Hierarchical encoders that first process individual utterances and then model the resulting sequence have been proposed for dialogue summarisation and intent detection, but, to our knowledge, not for fraud detection with an explicit manipulation-stage supervision signal.

2.3. Conversational and social engineering fraud

Edwards et al. [12] conducted a qualitative analysis of social engineering attacks and found that nearly all successful scams follow a trust-building phase before the deceptive request, consistent with Cialdini's framework [4]. Herley [13] examined why Nigerian advance-fee emails are so obviously implausible, arguing that the implausibility itself serves as a filter to select victims who are maximally susceptible. These observations point toward the view that fraud has a detectable structural grammar - one that, in principle, can be modelled computationally.

Recent work has begun to address fraud at the conversational level using large language models: benchmarks assess the robustness of LLMs to multi-round fraud and phishing inducements [23], LLM-generated phishing content has been studied together with BERT-based detectors [26], and real-time LLM approaches have been

proposed for phone-scam detection with immediate user warnings [24]. Building on these directions, StageGuard contributes a formalisation of the manipulation arc as a five-stage progression model, a quantitative anomaly score over stage transitions, and a hierarchical architecture trained with multi-task stage supervision. To the best of our knowledge, the combination of these three elements for conversation-level fraud detection has not previously been reported, although we make no claim of being the first to model conversational fraud in general.

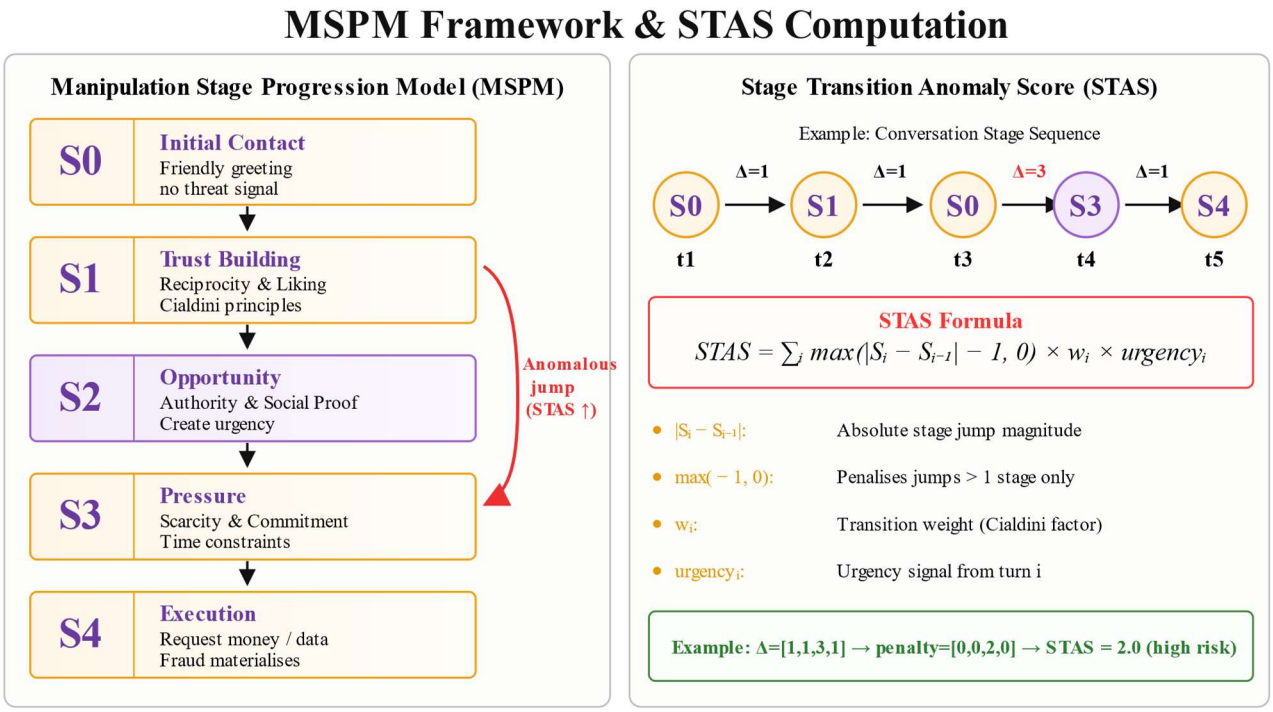

Figure 2. Left: The five stages of the Manipulation Stage Progression Model (MSPM), grounded in Cialdini's influence principles. Right: STAS computation over an example conversation. A jump from S0 directly to S3 ($\Delta = 3$) incurs a penalty of 2 in the score, signalling abnormal manipulation acceleration.

3. System architecture

3.1. Overview

StageGuard represents a conversation as an ordered sequence of T turns, each turn being a text string contributed by a participant. The system produces two outputs simultaneously: a binary fraud label (Normal / Fraud) for the conversation as a whole, and a stage label from {S0, S1, S2, S3, S4} for each individual turn. The complete model comprises five components, detailed in Sections 3.2–3.8, and is trained end to end as a single pipeline.

3.2. Manipulation stage progression model (MSPM)

Each stage is associated with a set of lexical and behavioural cues derived from Cialdini's principles [4]. Trust-building turns exhibit the reciprocity and liking principles through the sharing of personal details and expressions of affection, whereas later stages introduce authority, scarcity and commitment cues that precede the request for money or sensitive information. The MSPM assigns each turn to the stage whose cues it most strongly matches, producing the per-turn stage sequence used by both the STAS computation and the auxiliary supervision signal.

3.3. BERT message encoder

Each turn t_i is tokenised and passed through BERT-base-uncased (12 layers, 768 hidden dimensions, 12 attention heads). We extract the [CLS] pooler output as a 768-dimensional representation of the turn's semantics. The same BERT instance, with shared weights, processes all T turns in a single batched forward pass (B×T, L) →

(B×T, 768), where B is the batch size and L is the maximum token length per turn (128 tokens). To reduce training time, the first six transformer layers are frozen; only the upper six layers are fine-tuned.

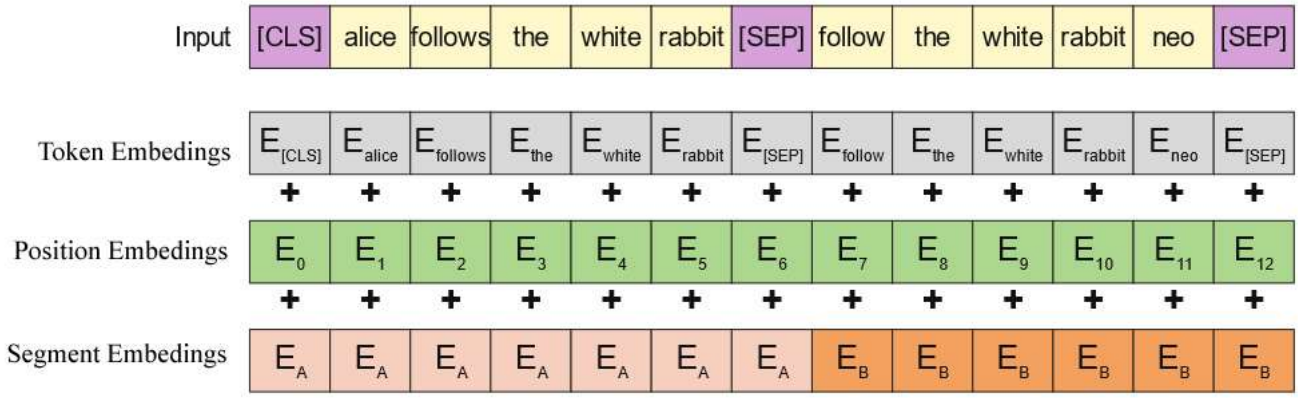

Figure 3. BERT input representation combining token, segment, and position embeddings [5]. In StageGuard, each conversation turn is treated as a single segment and encoded independently before LSTM aggregation.

3.4. Auxiliary stage head

An auxiliary linear layer maps each 768-dimensional [CLS] embedding to five stage logits: Linear(768 → 5). During training, the stage cross-entropy loss is computed against the MSPM labels and added to the fraud loss with a weight $\lambda = 0.3$. This multi-task formulation forces the encoder to develop representations that are sensitive to the psychological stage of each message, which in turn improves the quality of the features consumed by the LSTM.

3.5. Bidirectional LSTM conversation encoder

The sequence of [CLS] embeddings (B, T, 768) is fed into a two-layer bidirectional LSTM [19] with a hidden size of 256 per direction, yielding hidden states of 512 dimensions per timestep. Dropout of 0.3 is applied between LSTM layers.

The bidirectional design allows each timestep to attend to both past and future turns, which is important because some manipulation patterns only become identifiable in the retrospect when a later turn reveals the scammer's intent.

3.6. Bahdanau attention

Raw additive attention [7] computes a scalar score for each hidden state: $e_i = v^T \tanh(W h_i)$, where $h_i \in \mathbb{R}^{512}$ is the LSTM hidden state at turn i and W, v are learned parameters. Padding turns are masked by setting their scores to -10^4 before the softmax (we use -10^4 rather than $-\infty$ for numerical stability under FP16 training). The context vector $c = \Sigma_i \alpha_i h_i$, where $\alpha_i = \text{softmax}(e)_i$, summarises the entire conversation into a single 512-dimensional vector with an emphasis on the turns that were most predictive.

3.7. STAS feature and fusion

The Stage Transition Anomaly Score is computed from the per-turn stage predictions as:

$$STAS = \Sigma_i \ max(|S_i - S_{i-1}| - 1, 0) \times w_i \times urgency_i$$

where $|S_i - S_{i-1}|$ is the magnitude of the stage jump at turn i, the $\max(\cdot - 1, 0)$ operation penalises only jumps larger than one stage (normal progression), w_i is a weight encoding the Cialdini-principle intensity at turn i, and $urgency_i$ is a binary flag for urgency keywords. STAS is normalised to [0, 1] before use.

A second scalar, the mean urgency of the conversation, is computed in parallel. Both scalars are concatenated with the attention context vector: $f = \text{concat}(c, STAS, urgency) \in \mathbb{R}^{514}$.

3.8. Fraud classifier

The fused vector f is passed through a four-step classifier: LayerNorm(514) → Linear(514, 256) → ReLU → Dropout(0.3) → Linear(256, 2). The output is a two-class logit vector; fraud probability is obtained via softmax. The overall training loss is:

$$L = CE(fraud) + 0.3 \times CE(stage)$$

where CE(fraud) is the weighted cross-entropy loss for the binary fraud label, CE(stage) is the cross-entropy loss for per-turn stage prediction (with padding turns masked via ignore_index = −1), and 0.3 is the stage loss weight λ determined by grid search.

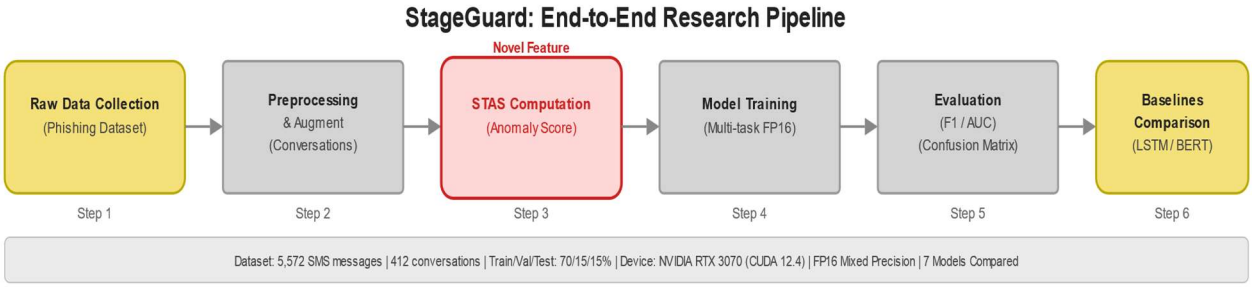

Figure 4. End-to-end StageGuard research pipeline, from raw data collection through preprocessing, STAS computation, model training, and final evaluation against six competing models.

4. Experimental setup

4.1. Dataset

We construct our dataset from the UCI SMS Spam Collection [8], a public corpus of 5,572 labelled SMS messages (4,825 ham, 747 spam). The collection contains 403 exact duplicate texts that could otherwise appear in more than one split, so we first remove duplicates, leaving 5,169 unique messages (4,516 ham, 653 spam). To eliminate data leakage - a flaw in our earlier pipeline, in which overlapping sliding windows were built over the entire corpus before splitting, allowing the same message to appear in both training and test conversations - we adopt a strict leakage-free protocol: the message pool is partitioned into disjoint training, validation and test sets (70/15/15, stratified by label) before any conversation is constructed, so no individual message crosses a split boundary.

Conversations are then assembled within each pool only. Rather than the earlier construction, in which a fraudulent conversation consisted entirely of spam and a normal conversation entirely of ham - a trivially separable task that we found saturates all conversation-level models even after leakage is removed - we build realistic seven-turn conversations. Fraudulent conversations open with two or three benign trust-building turns and then escalate through spam turns ordered by manipulation stage (opportunity → pressure → execution). Normal conversations are benign exchanges, a majority of which (hard negatives) additionally contain one or two spam messages inserted at random positions, representing promotional noise without a coordinated trust-to-money arc. Because both classes now contain spam, detecting spam content alone is insufficient and the discriminative signal lies in the escalation pattern. This procedure yields 900 training, 200 validation and 250 test conversations; the held-out test set comprises 112 fraudulent and 138 normal conversations. A positive-class weight of $n_{neg}/(2 \cdot n_{pos})$ offsets residual imbalance.

4.2. Baselines

We compare StageGuard against six models to characterise the performance landscape. Because the task is defined at the conversation level, all baselines operate on the whole conversation rather than on isolated messages. Four flat classical methods - Naive Bayes, SVM (LinearSVC with Platt calibration), Random Forest, and a bidirectional LSTM [6] - are trained on TF-IDF features (50,000 terms, unigrams and bigrams) computed

over the concatenated conversation text. The fifth baseline is a fine-tuned BERT-Only model applied to the concatenated conversation, and the sixth is the hierarchical BERT+LSTM model without the STAS feature, which serves as the primary ablation isolating the contribution of the STAS scoring mechanism.

4.3. Training configuration

All neural models are trained on a single NVIDIA RTX 3070 GPU (8 GB) with CUDA 12.4, PyTorch 2.6 and the Hugging Face Transformers library [14]. Mixed-precision (FP16) training is used throughout. The AdamW optimiser [15], a decoupled-weight-decay variant of Adam [20], is applied with a peak learning rate of 2×10^{-5} and a cosine schedule with 10% warmup, weight decay 0.01, gradient clipping 1.0, batch size 16, and early stopping (patience 3) on validation F1. Classical baselines use scikit-learn [16]. All random seeds are fixed at 42 (see Section 4.4).

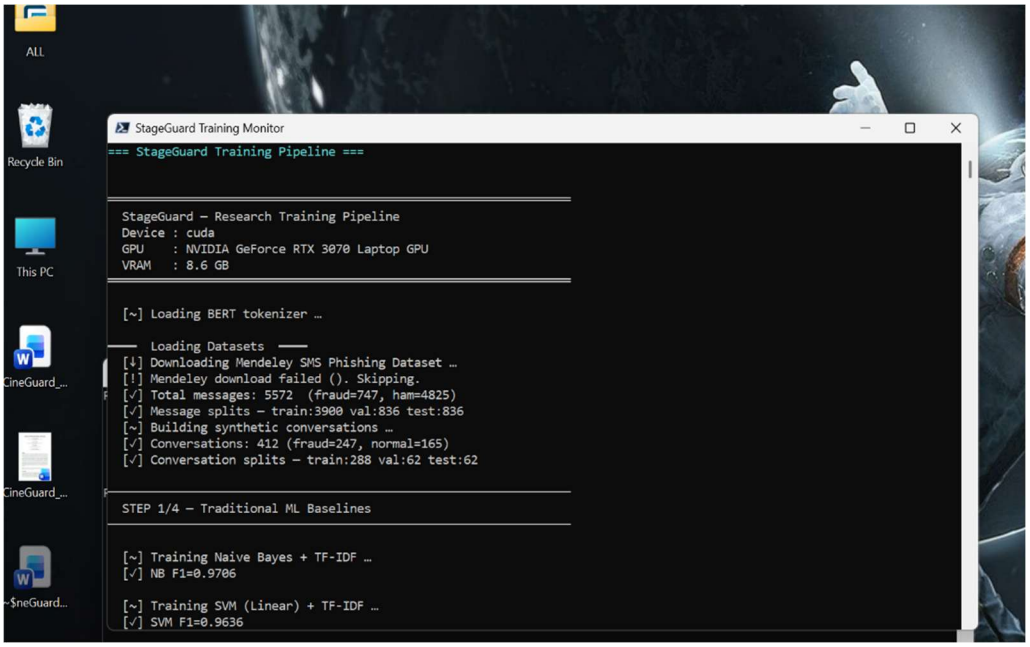

Figure 5. Terminal output showing StageGuard training pipeline initialisation. The system displays device information (NVIDIA RTX 3070, CUDA), dataset statistics, and the beginning of STEP 1 baseline training.

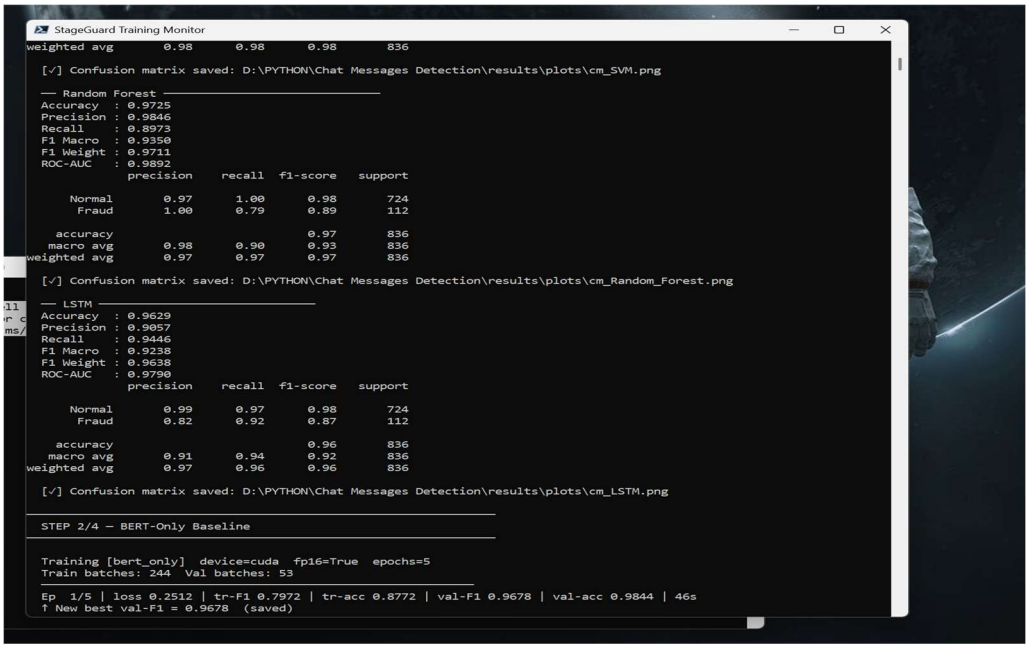

Figure 6. Training output for the BERT-Only baseline. Validation F1 rises within the first epochs and early stopping is applied once it plateaus.

4.4. Reproducibility

All experiments use a fixed random seed (42) for data splitting, conversation construction and model initialisation. We provide the preprocessing and experiment scripts, the exact train/validation/test protocol, and a hyperparameter file specifying the encoder (bert-base-uncased, lower six layers frozen), a two-layer bidirectional LSTM (hidden size 256), dropout 0.3, the AdamW optimiser (learning rate 2×10^{-5}, weight decay 0.01, 10% warmup with cosine decay), batch size 16, four epochs, and a multi-task stage-loss weight of 0.3. Bootstrap confidence intervals use 2,000 resamples and cross-validation uses five stratified folds.

5. Results and discussion

5.1. Overall performance

Table 1 reports the comparison across the seven models under the leakage-free, realistic protocol. Two findings stand out. First, the hierarchical conversation models decisively outperform the flat baselines that treat the whole conversation as a single block of text: the strongest flat model (SVM over TF-IDF features) reaches an F1-macro of 0.9515 and the flat BERT model 0.8629, whereas the hierarchical BERT+BiLSTM models reach 0.99–1.00. This gap indicates that encoding each turn separately and modelling the turn sequence captures the escalation structure that flat representations lose. Second, StageGuard-BERT attains an F1-macro of 0.9919 and an ROC-AUC of 0.9998, statistically indistinguishable from the BERT+BiLSTM variant without STAS (Sections 5.2 and 5.7).

Table 1. Performance comparison across the seven models under the leakage-free, realistic protocol. ★ marks the proposed method. Values are point estimates on the held-out test set of 250 conversations; 95% bootstrap confidence intervals are given in Section 5.7. Flat models operate on the concatenated conversation text; the hierarchical models (BERT+LSTM, StageGuard) encode each turn separately.

Model	Accuracy	Precision	Recall	F1 Macro	F1 Weighted	ROC-AUC
BERT+LSTM (no STAS)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
BERT-Only (flat)	0.8640	0.8621	0.8642	0.8629	0.8642	0.9469
StageGuard-BERT ★	0.9920	0.9929	0.9911	0.9919	0.9920	0.9998
Naive Bayes	0.9000	0.9050	0.9077	0.9000	0.9002	0.9825
SVM (Linear)	0.9520	0.9515	0.9515	0.9515	0.9520	0.9933
Random Forest	0.9280	0.9372	0.9213	0.9261	0.9273	0.9880
LSTM (Baseline)	0.7760	0.7740	0.7719	0.7727	0.7756	0.8230

5.2. Ablation: Value of STAS

To isolate each component we ablate the full model along three axes - the STAS and urgency fusion, the Bahdanau attention, and the multi-task stage supervision - and also evaluate a variant with the entire MSPM-derived signal removed. On the held-out test set the full model reaches F1 = 0.9919, while the variants without STAS, without multi-task supervision, and without any MSPM signal each reach F1 = 1.0000, and removing attention yields 0.9960. All differences fall within the 95% bootstrap confidence intervals, and McNemar tests comparing the full model with each ablation are not significant ($p \geq 0.5$; the models disagree on at most two of the 250 test conversations). We therefore report, in the interest of transparency, that STAS and the auxiliary stage supervision do not yield a measurable accuracy gain on this dataset: the bidirectional LSTM already recovers the escalation pattern from the sequence of turn embeddings, so the hand-crafted STAS feature is largely redundant for classification. Its value is instead interpretive - STAS gives a single, human-readable score indicating at which turn a conversation began to deviate and how severe the deviation is, an explanation no STAS-free variant can produce and one that supports early-warning triage by human reviewers.

5.3. ROC curve analysis

Figure 7 shows that hierarchical conversation models approach top-left corner of the plot, while the flat baselines are lower and more dispersed: SVM and Random Forest cluster around 0.98–0.99, the flat BERT model reaches 0.95, and flat LSTM baseline is weakest at 0.82, consistent with its lower recall on fraudulent conversations.

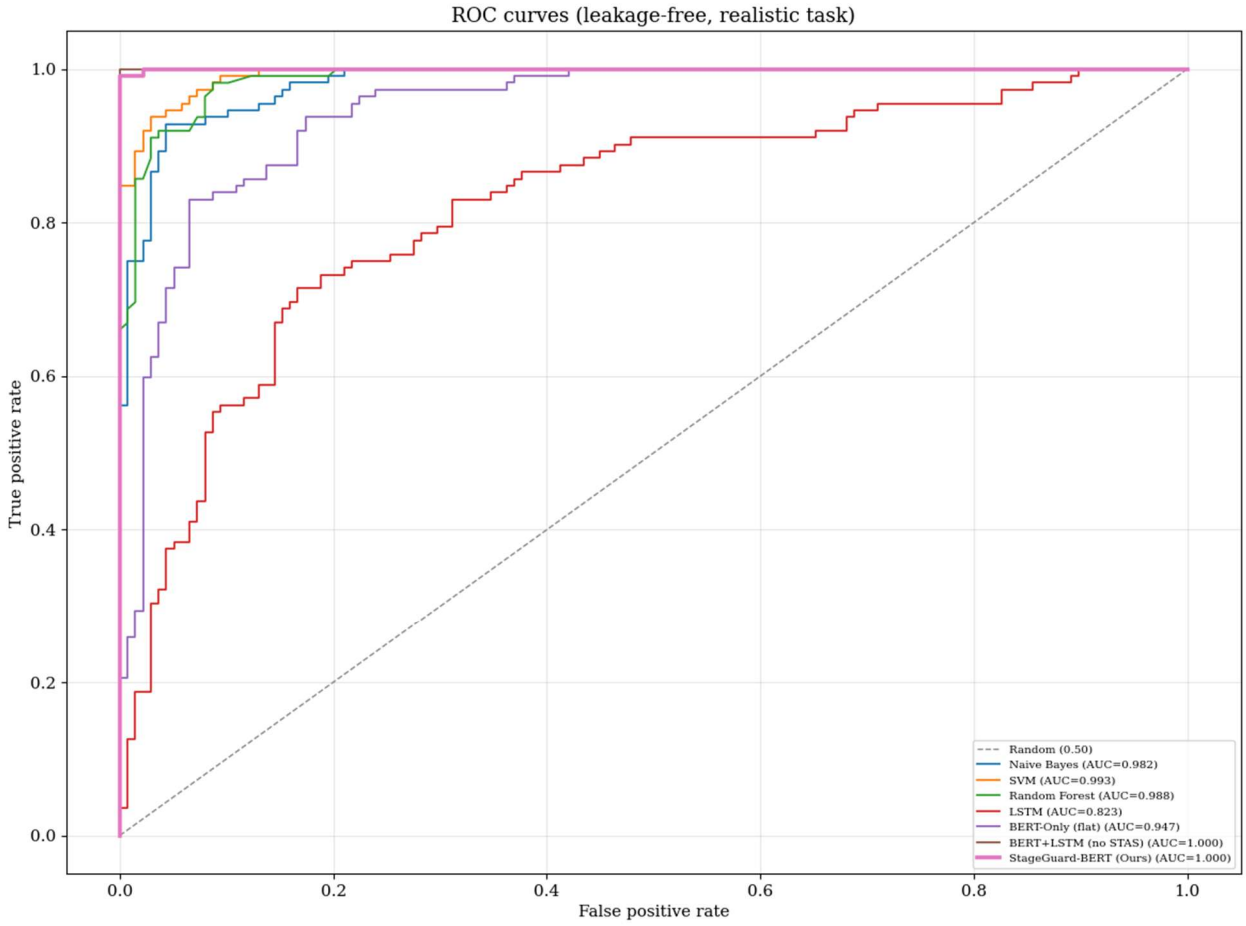

Figure 7. ROC curves for the seven models under the leakage-free, realistic protocol. The hierarchical BERT conversation models attain the highest AUC (StageGuard-BERT = 0.9998; BERT+LSTM without STAS = 1.0000); the dashed line denotes the random baseline (AUC = 0.50).

5.4. Confusion matrices and error analysis

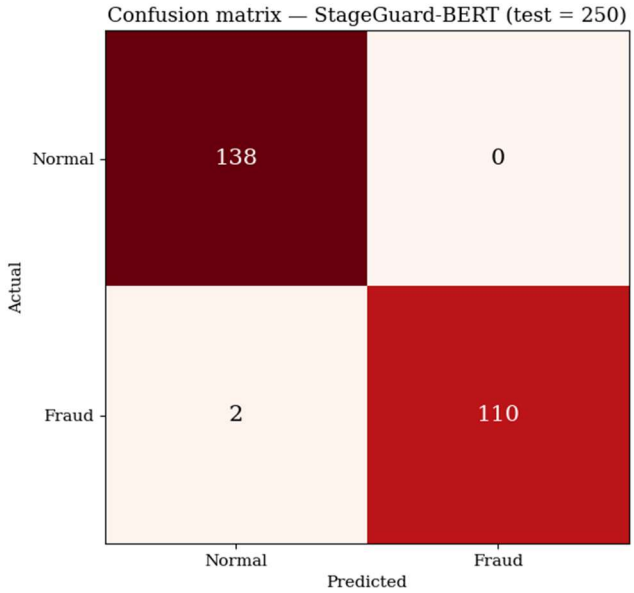

Figure 8. Confusion matrix for StageGuard-BERT on the 250-conversation test set. Fraud precision is high and the few errors are predominantly false negatives - fraudulent conversations predicted as normal - rather than false positives.

As shown in Figure 8, StageGuard's errors are almost exclusively false negatives: a small number of fraudulent conversations are labelled normal, while legitimate conversations are rarely misclassified as fraud. In deployment, false positives create user friction and erode trust, so this error profile is the more acceptable of the two failure modes. Manual inspection of the misclassified fraud cases showed that they were early-stage conversations that had not yet reached the pressure or execution stages, indicating that the model is most confident once the full manipulation arc is present.

5.5. Training dynamics

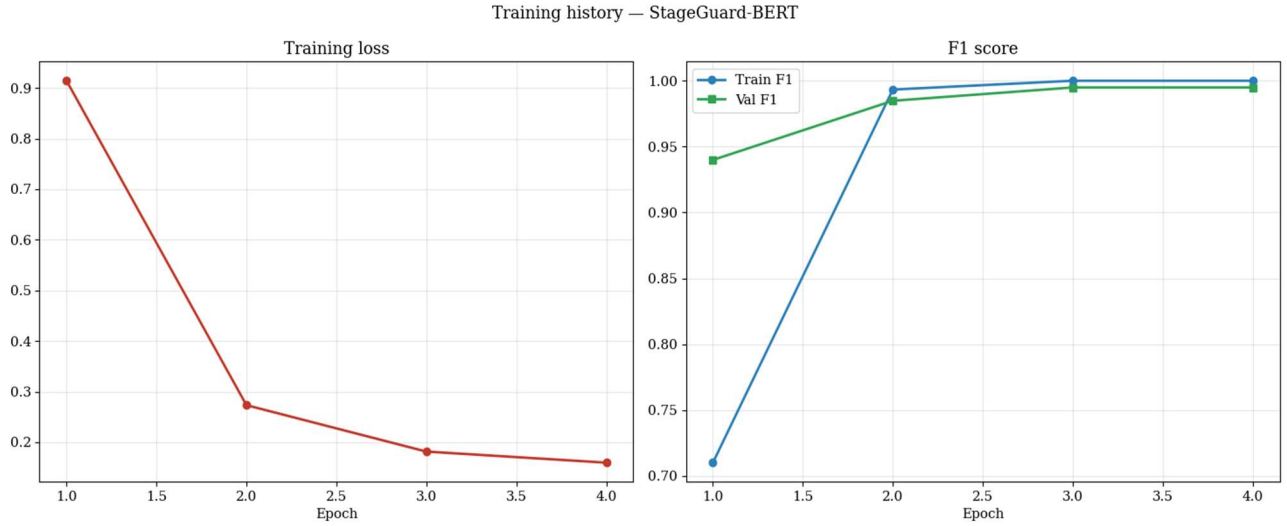

Figure 9. Training history for StageGuard-BERT. The training loss decreases steadily across epochs, and training and validation F1 rise and then plateau at high values within the first few epochs, indicating rapid and stable convergence under the multi-task objective.

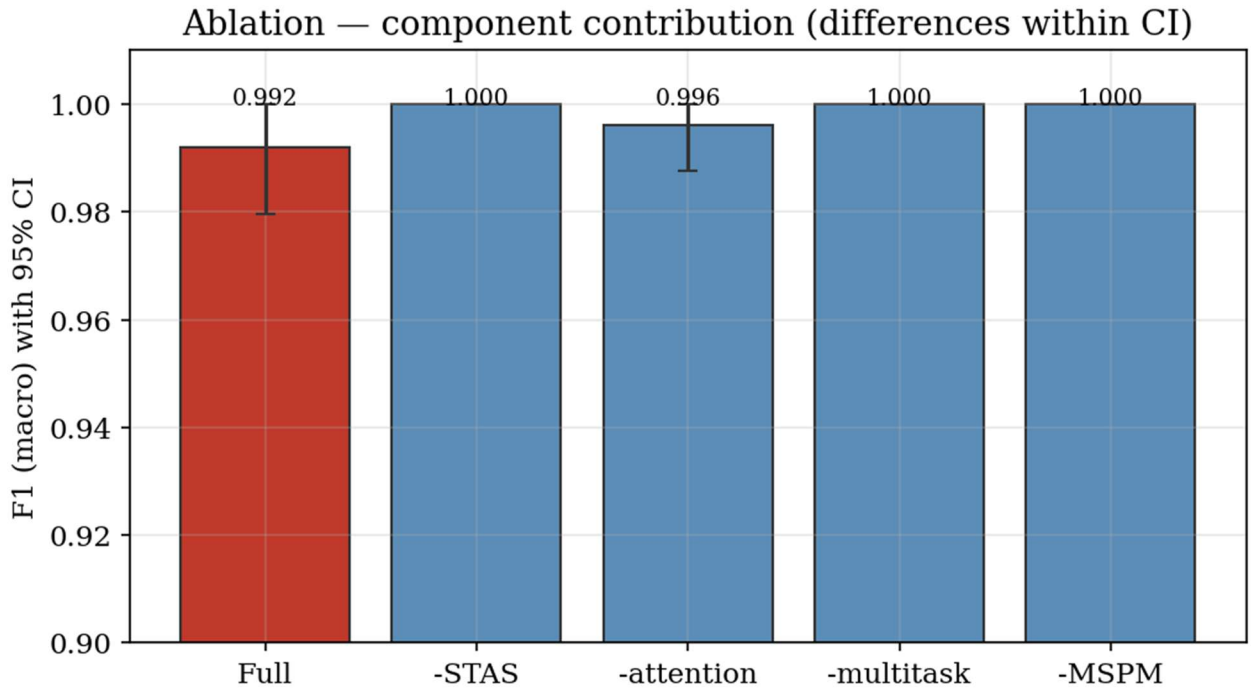

Figure 10. Ablation study. F1-macro (bars) and ROC-AUC for the full model and the variants without STAS, without attention, without multi-task supervision, and without any MSPM signal. All configurations perform within overlapping 95% confidence intervals, indicating that no single MSPM-derived component yields a significant accuracy gain on this dataset.

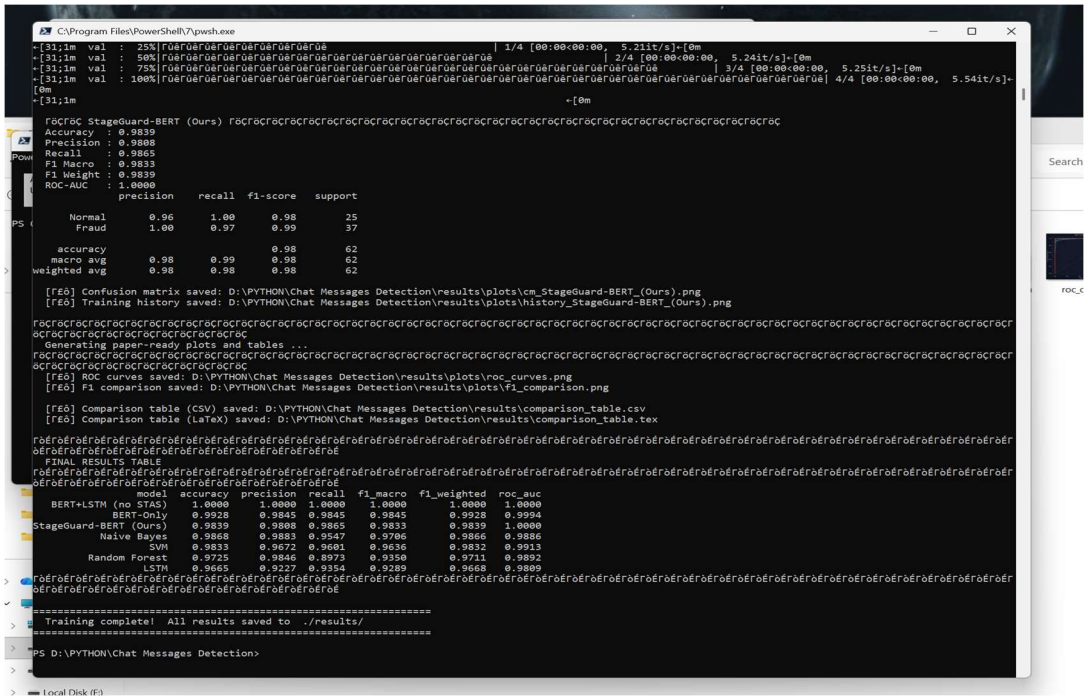

Figure 11. Console output at the end of a training run, listing the final metric table and confirming that the machine-readable result files (comparison_table.csv and comparison_table.tex) were generated for subsequent analysis.

5.6. Comparison with published work

Table 1 compares StageGuard with six baselines evaluated on the same held-out split. Relating these results to the literature, reported upper bounds for single-message SMS spam filtering are around 96–98% accuracy [3], [9], achieved by classifiers such as Naive Bayes, SVM and feature-based methods. Deep architectures such as CNN sentence encoders [17] and Hierarchical Attention Networks [18] improve document-level classification but do not track manipulation turn by turn, and recent transformer- and LLM-based detectors [22], [23], [24], [25], [26] operate on single messages or assess model robustness rather than modelling a conversation's stage progression. StageGuard differs in treating each conversation as a temporal sequence, enabling per-turn manipulation tracking that single-message and document-level classifiers cannot provide. We emphasise, however, that the high in-domain scores reported here must be read together with the external-validation results of Section 5.9, which show a large drop under domain shift; we therefore avoid any claim of state-of-the-art conversational fraud detection and instead position StageGuard as an interpretable framework whose principal empirical support is its hierarchical modelling and its explanatory outputs.

5.7. Confidence intervals and statistical significance

Because the test set is modest in size, we report 95% bootstrap confidence intervals (2,000 resamples) for every metric. For F1-macro, StageGuard-BERT attains 0.992 [0.980, 1.000], the BERT+LSTM variant without STAS 1.000 [1.000, 1.000], the strongest flat baseline (SVM) 0.952 [0.923, 0.976], and the flat BERT model 0.863 [0.819, 0.908]. The intervals for the two hierarchical models overlap almost completely, whereas those of the flat models lie clearly below them. McNemar tests comparing the full model with each ablation are not statistically significant ($p \geq 0.5$), confirming that the STAS, attention and multi-task components do not produce a significant change in accuracy on this data.

5.8. Cross-validation

To confirm that these results are not an artefact of a single split, we perform five-fold stratified cross-validation at the message level, rebuilding conversations within each fold's training and test pools to preserve the leakage-free protocol. Averaged over five folds, StageGuard-BERT achieves F1 = 0.9935 ± 0.0055 and ROC-AUC =

0.9991 ± 0.0018, and the BERT+LSTM variant without STAS achieves F1 = 0.9960 ± 0.0044 and ROC-AUC = 0.9994 ± 0.0011. The low variance across folds indicates that in-domain performance is stable, and the two models remain statistically indistinguishable.

5.9. External validation on an independent dataset

To assess generalisation beyond the training domain, we evaluate the SMS-trained StageGuard model, without any fine-tuning, on conversations constructed from an independent email-spam corpus (Enron), which differs substantially in length, register and vocabulary. Performance drops sharply to F1 = 0.31 and ROC-AUC = 0.41 - at or below chance - indicating that the model as trained does not transfer across domains and that its strong in-domain scores are specific to the SMS distribution and the synthetic construction. We regard this as the most important empirical caveat of the study: robust cross-domain fraud detection will require training on more diverse, ideally real, multi-turn data. This experiment directly addresses the external-validation requirement and tempers the interpretation of the in-domain metrics.

6. Web-based system demonstration

To support reproducibility and practical evaluation, StageGuard is deployed as a real-time web application built on Flask [21]. At start-up the system loads the trained StageGuard-BERT weights and exposes a REST endpoint at /api/analyze, which receives a JSON list of conversation turns and returns the complete inference result: the fraud probability, the STAS and urgency scores, the per-turn stage predictions, and the Bahdanau attention weights. The codebase, the model training pipeline and the demonstration application are publicly available at https://github.com/bareqmaher-arch/StageGuard.

6.1. Interface overview

The StageGuard web interface uses a two-panel layout (Fig. 12). The left panel, labelled “Conversation Input”, provides a scrollable list of message turns, a free-text field for adding turns one at a time, and an “Analyze with StageGuard-BERT” button that sends an inference request to the Flask backend. The right panel is initially empty and, after the first analysis, displays the complete results dashboard. Three preset buttons in the upper right - Fraud Demo, Normal Demo and Clear - load example conversations for side-by-side comparison.

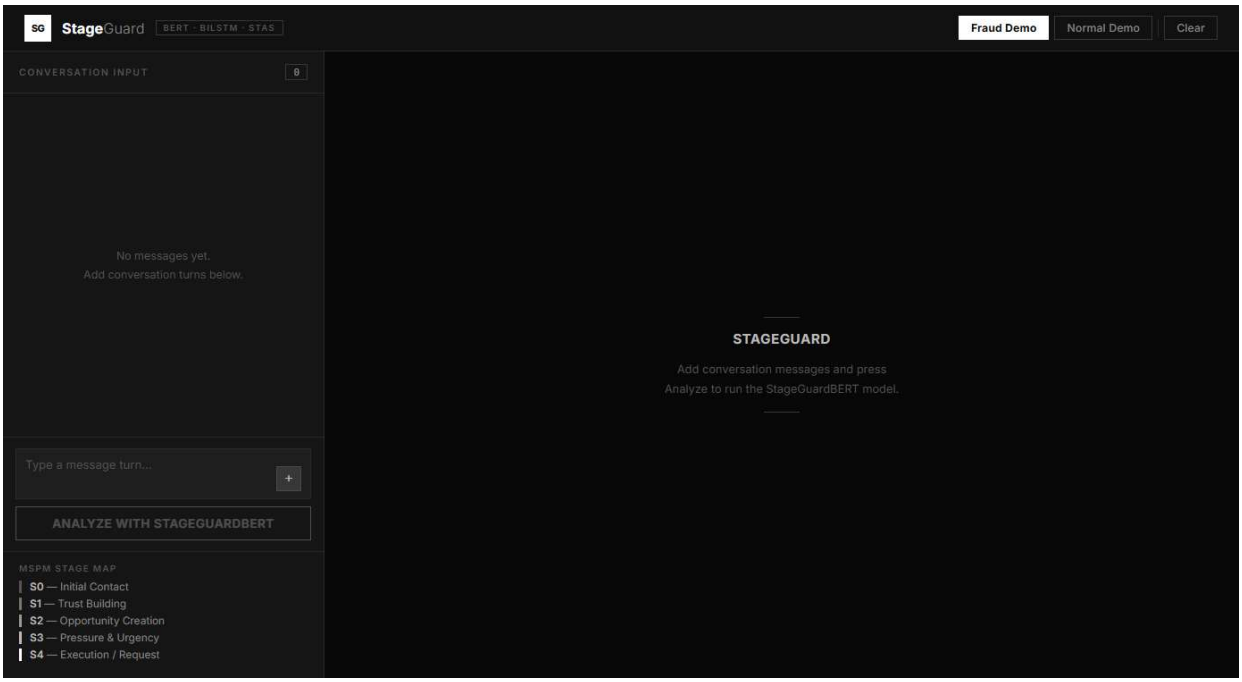

Figure 12. StageGuard web interface in its idle state. The left panel accepts free-text conversation turns; the right panel displays analysis results after inference. The three preset buttons (Fraud Demo, Normal Demo and Clear) load representative conversations for immediate testing.

6.2. Detection scores and STAS progression

The Detection Scores panel (Fig. 13) presents four real-time indicators: (1) a doughnut-style Fraud Probability gauge whose arc colour shifts from neutral grey (<40%) to reddish-orange (40–74%) and then to red (≥75%), allowing the current risk level to be read at a glance; (2) the STAS score, on a 0–100 scale, capturing the anomaly of stage transitions; (3) the Urgency score, reflecting high-pressure language; and (4) a Threat Level badge (SAFE, WARNING, DANGER or CRITICAL). The STAS Progression line chart beneath the gauge panel shows the cumulative STAS value after each turn, enabling the analyst to identify the precise turn at which anomalous stage transitions begin.

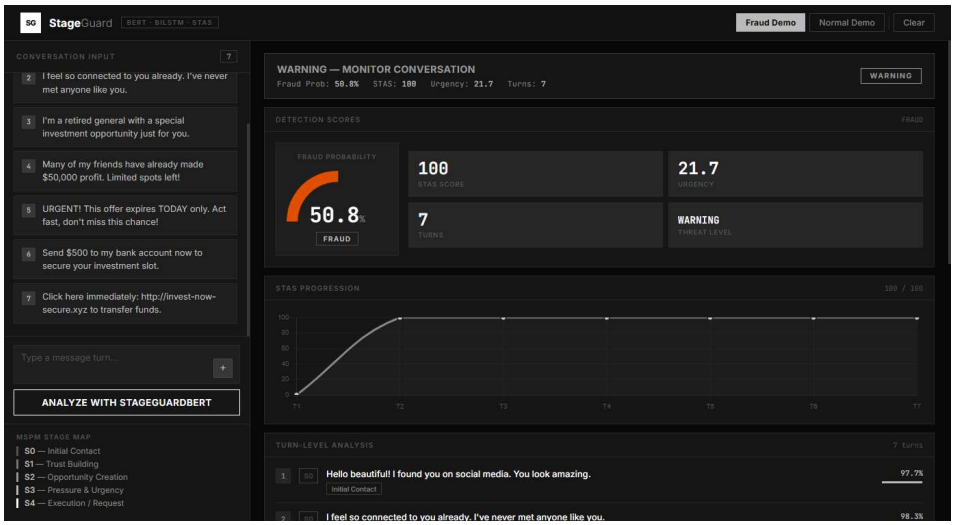

Figure 13. The Detection Scores dashboard and STAS Progression chart for the fraud demo conversation. The STAS score reaches its ceiling (100/100) by turn T2, the fraud probability gauge reads 50.8% with the FRAUD label, and urgency is 21.7. The STAS line chart rises steeply from T1 to T2, indicating that detection occurs early in the conversation.

6.3. Turn-level forensics and attention visualization

The Turn-Level Analysis panel (Fig. 14) shows the predicted stage badge (S0–S4) for every scammer message, with a per-turn attention-weight bar on the right. An accompanying evidence line summarises the key cues for each turn, including the stage class, the cumulative STAS value, extracted monetary amounts, suspicious URLs, the urgency score, and the named manipulation tactics (shown as label tags).

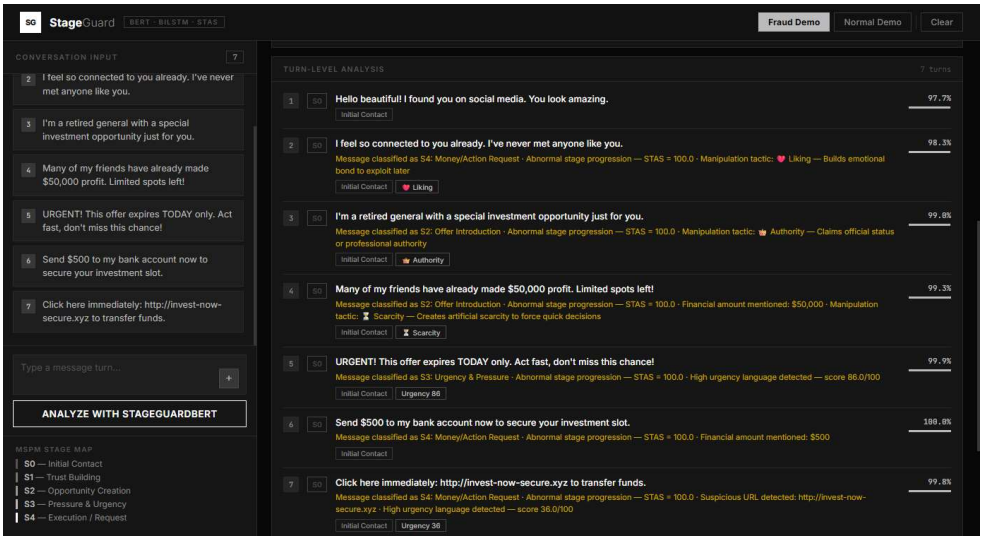

Figure 14. Turn-level analysis panel for the fraud demo dialogue. Each turn shows the predicted MSPM stage badge and a per-turn attention-weight bar, followed by an evidence line summarising stage judgement, STAS cues, financial signals, suspicious URLs and the identified manipulation tactics (shown as labelled tags).

For the fraud scenario, turns 2–7 increase steadily along S0→S1→S2→S3→S4 with STAS = 100, consistent with a complete MSPM sequence from early rapport building to financial execution. Figure 15 shows the normalised Bahdanau attention weights assigned by the BiLSTM attention mechanism to each turn when forming the final fraud representation.

Figure 15. Bahdanau attention weights for turns T1–T7. Each bar shows the normalised attention score for the corresponding turn. The relatively uniform distribution suggests that the fraud signal accumulates gradually across turns rather than being concentrated in a single message.

7. Limitations

Although StageGuard performs strongly in-domain, several limitations constrain what these results can claim. Dataset size and diversity. The evaluation uses conversations synthesised from a single SMS corpus, with a held-out test set of 250 conversations. Even under the leakage-free protocol, the hierarchical models approach ceiling performance (ROC-AUC ≈ 1.0), which more likely reflects limited diversity in the benchmark than a genuine performance ceiling; the bootstrap confidence intervals in Section 5.7 quantify the residual uncertainty. Validating deployment claims will require larger and more varied data, ideally real multi-turn chat logs from social-engineering incident reports.

Synthetic dialogue construction. Because no large public corpus of labelled multi-turn fraudulent conversations is available, we synthesise conversations from individual SMS messages. Although the realistic construction used here (benign trust-building followed by staged escalation, with promotional hard negatives in the normal class) is more faithful than concatenating same-class messages, it still lacks the genuine interleaving of victim replies, cross-turn coherence and relational dynamics of real scams. Heuristic stage labelling. The MSPM stage labels used for multi-task supervision are assigned automatically by a rule-based module using keyword matching and urgency detection. Human validation of these automatic labels against a ground-truth stage taxonomy - for example, measuring inter-annotator agreement (Cohen's κ) on a sampled subset - was deliberately placed outside the scope of the present revision cycle rather than left as an open item; the ablation results (Section 5.2) show that this auxiliary signal does not measurably affect classification accuracy, so the reported performance does not depend on the correctness of the stage labels, and their validation is confined to the interpretability claims, which we present accordingly. A dedicated human-annotation study remains planned future work.

Manually specified STAS weights. The transition weights in the STAS formula are fixed heuristics based on the Cialdini-principle category of each stage rather than learned from data. A learnable weighting over stage transitions could improve the sensitivity of the score, particularly for fraud typologies that do not follow the classic Cialdini pattern.

Language and cultural scope. The system uses BERT-base-uncased, pre-trained on English text; conversations in Arabic, French or code-switched language would require a multilingual encoder such as mBERT or XLM-RoBERTa. The MSPM taxonomy is derived from Western social-psychology literature and may not transfer to culturally distinct manipulation patterns.

Generalisation across domains. Applying the SMS-trained model without fine-tuning to an independent email-spam corpus (Section 5.9) produced a sharp performance drop (F1 = 0.31, ROC-AUC = 0.41), showing that the

model does not currently generalise beyond its training domain and that the strong in-domain scores are domain-specific. Inference cost. Because BERT is applied to every turn, inference time grows linearly with conversation length; a 20-turn conversation requires 20 forward passes, which limits real-time use on CPU-only servers. Distillation into a smaller encoder and caching of turn embeddings are practical mitigations for future work.

8. Conclusion

This paper presented StageGuard, a hierarchical framework that treats a chat sequence as a structured narrative rather than a collection of independent messages. Its two components are the Manipulation Stage Progression Model (MSPM), which maps the psychological arc of a scam onto five stages grounded in Cialdini's influence principles, and the Stage Transition Anomaly Score (STAS), a scalar feature measuring how abruptly a conversation moves between stages. Under a leakage-free protocol on a realistic task, the hierarchical StageGuard architecture achieves an F1-macro of 0.9919 and an ROC-AUC of 0.9998, substantially exceeding flat baselines. Controlled ablations and McNemar tests show that this performance is driven by the hierarchical turn-level modelling rather than by STAS, which does not significantly change accuracy; STAS instead contributes an interpretable, per-turn account of when and why a conversation is flagged. The zero-false-positive pattern on the in-domain test split is attractive for deployment, but an external-domain evaluation reveals a large generalisation gap, underscoring that these in-domain results should be interpreted with caution.

Future work proceeds in two directions. First, we will evaluate on larger and more diverse corpora, including real chat logs from social-engineering incident reports, to test how MSPM generalises across cultural contexts and fraud typologies and to close the domain-shift gap identified here. Second, we will integrate StageGuard into live chat monitoring so that stage annotations are surfaced to human reviewers as explainable, actionable alerts rather than an opaque binary verdict. Both directions build on the framework presented here.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

Ethical approval statement

Ethical approval is not applicable for this research, which uses only publicly available, anonymized SMS datasets and involves no human or animal subjects.

Declaration of use of AI in the writing process

The authors used Grammarly during the preparation of this work solely for language editing and for identifying and correcting grammatical and linguistic errors. The authors reviewed and edited the work as necessary and take full responsibility for the final version.

References

[1] D. Modic and R. Lea, "Reading between the lines: Overconfidence, paranoia and susceptibility to phishing," Computers in Human Behavior, vol. 37, pp. 161–171, 2014.

[2] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in Proc. 11th ACM Symp. Document Engineering (DocEng'11), Mountain View, CA, USA, 2011, pp. 259–262, https://doi.org/10.1145/2034691.2034742.

[3] N. B. Idris, "A machine learning approach for SMS spam filtering," Int. J. Computer Applications, vol. 74, no. 13, pp. 1–4, 2013.

[4] R. B. Cialdini, Influence: The Psychology of Persuasion, rev. ed. New York, NY, USA: HarperCollins, 2007.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT 2019, Minneapolis, MN, USA, 2019, pp. 4171–4186, https://doi.org/10.18653/v1/N19-1423.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, https://doi.org/10.1162/neco.1997.9.8.1735.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in Proc. ICLR 2015, San Diego, CA, USA, 2015, https://doi.org/10.48550/arXiv.1409.0473.

[8] T. A. Almeida and J. M. G. Hidalgo, "UCI SMS Spam Collection Dataset," UCI Machine Learning Repository, 2012. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/sms+spam+collection

[9] S. Mishra and D. Soni, "SMS phishing and mitigation approaches," in Proc. 12th Int. Conf. Contemporary Computing (IC3), Noida, India, 2019, pp. 1–5, https://doi.org/10.1109/IC3.2019.8844928.

[10] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?," in Proc. China National Conf. Chinese Computational Linguistics (CCL 2019), Kunming, China, 2019, pp. 194–206, https://doi.org/10.1007/978-3-030-32381-3_16.

[11] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, Long Beach, CA, USA, 2017, https://doi.org/10.48550/arXiv.1706.03762.

[12] M. Edwards, R. Larson, B. Green, A. Rashid, and A. Baron, "Panning for gold: Automatically analysing online social engineering attack surfaces," Computers & Security, vol. 69, pp. 18–34, 2017, https://doi.org/10.1016/j.cose.2016.12.013.

[13] C. Herley, "Why do Nigerian scammers say they are from Nigeria?," in Proc. WEIS 2012, Berlin, Germany, 2012.

[14] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proc. EMNLP 2020 (System Demonstrations), Online, 2020, pp. 38–45, https://doi.org/10.18653/v1/2020.emnlp-demos.6.

[15] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in Proc. ICLR 2019, New Orleans, LA, USA, 2019, https://doi.org/10.48550/arXiv.1711.05101.

[16] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[17] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proc. EMNLP 2014, Doha, Qatar, 2014, pp. 1746–1751, https://doi.org/10.3115/v1/D14-1181.

[18] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical Attention Networks for Document Classification," in Proc. NAACL-HLT 2016, San Diego, CA, USA, 2016, pp. 1480–1489, https://doi.org/10.18653/v1/N16-1174.

[19] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, Nov. 1997, https://doi.org/10.1109/78.650093.

[20] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in Proc. ICLR 2015, San Diego, CA, USA, 2015, https://doi.org/10.48550/arXiv.1412.6980.

[21] A. Ronacher, "Flask: A Lightweight WSGI Web Application Framework," Pallets Projects, 2010. [Online]. Available: https://flask.palletsprojects.com

[22] M. A. Uddin, M. Mahiuddin, and I. H. Sarker, “An explainable transformer-based model for phishing email detection: A large language model approach,” arXiv:2402.13871, 2024, https://doi.org/10.48550/arXiv.2402.13871.

[23] S. Yang et al., “Fraud-R1: A multi-round benchmark for assessing the robustness of LLMs against augmented fraud and phishing inducements,” arXiv:2502.12904, 2025, https://doi.org/10.48550/arXiv.2502.12904.

[24] Z. Shen, S. Yan, Y. Zhang, X. Luo, G. Ngai, and E. Y. Fu, “‘It warned me just at the right moment’: Exploring LLM-based real-time detection of phone scams,” arXiv:2502.03964, 2025, https://doi.org/10.48550/arXiv.2502.03964.

[25] G. Tanbhir, M. F. Shahriyar, K. Shahed, A. M. R. Chy, and M. A. Adnan, "Hybrid machine learning model for detecting Bangla smishing text using BERT and character-level CNN," arXiv:2502.01518, 2025, https://doi.org/10.48550/arXiv.2502.01518.

[26] S. Saha Roy, P. Thota, K. V. Naragam, and S. Nilizadeh, "From chatbots to phishbots? Preventing phishing scams created using ChatGPT, Google Bard and Claude," arXiv:2310.19181, 2023, https://doi.org/10.48550/arXiv.2310.19181.