Production Systems and Information Engineering
Volume 14 (1), pp. 1-26
doi: https://doi.org/10.32968/psaie.2026.1.1

DECISION-RULE-CONTROLLED MULTI-AGENT VALIDATION FOR HUNGARIAN BANKING COMPLAINT CLASSIFICATION

ZSOLT KRUTILLA
Obuda University, John von Neumann
Faculty of Informatics
krutilla.zsolt@nik.uni-obuda.hu

University of Dunaújváros, Institute of
Information Technology
krutilla.zsolt@uniduna.hu

Abstract. This paper presents a decision-rule-controlled multi-agent validation framework for Hungarian banking complaint classification. A previously trained DistilBERT model provides the initial category prediction, while Qwen2.5-7B, Qwen3-8B, and Mistral-7B are used as reasoning and validation agents. A rule-based decision agent determines whether the baseline label is accepted, revised, or routed to exception handling. Experiments were conducted on a 31-category random sample and a reduced Top6 + Exception balanced sample. The results show that unrestricted LLM-based relabeling may significantly degrade performance, while conservative decision rules better preserve baseline stability. The findings indicate that LLMs are more suitable as controlled validation components than as unrestricted classifiers in closed-label banking NLP tasks.

Keywords: Multi-Agent LLM; Decision-Controlled Validation; Banking NLP

EARLY ACCESS

1. Introduction

Automatic classification of banking customer complaints is a closed-label, multi-class NLP task in which each complaint must be assigned to a predefined category. In banking, this has practical relevance because the category may affect case routing, internal handling, deadlines, and regulatory reporting. Therefore, such systems must be accurate, stable, consistent, and auditable [1, 2].

Previous experiments have shown that fine-tuned transformer models, especially BERT-based architectures, can be effectively applied to Hungarian banking complaint classification [3–5]. Their contextual representations are useful when the relevant category is not explicitly stated or when a complaint refers to multiple products, transactions, or customer issues [3, 6]. However, even strong baseline models may make errors because the dataset is imbalanced, rare categories have limited representation, and several complaint categories are semantically close [7].

Large language models offer a new opportunity to interpret and validate

classification decisions. They can assess whether a baseline category is justified and suggest alternatives [8, 9]. However, their direct use as standalone classifiers is risky in regulated, closed-label banking environments. Generative models may interpret labels too freely, deviate from the predefined category system, or provide confident answers in uncertain cases [10].

For this reason, LLMs should not replace fine-tuned classifiers in this task. Instead, they can serve as controlled validation and decision-support components [8, 9]. The central research question is therefore how the effectiveness of LLM-based validation depends on the selected LLM and the rule system of the decision agent. A loose rule may increase relabeling but also harmful interventions, while a stricter rule may better preserve baseline stability.

This study investigates a controlled multi-agent validation architecture for Hungarian banking complaint classification. The fine-tuned DistilBERT classifier provides the initial category prediction, the LLM-based reasoning and validator agents examine this decision, and the rule-based decision agent determines whether the baseline label should be retained, revised, or routed to exception handling [4, 8, 9]. Thus, generative models are not used for unrestricted relabeling, but as controlled validation components within the banking complaint classification process.

2. Related Work

The automated processing of banking complaints is related to transformer-based text classification, domain-specific NLP, and recent LLM-based validation methods [1, 6, 8, 9]. The task is a closed-label, multi-class classification problem, but the texts must be interpreted in a banking, customer service, and regulatory context [1, 2, 11]. This makes the problem different from general text classification and motivates the use of controlled validation rather than unrestricted generative relabeling [8–10].

2.1. Transformer-based Text Classification

Transformer-based models have significantly improved text classification by using contextual token representations [3, 5, 6]. This is important when the target category cannot be inferred from isolated keywords but depends on the semantic structure and broader context of the text [3, 6].

Banking complaints are complex inputs because they vary in length and may refer to several products, transactions, fees, service issues, or suspected financial abuse within the same document [1, 2, 11]. Therefore, classification cannot be reliably reduced to dictionary-based or rule-based matching [6, 11].

In previous experiments on Hungarian banking complaint classification, DistilBERT proved to be the most favorable baseline from a practical perspective [11, 12]. Its advantage was not only classification performance, but also stable and reproducible operation under the available hardware conditions [4, 11, 12]. Experiments with HuBERT and other Hungarian language models also showed that language-specific pretraining alone does not guarantee better results [12, 16]. Practical performance depends on the interaction between model architecture, dataset structure, and fine-tuning settings [5, 7, 12].

2.2. Banking NLP and Closed-Label Complaint Classification

Banking NLP tasks depend strongly on institutional, business, and regulatory context [1, 11]. Complaint categories may be linked to different handling procedures, reporting obligations, and compliance consequences. Therefore, classification errors are not only statistical errors but may also create operational risk [1, 2, 11].

International studies on automated complaint processing provide relevant evidence for the proposed approach. Coussement and Van den Poel [1] demonstrated that linguistic-style features improve the automatic identification of customer complaint emails. Choe et al. [2] examined 27 complaint categories in a Korean telecommunications context and reported 75% domain-level accuracy but only 51% subcategory accuracy. Their findings also supported referring uncertain cases for expert review, highlighting the difficulty of fine-grained complaint classification.

More recent research has extended these principles to LLM-based validation and decision support. Guo et al. [9] identified role specialization and coordinated decision-making as important features of multi-agent architectures, while also highlighting the risks of unreliable agent outputs. Zheng et al. [29] demonstrated the potential of LLM-based evaluation but identified systematic judgment biases. Related research on selective prediction and learning to defer [22]–[24] further supports controlled intervention when automated decisions are uncertain.

These findings motivate the present approach, which combines a fixed banking complaint classifier with LLM-based reasoning and validation under deterministic decision rules. Unlike unrestricted LLM-based relabeling, the proposed framework explicitly measures beneficial and harmful interventions within a closed regulatory label space.

The closed-label structure is a key methodological constraint. The system cannot generate new categories; it must select one predefined label. This differs from the natural behavior of generative LLMs, which may produce alternative interpretations or free-text label suggestions [8–10]. In banking complaint classification, model outputs must remain controlled [11].

Based on previous results, DistilBERT is a suitable baseline, but it is not error-free [11, 12]. Semantic overlap between categories, rare classes, and long or incomplete complaint texts may all lead to incorrect predictions [7, 11]. The next research question is therefore how baseline decisions can be validated or routed to exception handling.

2.3. Large Language Models as Validation Agents

LLMs offer a new opportunity for content-based validation. They can interpret the complaint text, assess whether the baseline category is justified, and consider alternative categories [8, 9]. However, using them directly as final classifiers is risky in closed-label tasks, because they may interpret categories too freely or provide confident answers in ambiguous cases [10].

EARLY ACCESS

In this study, Qwen2.5-7B, Qwen3-8B, and Mistral-7B are therefore used as reasoning and validator agents, not as standalone classifiers [13–15]. The aim is not to determine which LLM is the best classifier, but to examine how their validation behavior differs: how often they intervene, what types of changes they propose, and whether those changes improve or degrade the final output.

2.4. Multi-Agent and Decision-Controlled Validation

The proposed multi-agent architecture separates the classification and validation roles [8, 9]. The base classifier agent, built on DistilBERT, produces the initial category prediction [4, 11]. The reasoning agent analyzes the complaint and the baseline prediction. The validator agent evaluates whether the baseline label is acceptable. Finally, the decision agent applies rule-based logic to decide whether the baseline label should be retained, revised, or routed to exception handling (Fig. 1).

This structure is important because the LLM cannot directly overwrite the baseline classifier. The decision rule acts as a control layer. The loose rule allows more relabeling and serves as a negative control for overly permissive validation. The conservative rule applies stricter conditions and tests whether LLM-based validation can preserve baseline stability with lower intervention risk. Business exception routing adds a further decision-support option by routing selected uncertain or risky cases to exception handling instead of forcing them into a main category [8–10].

2.5. Position of the Present Study

This study builds on previous transformer-based research on Hungarian banking complaint classification but addresses a different question [11, 12]. Earlier work focused on baseline classifier performance and fine-tuning configurations. The present study examines whether the decisions of a fixed DistilBERT baseline can be checked through LLM-based validation and rule-based decision control.

The contribution is therefore not another baseline model benchmark, but the analysis of a controlled agent-based validation process. The framework represents an intermediate solution between fully automated classification and human review: the baseline classifier provides a reproducible prediction, the LLM agents add content-based validation, and the decision agent regulates whether the system may intervene in the final category decision [8, 9, 11].

3. Baseline Classification Setting

The multi-agent validation framework used the DistilBERT model that had shown the most favorable results in previous studies as its baseline classifier. The selected configuration was DistilBERT b42e60t512, trained with a batch size of 42, 60 epochs, and a maximum input length of 512 tokens. In this study, this model is not presented as a new experimental result, but as the starting component of the multi-agent architecture.

The detailed training settings, performance indicators, and rationale for selecting the baseline model are reported in previous publications. Accordingly, this paper does not repeat the full DistilBERT classification results. Instead, it uses the model

as a reproducible and previously validated base classifier agent. Its role was to generate the initial label, which was then evaluated by the LLM-based reasoning and validator agents and finally controlled by the rule-based decision agent.

In addition to the original 31-category label space, the present study also included experiments with a reduced label space. This setting consisted of the six most frequent categories and a separate exception category. The Top6 + Exception setup was developed specifically for the multi-agent validation experiment, not as a repetition of earlier baseline model studies. Its purpose was to analyze how the validation and decision components behave in a simplified, business-oriented label space where uncertain or non-main-category cases can be routed to a separate handling path.

The business exception logic was also evaluated in the Top6 + Exception label space. In this setting, the decision agent did not only accept or revise the baseline label but also determined whether the complaint could be assigned to one of the main categories with sufficient reliability or should be routed to exception handling. Thus, the reduced label space served not only as a technical simplification, but also as a decision-support control aligned with banking operations.

4. Multi-Agent Validation Framework

The multi-agent validation framework is based on the sequential interaction of the baseline classifier, the LLM-based reasoning and validation components, and the rule-based decision layer [8, 9]. Its purpose is not to let the generative model take over the closed-label classification task, but to examine and, when justified, restrictively modify the initial DistilBERT prediction under controlled conditions [4, 10, 11].

The framework relies on the complementary strengths of fine-tuned transformer classifiers and generative large language models [3, 4, 8, 9]. DistilBERT provides fast, reproducible predictions within a fixed label space, while LLMs can support content-based interpretation, alternative reasoning, and contextual validation of the decision [4, 8, 10]. Therefore, the proposed architecture does not use the LLM as the primary classifier. Instead, it integrates the LLM as a reasoning and validation layer that checks the baseline classifier output before any final decision is made [8, 9].

EARLY ACCESS

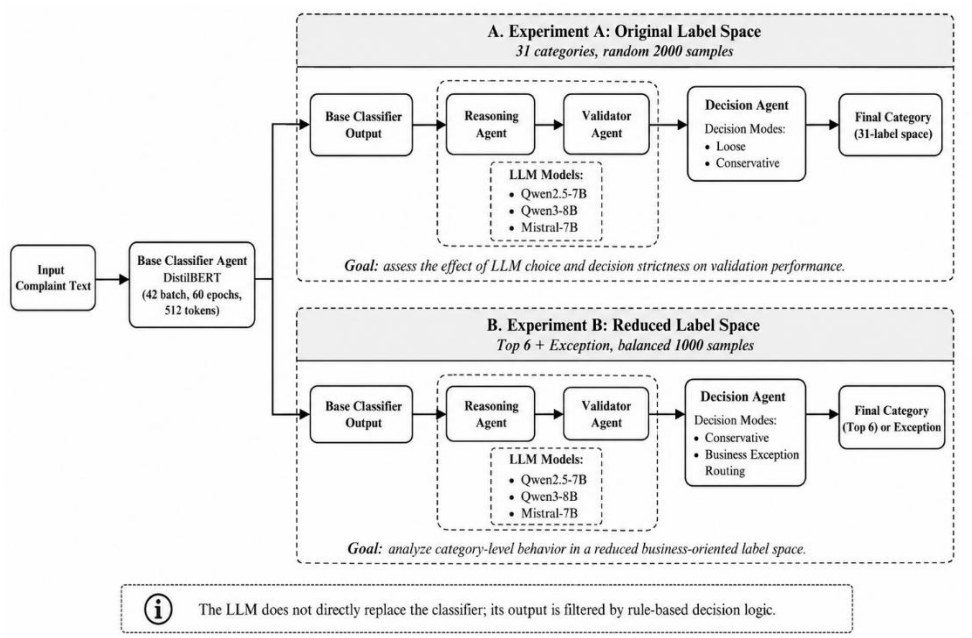


Figure 1. Multi-agent validation architecture for banking complaint classification with original and reduced label-space experiments

The first component of the system is the base classifier agent. It is built on the DistilBERT b42e60t512 model and produces the initial category prediction for each complaint text [4, 11]. Its operation is restricted to a closed label space: in the original setting, this refers to the 31 banking complaint categories, while in the reduced setting it refers to the Top6 + Exception label space [11].

The second component is the reasoning agent, which analyzes the complaint text and the baseline prediction. Its role is not to make the final decision, but to identify the main issue, relevant textual signals, possible category overlaps, and alternative classifications [8, 9]. This component adds an interpretive layer to the classification process, especially when a complaint refers to multiple issues or semantically close categories [1, 11].

The third component is the validator agent. It evaluates whether the baseline category is acceptable based on the complaint text and the reasoning output [8]. Its output may confirm the baseline label, suggest an alternative category, or indicate uncertainty. Thus, the validator agent does not simply relabel the case but checks the quality of the baseline decision [8, 10].

The fourth component is the decision agent, which determines the final output through rule-based logic. This component separates the proposed framework from simple LLM-based reclassification, because LLM suggestions cannot automatically overwrite the baseline prediction [8–10]. Instead, the decision agent evaluates whether the validation output provides sufficient justification for modification or whether the original DistilBERT label should be retained.

The decision rules supported three operating modes. The loose rule gave the LLM greater freedom to relabel cases and served as a negative control for testing the risk of overly permissive validation [10]. The conservative rule applied stricter conditions and was used to examine whether harmful relabeling could be reduced. The business exception mode was applied in the Top6 + Exception label space,

where the system could also route uncertain, non-main-category, or business-sensitive cases to exception handling [1, 11].

The validation process therefore follows four main steps: the base classifier produces the initial label; the reasoning agent provides content-based interpretation; the validator agent assesses the acceptability of the baseline prediction; and the decision agent determines whether the final outcome should be the baseline label, a revised category, or exception routing [8, 9].

A key feature of the framework is that the LLM has no unrestricted decision authority. Its reasoning and validation capabilities are embedded in a controlled architecture, where the final category decision is filtered by rule-based logic [8–10]. This is essential in banking, where incorrect relabeling may create operational and compliance risks [1, 11]. The proposed architecture therefore serves as an intermediate solution between fully automated classification and human review: the baseline classifier provides fast and reproducible predictions, the LLM agents add content-based control, and the decision agent regulates the intervention.

5. Experimental Design

The experimental design aimed to examine how the effectiveness of multi-agent validation depends on the selected large language model, the structure of the label space, and the rule system of the decision agent. All runs followed the same principle: the DistilBERT-based classifier produced the initial category prediction; the LLM-based reasoning and validator agents evaluated the complaint text and the baseline decision; and the decision agent determined the final output through rule-based logic.

The experiments were organized into two blocks. The first block examined multi-agent validation in the original 31-category label space. The second block used a reduced, business-oriented label space consisting of the six most frequent categories and a separate exception category. The two blocks served different purposes: the 31-category setting evaluated validation behavior in the full classification problem, while the Top6 + Exception setting focused on category-level behavior and the effect of exception routing.

The underlying dataset consists of Hungarian-language customer complaint records originating from a banking complaint-management environment. The classification task follows a predefined 31-category label taxonomy associated with the regulatory reporting framework of the “Magyar Nemzeti Bank” (MNB). The baseline model was developed using approximately 128,930 labeled training records and evaluated on a separate, held-out corpus containing 39,421 complaints.

The dataset is highly imbalanced. The four largest categories – “egyéb,” “megbízás teljesítése,” “elszámolás,” and “pénzügyi visszaélés” – account for approximately 87% of the held-out evaluation records, while several categories have substantially lower support. This distribution is particularly relevant when interpreting weighted and macro-level classification metrics.

The multi-agent experiments used two subsets drawn from the held-out evaluation corpus. Experiment A employed 2,000 randomly selected complaints in the original 31-category label space, preserving the naturally imbalanced structure. Experiment B employed 1,000 approximately class-balanced complaints under the

EARLY ACCESS

reduced Top6 + Exception label space. The same baseline predictions and sampled records were used across the corresponding LLM and decision-rule configurations within each experimental block.

5.1. Common Experimental Setup

All experiments used the same baseline model, DistilBERT b42e60t512, trained with a batch size of 42, 60 epochs, and a maximum input length of 512 tokens. In the multi-agent framework, this model served as the base classifier agent and produced the initial category prediction for each complaint text. The subsequent agents treated this output as a baseline decision to be validated.

The validation layer used three large language models: Qwen2.5-7B, Qwen3-8B, and Mistral-7B. The purpose was not to rank them as standalone classifiers, but to compare their validation behavior under the same baseline classifier. Accordingly, the LLMs were used as reasoning and validator agents, while the final decision was made by the decision agent.

The multi-agent validation workflow was implemented in Python. Data processing and result aggregation were performed using pandas, while PyTorch and Hugging Face Transformers supported local LLM inference [30]. Classification reports and aggregate evaluation metrics were generated using scikit-learn [31].

The reasoning and validator components used Qwen2.5-7B, Qwen3-8B, and Mistral-7B through model-specific tokenization and prompt formatting. Inference was performed locally using CUDA acceleration on an NVIDIA RTX 4090 GPU with 24 GB of memory. The agents produced structured validation outputs, which were processed by a deterministic Python-based decision agent implementing the loose, conservative, and business exception rules.

For each experimental run, the workflow recorded the baseline prediction, reasoning and validation outputs, decision source, and final category. Classification reports and additional CSV-based statistics were generated to support reproducibility and case-level analysis.

The decision agent operated with three rule settings. The *loose* rule allowed more LLM-proposed relabeling and served as a negative control for overly permissive validation. The *conservative* rule applied stricter conditions to reduce harmful interventions. *Business exception* routing was used only in the reduced Top6 + Exception label space, where selected complaints could be routed to exception handling instead of being forced into a main category.

5.2. Experiment A: Original 31-Category Label Space

The first experimental block was conducted in the original 31-category label space using 2,000 randomly selected complaint texts. Random sampling was used to evaluate multi-agent validation under the natural, imbalanced structure of the original dataset. This setting therefore reflects a more realistic banking environment, where category frequencies are not uniform.

In this block, each LLM was tested with two decision agent settings: loose and conservative. This resulted in six runs: Qwen2.5-7B loose, Qwen2.5-7B

conservative, Qwen3-8B loose, Qwen3-8B conservative, Mistral-7B loose, and Mistral-7B conservative. Since the base classifier agent was identical in all runs, the observed differences can be attributed to the validation behavior of the LLM and the applied decision rule.

The main research question in this block was whether LLM-based validation depends on the selected LLM and the rule system of the decision agent. The loose setting tested the risk of giving the LLM greater freedom to overwrite the baseline decision, while the conservative setting examined whether stricter validation conditions could reduce harmful relabeling. The structure of the 31-category experiments is summarized in Table 1.

5.3. Experiment B: Reduced Top6 + Exception Label Space

The second experimental block used a reduced label space consisting of the six most frequent categories and a separate exception category. The sample contained 1,000 balanced complaint texts. Balanced sampling was applied to evaluate the behavior of the main categories and the exception logic at the category level, rather than only through the dominant classes of the natural distribution.

The reduced label space was developed specifically for the multi-agent validation experiment and was not a simplified repetition of the earlier 31-category model experiments. The Top6 + Exception setting reflects the banking assumption that some complaints should not necessarily be forced into a main category. If a complaint does not fit any main category with sufficient reliability, or if the LLM-based validation indicates uncertainty, the decision agent may route the case to exception handling.

In this block, each LLM was tested with two decision settings: conservative and business exception routing. The conservative runs examined LLM validation under stricter intervention rules, while the business exception runs evaluated whether exception routing could separate uncertain or non-main-category complaints. The reduced label-space experiments are summarized in Table 1.

Table 1. Experimental setup for multi-agent validation

Experiment	Label space	Sample	LLMs	Decision rules
A	31 categories	Random 2000	Qwen2.5-7B, Qwen3-8B, Mistral-7B	Loose, Conservative
B	Top6 + Exception	Balanced 1000	Qwen2.5-7B, Qwen3-8B, Mistral-7B	Conservative, Business exception

5.4. Reproducibility Considerations

To support reproducibility, all runs used the same baseline classifier, input structure, and predefined decision agent logic. Only three factors varied across the experiments: the selected LLM, the decision rule, and the label space. This design ensured that the results were comparable and that observed differences were not caused by changes in the baseline model or input format.

EARLY ACCESS

In the 31-category block, the random 2,000-sample setting evaluated system behavior under the original class distribution. In the Top6 + Exception block, the balanced 1,000-sample setting supported category-level interpretation. For each run, the output files contained the baseline prediction, the LLM-based reasoning and validation output, the final decision agent output, and the final label. Evaluation was based on classification report metrics, including precision, recall, F1-score, and support, as well as additional statistical CSV files generated for each run.

Accordingly, the experimental design should not be interpreted as a general LLM benchmark, but as a controlled multi-agent validation study. By keeping the base classifier fixed and testing parallel LLM–decision rule combinations, the effects of LLM selection, decision strictness, and label-space reduction could be analyzed separately.

6. Evaluation Metrics

Evaluation of the multi-agent validation framework cannot be limited to standard classification metrics. Precision, recall, and F1-score measure final classification performance, but they do not show how the LLM-based reasoning and validator agents, together with the decision agent, intervened in the baseline classifier output [17, 21]. Therefore, the evaluation was conducted on three levels: classification report metrics, intervention metrics, and exception-oriented metrics in the reduced label space.

The evaluation followed the same logic in all runs. The DistilBERT-based base classifier agent produced the initial category prediction, and the multi-agent system then generated the final output through the reasoning, validator, and decision agents. Accordingly, the analysis examined not only whether the final label matched the true label, but also whether the system accepted the baseline decision, revised it, or routed the case to exception handling.

6.1. Classification Report Metrics

A classification report was generated for each run to measure basic classification performance. It included class-level precision, recall, F1-score, and support values, as well as aggregated metrics [17]. Precision measures how many cases assigned to a given category were correct, while recall indicates how many true cases of that category were identified. The F1-score is the harmonic mean of precision and recall and is especially useful when the two metrics differ [17, 21].

For class k, the metrics are defined as:

$$Precision_k = \frac{TP_k}{TP_k + FP_k}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k}$$

$$F1_k = 2 \cdot \frac{Precision_k \cdot Recall_k}{Precision_k + Recall_k}$$

where TP_k denotes correct positive classifications for class k, FP_k denotes cases incorrectly assigned to class k, and FN_k denotes cases that belonged to class k but were assigned to another category.

The support value shows the number of samples belonging to each category in the evaluation set. This is important because the natural distribution of the 31-category banking complaint dataset is strongly imbalanced. Therefore, accuracy alone is insufficient, as it may disproportionately reflect the performance of dominant classes [18–20].

Among the aggregated metrics, macro average and weighted average were emphasized. Macro average gives equal weight to each category and is therefore more sensitive to weak performance on rare classes. Weighted average, in contrast, weights categories by support and better reflects overall performance on the evaluation set [17, 18]. Both metrics are needed to assess whether multi-agent validation preserves performance not only for dominant categories, but also for less frequent ones.

In the 31-category block, the classification report was calculated for the full original label space. In the Top6 + Exception block, evaluation was performed on the reduced label space, where the exception category was treated as a separate evaluation class. In this setting, the classification report also indicated how the exception category functioned as a practical control mechanism for separating cases from the main decision path.

6.2. Baseline and Final Output Comparison

A central element of the evaluation was the comparison between the baseline prediction and the final system output. The baseline label refers to the initial category predicted by the DistilBERT-based classifier, while the final label refers to the decision produced after the complete multi-agent process. This comparison shows whether the reasoning, validator, and decision agents changed the baseline decision.

Let y_i denote the true label of sample i, $\hat{y}i^{base}$ the label predicted by the base classifier agent, and $\hat{y}i^{final}$ the final label produced by the multi-agent system. The correctness of the baseline decision can be defined as:

$$B_i = \begin{cases} 1, & \text{if } \hat{y}_i^{base} = y_i \\ 0, & \text{otherwise} \end{cases}$$

The correctness of the final decision is defined as:

$$F_i = \begin{cases} 1, & \text{if } \hat{y}_i^{final} = y_i \\ 0, & \text{otherwise} \end{cases}$$

This comparison is important because the multi-agent system operates as a validation and decision-support layer rather than as a standalone classifier. Therefore, evaluation must consider not only whether final performance improved or degraded, but also when LLM-based interventions corrected errors, introduced new errors, or preserved the baseline decision [17, 18].

EARLY ACCESS

6.3. Intervention Metrics

Intervention metrics were used to measure how often and how effectively the multi-agent system changed the baseline classifier's decisions. This is essential because frequent LLM-based modifications are not necessarily useful; the quality of interventions is as important as their quantity [17, 22–25].

The first metric is the change rate, which measures the proportion of cases in which the final label differed from the baseline label:

$$ChangeRate = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\hat{y}i^{final} \neq \hat{y}i^{base}\right) \quad (1)$$

A high change rate indicates more active intervention, while a low value indicates that the baseline prediction was usually retained. This interpretation is consistent with selective prediction and reject-option approaches, where the system is evaluated not only by its predictions, but also by its decision to intervene, abstain, or defer [22–24].

The quality of interventions was measured by beneficial and harmful changes. A beneficial change occurs when the baseline prediction is incorrect, but the final multi-agent decision is correct:

$$Beneficial_i = \mathbb{1}\left(\hat{y}i^{base} \neq y_i \wedge \hat{y}i^{final} = y_i\right) \quad (2)$$

A harmful change occurs when the baseline prediction is correct, but the final multi-agent decision becomes incorrect:

$$Harmful_i = \mathbb{1}\left(\hat{y}i^{base} = y_i \wedge \hat{y}i^{final} \neq y_i\right) \quad (3)$$

The corresponding rates are defined as:

$$BeneficialRate = \frac{\sum_{i=1}^{N} B\,eneficial_i}{N}$$

$$HarmfulRate = \frac{\sum_{i=1}^{N} H\,armful_i}{N}$$

The balance between useful and harmful interventions is captured by the net correction value:

$$NetCorrection = \sum_{i=1}^{N} B\,eneficial_i - \sum_{i=1}^{N} H\,armful_i$$

A positive value indicates that the multi-agent system corrected more errors than it introduced, while a negative value indicates that harmful relabeling outweighed beneficial corrections.

Finally, intervention precision measures the proportion of useful changes among all executed modifications:

$$InterventionPrecision = \frac{\sum_{i=1}^{N} B\,eneficial_i}{\sum_{i=1}^{N} \mathbb{1}\,(\hat{y}i^{final} \neq \hat{y}i^{base})}$$

This metric is particularly important when comparing loose and conservative decision rules. A loose rule may produce a higher change rate, but this is only favorable if many of the changes are beneficial. A conservative rule may intervene less often, but it can be preferable if it reduces the proportion of harmful changes [22–25].

6.4. Decision Rule Evaluation

The operation of the decision agent was evaluated as a separate analytical level. One of the main research questions was whether the effectiveness of LLM-based validation depends on the strictness of the decision rule. Therefore, the loose, conservative, and business exception settings were treated not merely as technical parameters, but as decision logics with different risk profiles [22–26].

For the loose decision rule, the key question was whether greater intervention freedom improves final classification performance or increases harmful relabeling. In this setting, change rate, harmful change rate, and net correction were especially important. If both change rate and harmful change are high, the loose rule can be considered risky.

For the conservative rule, the main question was whether stricter intervention conditions can preserve baseline stability while still correcting some incorrect baseline predictions. In this case, a lower change rate does not necessarily indicate weak performance. On the contrary, fewer but more useful interventions may represent a more favorable operating profile in a banking environment.

For business exception routing, evaluation was not limited to the correctness of revised categories. It was also important to analyze which cases were routed to exception handling and whether they truly failed to fit the main categories with sufficient reliability. This logic is related to reject-option and learning-to-defer approaches, where the system is allowed to avoid or defer uncertain decisions instead of forcing a potentially incorrect classification [22–26].

6.5. Exception Metrics

Exception metrics were used only in the Top6 + Exception experiments. In this setting, the exception category was not treated as a residual class, but as a decision-support control. Cases routed to exception handling were interpreted as complaints for which the system did not consider a main-category assignment sufficiently reliable, or which did not fit any of the Top6 categories within the reduced label space [22, 23, 27, 28].

The first exception metric is the exception rate:

$$ExceptionRate = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\hat{y}_i^{final} = Exception\right) \quad (4)$$

This metric shows the proportion of samples routed to exception handling. A very low exception rate may indicate that the system rarely signals uncertainty, while a

EARLY ACCESS

very high exception rate may suggest that the decision agent is overly cautious and removes too many cases from automated main-category classification [22, 23, 27].

Exception precision measures how many cases routed to exception handling truly belonged to the exception category:

$$ExceptionPrecision = \frac{TP_{Exception}}{TP_{Exception} + FP_{Exception}}$$

Exception recall shows how many true exception cases were successfully routed to the exception branch:

$$ExceptionRecall = \frac{TP_{Exception}}{TP_{Exception} + FN_{Exception}}$$

Exception F1-score is the harmonic mean of these two metrics:

$$ExceptionF1 = 2 \cdot \frac{ExceptionPrecision \cdot ExceptionRecall}{ExceptionPrecision + ExceptionRecall}$$

These metrics are important because the value of business exception routing is not determined by routing as many complaints as possible to exception handling. Its purpose is to separate cases that are genuinely uncertain, outside the main categories, or require further review [22, 23, 27, 28]. Effective exception handling is therefore characterized by an appropriate precision–recall balance, not by a high exception rate alone [17, 21].

In the Top6 + Exception experiments, the six main categories were also evaluated separately. This was necessary because an exception-oriented decision logic is not useful if it identifies exception cases while substantially degrading precision or recall in the main categories. Therefore, system performance was assessed jointly in terms of exception handling and main-category stability [17, 18, 22, 27].

6.6. Comparative Evaluation Across LLMs and Decision Rules

The final level of evaluation compared the LLMs and decision rule settings. In the 31-category label space, Qwen2.5-7B, Qwen3-8B, and Mistral-7B were evaluated with both loose and conservative decision logic [13–15]. The aim was to determine how their validation behavior differed and how decision-rule strictness affected final performance and intervention patterns [8, 9, 22, 27].

In the Top6 + Exception label space, the comparison focused on the conservative and business exception routing settings. The main question was which LLM–decision rule combination produced more favorable category-level behavior, and how effectively the business exception logic separated uncertain or non-main-category cases [22, 23, 27, 28]. The metric groups used for the comparative evaluation are summarized in Table 2.

Table 2. Evaluation metric groups for assessing classification performance, intervention behavior, and exception routing in the multi-agent validation

framework

Metric group	Main indicators	Purpose
Classification metrics	Precision, recall, F1-score, support, accuracy, macro average, weighted average	Final classification performance
Intervention metrics	Change rate, beneficial change, harmful change, net correction, intervention precision	Effect of LLM-based intervention
Decision rule metrics	Accepted baseline labels, revised labels, rejected revisions	Behavior of rule-based control
Exception metrics	Exception rate, exception precision, exception recall, exception F1-score	Performance of exception routing
Comparative metrics	Difference from baseline, difference across LLMs, difference across rules	Model- and rule-level comparison

This evaluation structure ensures that the multi-agent system is not assessed solely on final accuracy metrics. Its practical value depends on whether it can correct baseline classifier errors in a controlled way without disproportionately increasing harmful interventions. In banking complaint classification, the most important evaluation criterion is therefore not a higher F1-score alone, but the joint assessment of performance, intervention risk, and exception-handling stability.

7. Results

The results of the multi-agent validation framework were evaluated in two experimental blocks. The first block examined the effect of LLM-based validation in the original 31-category label space using 2,000 randomly selected samples. The second block analyzed conservative validation and business exception routing in the reduced Top6 + Exception label space using 1,000 balanced samples. In addition to standard classification report metrics, the evaluation included case-level comparison, decision source statistics, and exception routing indicators.

When interpreting the results, it is important to emphasize that the multi-agent system was not used as a standalone classifier, but as a validation and decision layer operating above the DistilBERT-based baseline predictions. Therefore, the analysis focused not only on whether the final weighted F1-score increased or decreased, but also on how often the system intervened, whether those interventions improved or degraded the baseline decisions, and how effectively the decision agent controlled the LLM suggestions.

7.1. Results on the Original 31-Category Label Space

In the 31-category label space, the DistilBERT baseline achieved 68.85% accuracy, 69.20% weighted precision, 68.85% weighted recall, and 68.70% weighted F1-score on the same 2,000-sample evaluation set. This served as the reference point for all multi-agent runs.

The most stable results were obtained with the conservative decision rule. The Mistral-7B conservative run fully preserved the baseline performance: accuracy remained 68.85%, weighted F1-score remained 68.70%, and all aggregate deltas were 0.00%. At the case level, both the baseline and the multi-agent output were

EARLY ACCESS

correct in 1,377 cases and both were incorrect in 623 cases. No improved or degraded cases occurred:

$$NetCorrection = 0 - 0 = 0$$

This indicates that the decision agent did not allow any LLM-based modification to change the final label.

The Qwen3-8B conservative run also remained very close to the baseline. Accuracy decreased only to 68.75%, while weighted F1-score was 68.57%, corresponding to a 0.13 percentage-point decrease in weighted F1. At the case level, 1,375 cases were correct in both systems, 623 were incorrect in both, and only two harmful changes occurred:

$$NetCorrection = 0 - 2 = -2$$

The decision source statistics confirm a strongly baseline-preserving behavior: 91.25% of decisions accepted the baseline label, while only 0.25% involved a high-confidence validator-based modification.

The Qwen2.5-7B conservative run showed a larger decline. Accuracy decreased to 65.80% and weighted F1-score to 66.40%, representing a 2.30 percentage-point weighted F1 decrease. Although eight baseline errors were corrected, 69 previously correct baseline decisions were degraded:

$$NetCorrection = 8 - 69 = -61$$

This shows that the conservative rule did not filter harmful interventions equally well across all LLMs.

The loose decision rule produced substantially less favorable results. With Qwen2.5-7B, accuracy fell to 44.20% and weighted F1-score to 47.62%. The system corrected 46 baseline errors, but degraded 539 originally correct baseline decisions:

$$NetCorrection = 46 - 539 = -493$$

Mistral-7B showed a similar pattern. Accuracy decreased to 48.35% and weighted F1-score to 48.06%, with 50 beneficial and 460 harmful changes:

$$NetCorrection = 50 - 460 = -410$$

In these runs, the share of loose-rule validator modifications was high: 41.00% for Qwen2.5-7B and 34.50% for Mistral-7B. This explains the performance loss, as the permissive rule allowed too many LLM suggestions to overwrite correct baseline decisions.

In contrast, the Qwen3-8B loose run caused only a minor decrease. Accuracy was 68.50% and weighted F1-score was 68.52%, only 0.18 percentage points below the baseline. At the case level, five beneficial and twelve harmful changes occurred:

$$NetCorrection = 5 - 12 = -7$$

The *validator_modified_loose_rule* rate was only 1.35%, showing that Qwen3-8B did not become aggressive even under the loose rule. Overall, in the 31-category setting, Qwen3-8B behaved as a substantially more stable LLM validator than Qwen2.5-7B or Mistral-7B under permissive decision conditions.

Table 3. Baseline and multi-agent performance in the original 31-category label space

Configuration	Accuracy	Weighted precision	Weighted recall	Weighted F1	Δ Weighted F1
DistilBERT baseline	68.85%	69.20%	68.85%	68.70%	—
Mistral-7B conservative	68.85%	69.20%	68.85%	68.70%	0.00%
Qwen3-8B conservative	68.75%	68.89%	68.75%	68.57%	-0.13%
Qwen3-8B loose	68.50%	69.10%	68.50%	68.52%	-0.18%
Qwen2.5-7B conservative	65.80%	67.69%	65.80%	66.40%	-2.30%
Mistral-7B loose	48.35%	58.06%	48.35%	48.06%	-20.64%
Qwen2.5-7B loose	44.20%	60.34%	44.20%	47.62%	-21.08%

Based on the 31-category results, multi-agent validation did not improve aggregate performance over the DistilBERT baseline. However, the runs clearly demonstrated the control role of the decision rule. The conservative rule prevented harmful interventions in several configurations, while the loose rule led to substantial performance degradation with Qwen2.5-7B and Mistral-7B. This confirms that, in closed-label banking complaint classification, LLM suggestions should not be transferred to the final decision without strict rule-based control.

7.2. Intervention Metrics on the 31-Category Label Space

The intervention metrics provide a more detailed view of how the multi-agent system changed the baseline decisions. Change rate measures the proportion of cases in which the final label differed from the baseline label (Eq. 1). Based on the decision source statistics, the actual LLM-based modification rate was highest in the loose-rule runs. In the Qwen2.5-7B loose setting, 820 cases were modified under the loose rule, corresponding to 41.00% of the sample. In the Mistral-7B loose setting, the same rate was 34.50%. By contrast, the Qwen3-8B loose run produced only 27 such modifications, corresponding to 1.35%.

The quality of the interventions was evaluated using beneficial and harmful change indicators (Eq. 2, Eq. 3). In the Qwen2.5-7B loose run, 46 beneficial changes were accompanied by 539 harmful interventions. In the Mistral-7B loose run, 50 beneficial changes occurred alongside 460 harmful interventions. These results show that the loose decision rule did not merely increase the number of interventions; in these configurations, most interventions contributed to the degradation of baseline performance.

EARLY ACCESS

Table 4. Intervention behavior in the original 31-category label space

Configuration	Modified or accepted LLM changes	Beneficial changes	Harmful changes	Net correction
Mistral-7B conservative	0	0	0	0
Mistral-7B loose	690	50	460	-410
Qwen2.5-7B conservative	104	8	69	-61
Qwen2.5-7B loose	820	46	539	-493
Qwen3-8B conservative	5	0	2	-2
Qwen3-8B loose	27	5	12	-7

The table confirms that intervention volume alone is not a favorable property. The Qwen2.5-7B and Mistral-7B loose runs executed many modifications, but most of them were harmful. In contrast, Qwen3-8B intervened only rarely, which kept performance degradation low. The Mistral-7B conservative run fully preserved the baseline output, indicating that the decision agent filtered out all potentially risky modifications.

7.3. Results on the Reduced Top6 + Exception Label Space

In the reduced Top6 + Exception label space, the DistilBERT baseline achieved 58.50% accuracy, 65.08% weighted precision, 58.50% weighted recall, and 58.18% weighted F1-score. The conservative Mistral-7B and Qwen3-8B runs fully preserved this baseline performance. In both cases, 585 samples were correct in both systems and 415 were incorrect in both, with no improved or degraded cases. Thus, these configurations showed complete baseline-preserving behavior in the reduced label space.

The Qwen2.5-7B conservative run showed a moderate decline. Accuracy decreased to 56.40%, while weighted F1-score decreased to 55.69%, corresponding to a 2.49 percentage-point drop in weighted F1. At the case level, eight beneficial and 29 harmful changes occurred:

$$NetCorrection = 8 - 29 = -21$$

This pattern is consistent with the 31-category Qwen2.5-7B conservative run, where the same model also introduced more harmful than beneficial interventions.

The business exception routing runs produced a different pattern. Qwen2.5-7B achieved 30.30% accuracy and 27.77% weighted F1-score, while Mistral-7B achieved 34.90% accuracy and 33.85% weighted F1-score. In both cases, weighted precision remained high, above 72%, but recall and F1-score dropped sharply. This indicates that the business exception logic became overly restrictive: it favored precision but routed too many cases away from the correct main category.

By contrast, Qwen3-8B business exception routing was substantially more stable. Accuracy was 56.70% and weighted F1-score was 56.50%, only 1.68 percentage

points below the baseline weighted F1-score. This confirms the pattern observed in the 31-category block: Qwen3-8B behaved as a less aggressive and more stable validator than Qwen2.5-7B or Mistral-7B.

Table 5. Baseline and multi-agent performance in the reduced Top6 + Exception label space

Configuration	Accuracy	Weighted precision	Weighted recall	Weighted F1	Δ Weighted F1
DistilBERT baseline	58.50%	65.08%	58.50%	58.18%	—
Mistral-7B conservative	58.50%	65.08%	58.50%	58.18%	0.00%
Qwen3-8B conservative	58.50%	65.08%	58.50%	58.18%	0.00%
Qwen3-8B business exception	56.70%	64.75%	56.70%	56.50%	-1.68%
Qwen2.5-7B conservative	56.40%	61.80%	56.40%	55.69%	-2.49%
Mistral-7B business exception	34.90%	72.84%	34.90%	33.85%	-24.33%
Qwen2.5-7B business exception	30.30%	73.12%	30.30%	27.77%	-30.41%

In the reduced label space, the conservative decision rule again proved more stable, while business exception routing showed strong model dependence. With Qwen3-8B, exception routing caused only moderate performance degradation. In contrast, Qwen2.5-7B and Mistral-7B routed too many cases to the exception branch, resulting in substantial recall and F1-score losses.

7.4. Exception Routing Results

For a detailed evaluation of exception routing, both the number and correctness of routed cases were analyzed. In the business exception setting, the exception rate is defined as Eq. 4. In the Qwen2.5-7B business exception run, 643 cases were routed to the exception branch, corresponding to an exception rate of 64.30%. Of these, 67 were correctly routed and 576 were incorrectly routed. In the Mistral-7B run, 577 cases, or 57.70% of the sample, were routed to exception handling; 65 were correct and 512 were incorrect. By contrast, Qwen3-8B routed only 49 cases to exception handling, corresponding to 4.90%; 7 of these were correct and 42 were incorrect.

Routing precision for actively routed exception cases was calculated as:

$$RoutingPrecision = \frac{CorrectlyRoutedToException}{RoutedToException}$$

For Qwen2.5-7B, this gives:

$$\frac{67}{643} \approx 10.42$$

For Mistral-7B:

EARLY ACCESS

$$\frac{65}{577} \approx 11.27$$

For Qwen3-8B:

$$\frac{7}{49} \approx 14.29$$

These results show that business exception routing, in its current form, did not identify exception cases with sufficient precision. Qwen2.5-7B and Mistral-7B routed too many cases to the exception branch, most of which were incorrect. Qwen3-8B operated with a much lower exception rate, but its routing precision also remained limited.

Table 6. Exception routing behavior in the reduced Top6 + Exception label space

Configuration	Routed to exception	Exception rate	Correctly routed	Wrongly routed	Routing precision
Qwen2.5-7B business exception	643	64.30%	67	576	10.42%
Mistral-7B business exception	577	57.70%	65	512	11.27%
Qwen3-8B business exception	49	4.90%	7	42	14.29%

The business exception routing results indicate that the current exception-handling logic requires further refinement. For Qwen2.5-7B and Mistral-7B, the decision agent behaved too cautiously and routed many main-category cases to the exception branch. With Qwen3-8B, routing was much more restrained, and aggregate performance therefore remained closer to the baseline. This suggests that exception routing depends not only on the decision rule, but also strongly on the validation behavior of the selected LLM.

7.5. Comparative Interpretation

Three main conclusions can be drawn from the results. First, multi-agent validation did not improve the aggregate DistilBERT baseline performance in this experimental setting. The best runs either fully preserved the baseline or caused only minor degradation. However, this does not make the framework irrelevant. Its main value lies in making the intervention risk of LLM-based validation measurable and showing that the decision agent plays a key role in filtering harmful modifications.

Second, the loose decision rule proved risky in the 31-category label space. For Qwen2.5-7B and Mistral-7B, it allowed many LLM-based modifications, most of which were harmful. This supports the assumption that, in closed-label banking classification, generative model suggestions should not be applied through unrestricted relabeling.

Third, Qwen3-8B showed the most stable validation behavior in both experimental blocks. In the 31-category setting, the difference between loose and conservative rules was minimal, while in the Top6 + Exception setting, business exception routing caused substantially less performance loss than with Qwen2.5-7B or Mistral-7B. This indicates that Qwen3-8B intervened less aggressively and was better aligned with the controlled validation role.

Overall, the results answer the main research question by showing that the effectiveness of LLM-based validation depends on both the selected LLM and the decision agent rule system. The effect of the rule system was particularly strong: with the same LLM, the difference between loose and conservative settings exceeded 20 percentage points in weighted F1-score in some cases. Therefore, in banking complaint classification, multi-agent validation should be used only with a controlled, rule-based decision layer.

8. Discussion

The experimental results show that the main value of multi-agent validation was not a general improvement over the DistilBERT baseline. In several runs, the baseline remained more stable than the LLM-modified output, especially when the decision agent applied a permissive relabeling rule. The key finding is therefore methodological: in closed-label banking complaint classification, LLM-based validation can only be used reliably if its suggestions are filtered by a controlled decision rule.

8.1. The Risk of Loose Relabeling

Loose relabeling proved to be the riskiest operating mode. In a closed-label task, the objective is not free textual interpretation, but the consistent application of a predefined category system. Although an LLM may provide a plausible alternative interpretation, this alternative does not necessarily match the labeling logic of the training data or the operational meaning of the banking categories.

The 31-category results clearly demonstrated this risk. In the loose Qwen2.5-7B and Mistral-7B runs, the decision agent allowed a large number of LLM-based modifications, but most of them were harmful. As shown in the intervention analysis, both configurations produced strongly negative net correction values. This indicates that greater intervention freedom did not improve performance; instead, it increased the probability of overwriting correct baseline decisions.

This is especially problematic in banking complaint handling. A wrong category is not only a technical prediction error: it may affect case routing, internal reporting, compliance monitoring, and the quality of regulatory data. Therefore, unrestricted LLM-based relabeling is not appropriate for this type of task.

8.2. The Role of Conservative Validation

The conservative decision rule provided a more stable operating profile. Its purpose is not to maximize the number of corrected cases, but to reduce the probability of harmful intervention. In a banking environment, this is a reasonable design principle, because the cost of degrading a correct baseline decision may be higher

EARLY ACCESS

than the benefit of correcting a small number of baseline errors.

The conservative Mistral-7B run fully preserved the DistilBERT baseline in the 31-category setting, while the same model under the loose rule caused substantial degradation. This contrast shows that the decision rule can act as an effective control layer over LLM suggestions. The Qwen3-8B conservative run also remained very close to the baseline, with only minimal degradation and a strongly baseline-preserving decision pattern.

However, the Qwen2.5-7B conservative results also show that conservative validation is not sufficient by itself. Even under stricter rules, this model introduced more harmful than beneficial changes. Thus, the success of the architecture depends on both components: the strictness of the decision agent and the validation behavior of the selected LLM.

8.3. Business Meaning of Exception Routing

Business exception routing follows a different logic from simple relabeling. Its aim is not only to accept or revise the baseline category, but also to identify cases that should be routed outside the automatic main-category decision path. This is meaningful in banking complaint handling, where some cases are ambiguous, involve several issues, or require further review.

The Top6 + Exception setting should therefore be interpreted as a decision-support scenario, not merely as a reduced classification task. The six main categories represent frequent operational cases, while the exception category provides a controlled exit path for complaints that do not fit the main categories with sufficient reliability.

The results show that this logic requires further calibration. Qwen2.5-7B and Mistral-7B routed too many cases to exception handling, which caused substantial recall and F1-score losses. Qwen3-8B produced a more restrained exception-routing pattern and remained much closer to the baseline. This again confirms that exception routing depends not only on the rule design, but also on the validation style of the underlying LLM.

From a business perspective, exception routing is useful only if it separates genuinely uncertain or risky cases. A very high exception rate may overload human review processes and reduce the operational value of automation. A very low exception rate, however, may fail to identify cases that should not be handled automatically. The practical goal is therefore not to maximize exception routing, but to find an appropriate balance between exception precision, recall, and main-category stability.

8.4. Implications for Banking Complaint Classification

The results suggest that multi-agent validation should not be treated as a simple performance-enhancement technique. Its primary value lies in risk-controlled decision support. LLMs may help interpret complaint texts, identify possible category conflicts, and support validation, but their outputs should not directly determine the final category.

This has practical relevance for banking applications. Automated complaint classification must be accurate, but it must also be predictable, auditable, and operationally safe. A system that frequently changes baseline decisions may appear more intelligent at the textual level, while still reducing aggregate and case-level performance. Therefore, intervention behavior must be evaluated together with standard classification metrics.
A conservative multi-agent architecture offers a more suitable direction. In this setup, the LLM does not replace the classifier, but acts as a validation component. The decision agent then determines whether the LLM suggestion is strong enough to influence the final output. This creates a more controlled architecture between fully automated classification and human review.

8.5. Limitations and Future Refinement

A limitation of the present experiments is that the multi-agent framework did not produce a general aggregate improvement over the DistilBERT baseline. This does not invalidate the approach, but it shows that LLM-based validation requires careful rule design and empirical calibration. Loose rules may cause uncontrolled relabeling, while overly cautious exception routing may remove too many cases from the automatic decision path.

Future work should therefore refine the decision agent. Instead of a simple accept-or-reject logic, the rule system could incorporate additional signals, such as the LLM validation confidence, the baseline model confidence, category frequency, and the risk profile of the proposed category change. A modification from a rare category to a frequent category, for example, may require different control conditions than a change between two frequent categories.

Exception routing also requires further calibration. Routing should depend on a combination of signals, such as LLM uncertainty, low baseline confidence, and weak category fit. This may reduce false exception routing while preserving the business value of separating cases that require further review.

9. Conclusion

This study examined a multi-agent validation framework for Hungarian banking complaint classification. The framework used a previously selected DistilBERT model as the base classifier agent, while Qwen2.5-7B, Qwen3-8B, and Mistral-7B were applied as reasoning and validator agents. A rule-based decision agent determined whether the baseline label should be retained, revised, or routed to exception handling.

The results show that LLM-based validation should not be treated as an automatic performance-improvement mechanism. In the 31-category setting, unrestricted or loose relabeling often degraded the DistilBERT baseline, especially with Qwen2.5-7B and Mistral-7B. Conservative decision rules produced a more stable operating profile, confirming that LLM suggestions require strict rule-based control in closed-label banking classification.

The reduced Top6 + Exception experiments further showed that business exception routing is a meaningful decision-support concept, but it requires careful calibration. Qwen2.5-7B and Mistral-7B routed too many cases to the exception branch, while

EARLY ACCESS

Qwen3-8B showed a more restrained and stable validation pattern. This indicates that the effectiveness of the framework depends jointly on the selected LLM and the decision rule applied by the decision agent.

The main contribution of the study is that it evaluates generative LLMs not as standalone classifiers, but as controlled validation components within a multi-agent architecture. This distinction is important in regulated domain-specific NLP tasks, where free-form relabeling is not an acceptable operating mode. The proposed framework separates baseline prediction, LLM-based reasoning, validation, and final rule-based decision-making.

Overall, the findings suggest that multi-agent LLM validation can support banking complaint classification only in a controlled form. It does not replace a fine-tuned closed-label classifier, and without decision control it may reduce performance. However, with a properly designed decision agent, it can serve as a validation and decision-support layer that helps identify uncertain cases, limit harmful relabeling, and prepare selected complaints for further human or business review.

Future work should focus on refining the decision agent. Further experiments should incorporate baseline confidence, LLM validation confidence, category frequency, and category-pair risk into the intervention logic. Exception routing also requires additional calibration to separate genuinely uncertain or risky complaints without creating excessive manual review workload.

References

[1] Coussement, K. and Van den Poel, D.: Improving customer complaint management by automatic email classification using linguistic style features as predictors. Decision Support Systems, 44(4), pp. 870–882, 2008. https://doi.org/10.1016/j.dss.2007.10.010

[2] Choe, P., Lehto, M. R., Shin, G. C. and Choi, K. Y.: Semiautomated identification and classification of customer complaints. Human Factors and Ergonomics in Manufacturing & Service Industries, 23(2), pp. 149–162, 2013. https://doi.org/10.1002/hfm.20325

[3] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019, pp. 4171–4186, 2019. https://doi.org/10.18653/v1/N19-1423

[4] Sanh, V., Debut, L., Chaumond, J. and Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv, 2019. https://doi.org/10.48550/arXiv.1910.01108

[5] Sun, C., Qiu, X., Xu, Y. and Huang, X.: How to Fine-Tune BERT for Text Classification? In: Chinese Computational Linguistics, Lecture Notes in Computer Science, vol. 11856, pp. 194–206, Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-32381-3_16

[6] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M. and Gao, J.: Deep Learning Based Text Classification: A Comprehensive Review. ACM Computing Surveys, 54(3), Article 62, 2021. https://doi.org/10.1145/3439726

[7] Han, M., Li, A., Gao, Z., Mu, D. and Liu, S.: A survey of multi-class imbalanced data classification methods. Journal of Intelligent & Fuzzy Systems, 44(3), pp. 4581–4599, 2023. https://doi.org/10.3233/JIFS-221902

[8] Gu, J., Jiang, X., Shi, Z., Tian, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Lin, Z., Zhang, B., Ni, L. M.-S., Gao, W., Wang, Y. and Guo, J.: A survey on LLM-as-a-judge. The Innovation, 7(6), Article 101253, 2026. https://doi.org/10.1016/j.xinn.2025.101253

[9] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O. and Zhang, X.: Large Language Model Based Multi-agents: A Survey of Progress and Challenges. Proceedings of IJCAI 2024, pp. 8048–8057, 2024. https://doi.org/10.24963/ijcai.2024/890

[10] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B. and Liu, T.: A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ACM Transactions on Information Systems, 43(2), Article 42, 2025. https://doi.org/10.1145/3703155

[11] Krutilla, Z.: DistilBERT-based Classification of Hungarian Banking Complaints under Regulatory Constraints. Acta Polytechnica Hungarica, 23(7), pp. 21–41, 2026. https://doi.org/10.12700/APH.23.7.2026.7.2

[12] Krutilla, Z.: Stability, Efficiency, and Language Specificity in Transformer-Based Hungarian Banking NLP. In: CANDO-EPE 2026 - IEEE 8th International Conference and Workshop in Obuda on Electrical and Power Engineering, Proceedings, pp. 325–328, 2026. https://doi.org/10.1109/CANDO-EPE71091.2026.11569408

[13] Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., et al.: Qwen2.5 Technical Report. arXiv, 2024. https://doi.org/10.48550/arXiv.2412.15115

[14] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., et al.: Qwen3 Technical Report. arXiv, 2025. https://doi.org/10.48550/arXiv.2505.09388

[15] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. de las, Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., et al.: Mistral 7B. arXiv, 2023. https://doi.org/10.48550/arXiv.2310.06825

FAIR ACCESS

[16] Ács, J., Lévai, D., Nemeskey, D. M. and Kornai, A.: Evaluating Contextualized Language Models for Hungarian. arXiv, 2021. https://doi.org/10.48550/arXiv.2102.10848

[17] Sokolova, M. and Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), pp. 427–437, 2009. https://doi.org/10.1016/j.ipm.2009.03.002

[18] He, H. and Garcia, E. A.: Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering, 21(9), pp. 1263–1284, 2009. https://doi.org/10.1109/TKDE.2008.239

[19] Japkowicz, N. and Stephen, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis, 6(5), pp. 429–449, 2002. https://doi.org/10.3233/IDA-2002-6504

[20] Wong, T.-T.: Linear Approximation of F-Measure for the Performance Evaluation of Classification Algorithms on Imbalanced Data Sets. IEEE Transactions on Knowledge and Data Engineering, 34(2), pp. 753–763, 2022. https://doi.org/10.1109/TKDE.2020.2986749

[21] Goutte, C. and Gaussier, E.: A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In: Advances in Information Retrieval, ECIR 2005, Lecture Notes in Computer Science, vol. 3408, pp. 345–359, Springer, Berlin, Heidelberg, 2005. https://doi.org/10.1007/978-3-540-31865-1_25

[22] Chow, C. K.: On optimum recognition error and reject tradeoff. IEEE Transactions on Information Theory, 16(1), pp. 41–46, 1970. https://doi.org/10.1109/TIT.1970.1054406

[23] Herbei, R. and Wegkamp, M. H.: Classification with reject option. The Canadian Journal of Statistics, 34(4), pp. 709–721, 2006. https://doi.org/10.1002/cjs.5550340410

[24] Mozannar, H. and Sontag, D.: Consistent Estimators for Learning to Defer to an Expert. Proceedings of the 37th International Conference on Machine Learning, PMLR, 119, pp. 7076–7087, 2020. https://doi.org/10.48550/arXiv.2006.01862

[25] Bondi, E., Koster, R., Sheahan, H., Chadwick, M., Bachrach, Y., Cemgil, T., Paquet, U. and Dvijotham, K.: Role of Human-AI Interaction in Selective Prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 36(5), pp. 5286–5294, 2022. https://doi.org/10.1609/aaai.v36i5.20465

[26] Hemmer, P., Thede, L., Vössing, M., Jakubik, J. and Kühl, N.: Learning to Defer with Limited Expert Predictions. Proceedings of the AAAI Conference on Artificial Intelligence, 37(5), pp. 6002–6011, 2023. https://doi.org/10.1609/aaai.v37i5.25742

[27] Hendrickx, K., Perini, L., Van der Plas, D., Meert, W. and Davis, J.: Machine learning with a reject option: a survey. Machine Learning, 113, pp. 3073–3110, 2024. https://doi.org/10.1007/s10994-024-06534-x

[28] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V. and Nahavandi, S.: A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion, 76, pp. 243–297, 2021. https://doi.org/10.1016/j.inffus.2021.05.008

[29] L. Zheng et al., “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena,” Advances in Neural Information Processing Systems, vol. 36, 2023. https://doi.org/10.52202/075280-2020.

[30] T. Wolf et al., “Transformers: State-of-the-Art Natural Language Processing,” in Proc. EMNLP 2020: System Demonstrations, pp. 38–45, 2020. DOI: https://doi.org/10.18653/v1/2020.emnlp-demos.6.

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