Semantic Vectorization and LLMs for Predicting Emergency Fire Service Activity from Weather Bulletins

Naoufal Sirri*[1] and Christophe Guyeux[2]

FEMTO-ST Institute, UMR 6174 CNRS, University of Franche-Comté, Belfort, France, {naoufal.sirri,christophe.guyeux}@univ-fcomte.fr

Abstract. We investigate whether Large Language Models (LLMs) augmented with Retrieval-Augmented Generation (RAG) can improve the prediction of firefighter intervention activity from French weather vigilance bulletins. Using a decade-long dataset (2015–2025) of 335,820 intervention records from the Doubs Fire and Rescue Service (SDIS 25) paired with Météo-France vigilance bulletins, we semantically vectorize textual bulletins using two embedding models (Mistral and OpenAI), store them in a Milvus vector database, and compare three families of predictors: (i) classical machine learning models (Linear Regression, Random Forest, Support Vector Regression) trained on raw embeddings; (ii) a retrieval-based k-NN baseline that averages intervention counts of the most similar past bulletins; and (iii) LLMs (GPT and Mistral) conditioned via RAG on the same retrieved context. On the test set, GPT with Mistral embeddings reaches the best scores (RMSE = 2.30, MAE = 2.10, $R^2 = 0.84$), ahead of the k-NN retrieval baseline ($R^2 = 0.72$) and of embedding-only ML models ($R^2 \leq 0.60$, down to $R^2 = 0.06$ for SVR). The gain of LLM+RAG over k-NN retrieval ($\Delta R^2 \approx 0.12$) is of the same order as the gain of retrieval over the best embedding-only baseline ($\Delta R^2 = 0.12$ against RF), and much larger when compared to weaker embedding-only baselines. Beyond point predictions, LLMs produce textual justifications that could support operator interpretation, though we do not evaluate this utility with end-users. We discuss limitations, including protocol caveats (split strategy, retrieval leakage control, LLM stochasticity), cost/latency trade-offs, and dependency on proprietary APIs.

Keywords: Firefighter interventions · Vector embeddings · Retrieval-augmented generation (RAG) · Machine learning · Large language models (LLMs) · Prompt engineering

1 Introduction

Predicting the operational load of fire and rescue services is a practical problem with direct implications for resource allocation and response times. Weather conditions are one of its main drivers: storms, heatwaves, frost episodes and heavy

rainfall each shift the type and volume of interventions a service has to handle. When a vigilance bulletin is issued by a national weather agency, the service must decide how much to pre-position; an inaccurate anticipation leads either to under-deployment during a peak or to wasted resources during a false alarm. Machine learning models, including large language models (LLMs) [1], offer a way to turn these textual bulletins into numerical predictions. We study this problem over the 2015–2025 decade, comparing classical machine learning algorithms on raw text embeddings, a retrieval-based k-NN baseline that averages the intervention counts of semantically similar past events, and LLMs (Mistral and GPT) conditioned on the same retrieved context via a RAG pipeline.

Several recent studies have applied generative models to prediction tasks on unstructured inputs, in domains adjacent to ours. [2] evaluated multiple LLMs for mental health prediction from online text, reporting gains from instruction fine-tuning but also persistent ethical and robustness concerns. [3] applied LLMs to click-through rate prediction over long textual user behaviour sequences and introduced BAHE, a hierarchical encoding architecture that cuts training cost in an industrial setting. [4] studied LLM assistance of human forecasters, showing that even imperfect assistants can improve predictive accuracy on cognitively demanding tasks. Closer to our setting, [5] proposed LC-LLM, a chain-of-thought LLM for lane-change intention and trajectory prediction in autonomous driving, with interpretable textual outputs.

The broader landscape of AI in emergency services has been recently reviewed in [7], which surveys machine learning and deep learning applications across call triage, dispatch optimisation and resource allocation. Work focused specifically on firefighter intervention prediction is still sparse, and the use of generative models in this domain has barely been explored. [6] used natural language processing on weather bulletins and reported accuracies of 80–92% in forecasting intervention peaks triggered by rare events. More recently, [9] showed that LLM-generated meteorological features can improve intervention models, especially for high-activity periods, while noting limitations on low-frequency events. Earlier work by the same group has assessed the reliability of such prediction systems for resource allocation [10, 11] and looked at the influence of air quality, solar activity, river levels, epidemics and weather on firefighter operations [12–16].

We compare three predictive paradigms on the same vectorized bulletins: classical machine learning on raw embeddings, a k-NN retrieval baseline, and LLMs conditioned on retrieved context via a RAG pipeline. Our aim is to attribute predictive accuracy to the right component of the stack (embedding, retrieval, or LLM reasoning) rather than treating the pipeline as a black box.

Contributions. The main contributions of this work are the following:

- We build, to the best of our knowledge, the first decade-long benchmark (2015–2025, 335,820 interventions) coupling French weather vigilance bulletins with fire service activity, suitable for evaluating retrieval-augmented text-to-count regression.

- We disentangle the contributions of embedding quality, retrieval-based reasoning (a k-NN mean baseline) and LLM reasoning on top of retrieval, showing that a large fraction of the observed gain comes from the retrieval step itself rather than from the LLM.
- We compare two production-grade embedding models (Mistral-Embed and OpenAI `text-embedding-3-small`) on domain-specific French textual data, and report which embedding works best for each downstream model family.
- We release the prompts and pipeline description in sufficient detail to enable replication on similar operational datasets.

Research questions. Concretely, we ask: (RQ1) how much predictive signal is carried by the semantic similarity of bulletins alone, as captured by a k-NN retrieval baseline with no learned parameters? (RQ2) what is the marginal contribution of LLM reasoning [17] on top of the same retrieved context? (RQ3) does the answer depend on the embedding model, the meteorological risk category or the season?

The remainder of the paper is organised as follows. Section 2 describes the data, the semantic vectorization pipeline and the three families of predictors. Section 3 presents the experimental results. Section 4 discusses them, together with their limitations (Section 4.1). Section 5 concludes.

2 Material and Methods

2.1 Data Preparation

Data Origin and Retrieval The dataset contains 335,820 intervention records from January 1, 2015 to May 31, 2025, obtained from the Fire and Rescue Services of Doubs, France. Each record holds timestamps for the start and end of the intervention, a geographical location, a duration, and an intervention type. Meteorological variables were derived from Météo France's weather vigilance bulletins for the Doubs region. These bulletins combine a colour-coded alert system with descriptive textual content, and are summarised in Table 1 and illustrated in Figure 1 [18].

Table 1: Classification of Meteorological Hazards by Alert Severity

Atmospheric risk	**Green**	**Yellow**	**Orange**	**Red**
Wind	76792	1321	2722	420
Rainfall	75769	1820	3244	189
Lightning storms	73355	3995	2865	889
Frozen precipitation	82754	12	465	134
Extreme heat	82146	358	987	75
Cold wave	81322	21	117	11

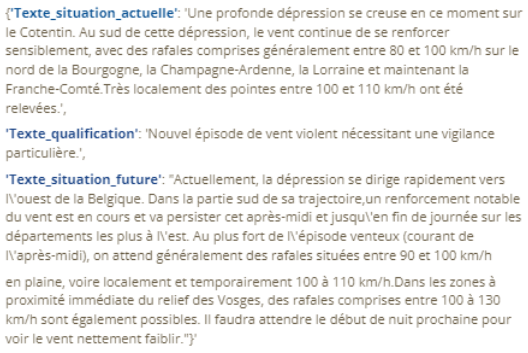


Fig. 1: Illustrative weather vigilance bulletin (in French), with the three structured fields used as input to the embedding model: current situation, event qualification, and forecast.

Data Preprocessing and Transformation The weather vigilance bulletins are grouped by event category into separate directories and stored in JSON files sharing a common schema: risk type, alert level, start and end dates, current situation description, event characterization, and forecast. Events are then aggregated by period (based on their start and end dates) and sorted chronologically.

For each event period, the target variable is the *mean daily number of fire service interventions* observed between the start and end dates of the event. Intervention records are filtered to match the event bounds; the mean is computed across days of the event and rounded to the nearest integer, for reporting consistency with the aggregated bulletin granularity. When no intervention is recorded in the period, the target is zero.[1] The end product is a table in which each weather event is associated with a single integer-valued target.

Exploratory Data Analysis Fire and Rescue Services of Doubs handle about 32,000 interventions per year on average (roughly four incidents per hour), with a steady annual increase over the decade that can plausibly be attributed to demographic growth, population ageing, climate change and the closure of smaller local hospitals. To visualise the semantic structure of the weather bulletins, we vectorized them with Mistral-Embed [19], a model trained for French-language text, and projected the resulting high-dimensional vectors in two dimensions using t-SNE [20]; the result is shown in Figure 2.

The 2D projection is meteorologically sensible. Frozen precipitation and cold wave events sit close to each other in the lower part of the plot, consistent with their shared winter regime of low, often sub-zero temperatures. Wind, lightning

[1] Using a mean rather than a total reflects the heterogeneous duration of weather events, which ranges from a few hours to several days; we discuss this design choice and its implications in Section 4.1.

storms and rainfall are grouped together in the upper region, with frequent overlaps that reflect their physical coupling: convective storms typically combine heavy precipitation and strong winds. The extreme-heat category occasionally overlaps with lightning storms, which is also consistent with the atmospheric physics (heat episodes frequently end with convective events) and serves as a sanity check on the embedding space.

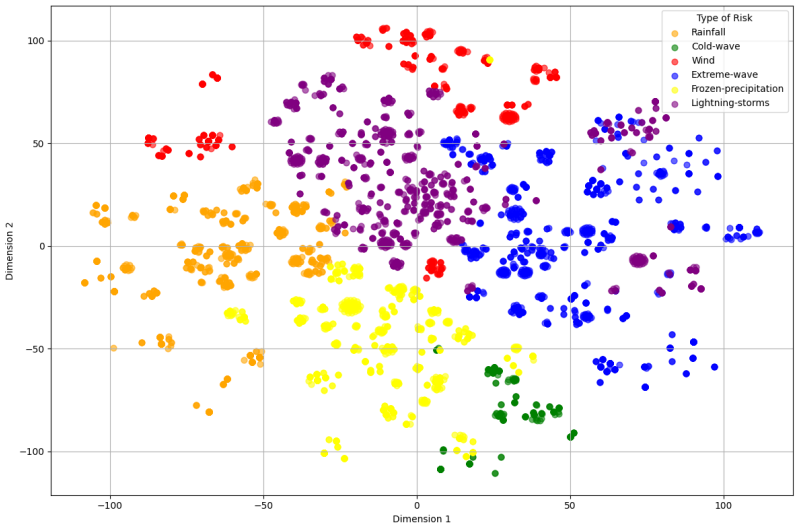


Fig. 2: Two-dimensional t-SNE projection of Mistral-Embed vectors for the weather vigilance bulletins, coloured by risk category. Category labels follow Table 1; "Extreme-heat" replaces the older "Extreme-wave" label used by the t-SNE script in an earlier version of the figure.

2.2 Semantic Vectorization and Predictive Modeling

Semantic Vectorization The vectorization pipeline maps each textual vigilance bulletin to a dense vector using two embedding models, Mistral-Embed and OpenAI `text-embedding-3-small`. The resulting vectors are stored in Milvus, a vector database supporting approximate nearest-neighbour search. The same index is then queried in two ways: by the k-NN retrieval baseline, which averages the intervention counts of the top-k retrieved bulletins, and by the LLM predictors, which receive the retrieved bulletins as context in a Retrieval-Augmented Generation (RAG) prompt. The pipeline is orchestrated with LangChain, and follows the same overall template as recent work of our group that applies vector stores and information retrieval to domain-specific textual corpora in unrelated fields [8].

1. Text embedding. To map each bulletin to a numerical representation, we use two production-grade embedding models.
   Mistral-Embed dimension = 1024, optimised for French-language inputs.
   OpenAI text-embedding-3-small dimension = 1536, known for its robustness and generalisation on multilingual textual data.
   Let $T = \{t_1, t_2, \ldots, t_n\}$ denote the set of vigilance bulletins, where each t_i represents a textual segment. The embedding function $f : T \to \mathbb{R}^d$ maps each bulletin t_i to a high-dimensional vector $v_i = f(t_i)$, where the dimensionality d is either 1024 or 1536, contingent on the specific embedding model employed. Consequently, the resulting embedding matrices are formulated as: $\mathbf{V}^{\text{Mistral}} \in \mathbb{R}^{n \times 1024}$ and $\mathbf{V}^{\text{OpenAI}} \in \mathbb{R}^{n \times 1536}$.
2. Semantic Storage with Milvus: The high-dimensional vectors were stored in Milvus, a high-performance vector database designed for similarity search at scale. This step enables efficient approximate nearest neighbor (ANN) queries during the RAG. The structure of each entry in the Milvus collection included.
   - The embedded vector representation of the bulletin.
   - The original bulletin text.
   - The associated number of emergency interventions.

   Milvus was used for experiments involving LLMs and the k-NN retrieval baseline; traditional predictive models were trained separately using only the embeddings as input features, without access to retrieved neighbours. This asymmetry in the information available to each family of models must be kept in mind when interpreting the comparative results reported in Section 3.
3. RAG for Inference: To support prompt generation for LLMs and the k-NN retrieval baseline, we implemented a RAG procedure. For a given query bulletin, semantically similar weather bulletins (excluding the query itself) were retrieved from the Milvus index and used both to construct the LLM context window and to feed a simple k-NN averaging procedure over historical targets. Let $q \in \mathbb{R}^d$ be the embedding of the input query, and let $\{v_1, v_2, \ldots, v_k\} \subset \mathbb{R}^d$ be the stored vectors in Milvus. The retrieval step identifies the top-k vectors most similar to the query, using inner product similarity (Equation (1)).

$$\text{similarity}(\mathbf{q}, \mathbf{v}_i) = \mathbf{q} \cdot \mathbf{v}_i = \sum_{j=1}^{d} q_j v_{ij} \tag{1}$$

   $\mathbf{q}$ is the query vector.
   $\mathbf{v}_i$ is the i-th vector in your database.
   d is the dimension of the vectors.
   q_j is the j-th component of the vector $\mathbf{q}$.
   v_{ij} is the j-th component of the i-th vector $\mathbf{v}_i$.

   This dual approach allows us to disentangle the contribution of retrieval (captured by the k-NN baseline) from the additional contribution of LLM reasoning on top of the same retrieved context.

4. Orchestration with LangChain: retrieval, prompt assembly and LLM inference are glued together using the LangChain framework, which makes it straightforward to swap embedding models, prompt templates or LLMs without rewriting the surrounding pipeline.

Predictive Modeling To generate predictions and enable controlled comparisons, we employed three families of predictors: modern LLMs, classical machine learning models on raw embeddings, and a k-NN retrieval baseline.

Two LLMs are used through their RAG-augmented inference mode: GPT [21] and Mistral [22]. GPT serves as a generalist baseline, while Mistral provides a more compact alternative better tuned to French-language input. No fine-tuning is performed on the intervention dataset; the LLMs are used off the shelf.

The embedding-only path relies on three classical regressors: Random Forest (RF) [23], Linear Regression (LR) [24], and Support Vector Regression (SVR) [25]. RF captures non-linear ensemble patterns, LR gives a linear baseline, and SVR is a margin-based regressor in high-dimensional embedding spaces.

The k-NN retrieval baseline predicts the mean intervention count of the top-k bulletins returned by the Milvus query. It shares the retrieved context with the LLM path but discards the LLM reasoning step, which makes it the natural ablation for isolating the contribution of LLM reasoning over retrieval alone. We describe it as a k-NN retrieval baseline, rather than a "human-aligned" one, to reflect what the method actually computes.

1. GPT and Mistral (LLMs). Both are used through their standard autoregressive inference, $P(y \mid x) = \prod_t P(y_t \mid y_{<t}, x; \theta)$, with no fine-tuning on the intervention dataset; the numerical prediction is parsed from the model's textual answer to a templated prompt (Figure 8).
2. Random Forest (RF), Equation (2).

$$\hat{y}_i = \frac{1}{K} \sum_{k=1}^{K} f_k(\mathbf{x}_i) \tag{2}$$

$\hat{y}_i$ denotes the predicted value for the i-th observation.
K is the total number of decision trees in the random forest ensemble.
f_k represents the output function of the k-th tree.
$\mathbf{x}_i$ is the feature vector corresponding to observation i.

3. Linear Regression (LR), Equation (3).

$$\hat{y}_i = \mathbf{w}^\top \mathbf{x}_i + b \tag{3}$$

$\hat{y}_i$ denotes the predicted value for observation i.
$\mathbf{x}_i$ is the feature vector for observation i.
$\mathbf{w}$ is the vector of learned coefficients (weights).
b is the bias term (intercept) of the model.

4. Support Vector Regression (SVR), Equation (4).

$$\hat{y}_i = \sum_{j=1}^{n} (\alpha_j - \alpha_j^*) K(\mathbf{x}_i, \mathbf{x}_j) + b \tag{4}$$

$\hat{y}_i$ denotes the predicted value for the i-th input.
α_j and α_j^* are the Lagrange multipliers associated with the support vectors.
$K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function evaluating the similarity between inputs $\mathbf{x}_i$ and $\mathbf{x}_j$.
b is the bias term.
n is the total number of support vectors.

5. k-NN Retrieval Baseline, Equation (5).

$$\hat{y}_{k\mathrm{NN}} = \frac{1}{k} \sum_{i=1}^{k} y_i \tag{5}$$

$\hat{y}_{k\mathrm{NN}}$ denotes the estimated number of interventions obtained by averaging the targets of the top-k retrieved bulletins.
y_i is the intervention count associated with the i-th retrieved bulletin.
k is the number of neighbours used in the averaging process.

2.3 Predictive pipeline

Once bulletins have been vectorized and indexed as described above, the same embeddings feed the three predictive paths compared in this paper (Figure 3).

In the first path, embeddings are fed directly as features to RF, LR and SVR, with the intervention count as regression target; no retrieval is involved. In the second path, the k-NN retrieval baseline queries Milvus with the embedding of the new bulletin (inner-product similarity), retrieves the top-k most similar training bulletins, and predicts the mean of their intervention counts. In the third path, the same retrieved bulletins are injected as context into a prompt given to a generative LLM (GPT or Mistral), which is asked to output a predicted intervention count together with a short textual justification. The second and third paths share the same retrieved context, so that comparing them isolates the marginal contribution of LLM reasoning over retrieval alone.

To evaluate the models, the dataset was partitioned into 80% for training and 20% for testing. The training bulletins were indexed in Milvus; at inference time, queries retrieve neighbours exclusively from this training index and any retrieved neighbour identical to the query is filtered out, so that the k-NN baseline and the LLM+RAG predictor are evaluated on strictly held-out queries.[2] The performance of each model was assessed using four standard regression metrics, which are defined as follows:

[2] Temporal aspects of the split (random vs. chronological) and their implications for extrapolation to future dates are discussed in Section 4.1.

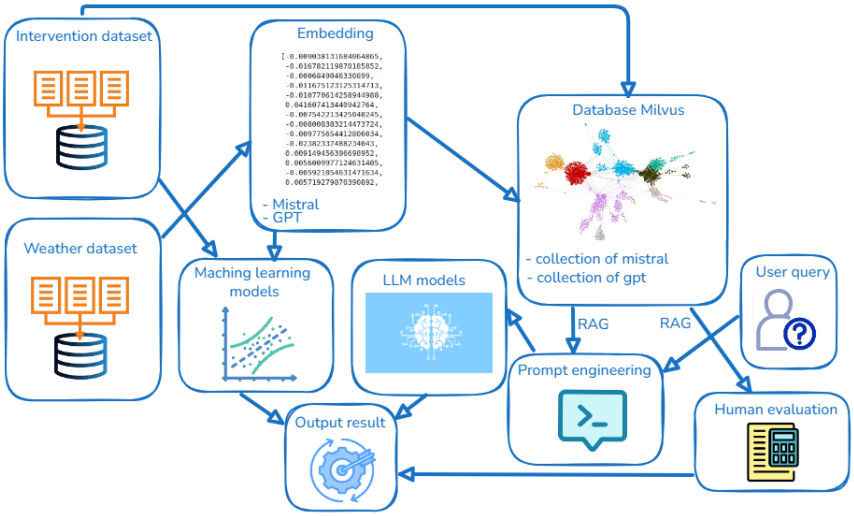


Fig. 3: Proposed architecture for meteorology-based intervention prediction. During training, bulletins and their intervention targets are embedded and stored in Milvus; during inference, a query bulletin is embedded and used either directly by an embedding-only ML model (bottom-left path) or to retrieve top-k similar past bulletins whose targets are averaged (k-NN baseline) or injected as context into an LLM prompt (RAG path). "Machine learning models" in the figure refers to the classical ML models (RF, LR, SVR).

1. Root Mean Square Error (RMSE), Equation (6).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{6}$$

2. Mean Absolute Error (MAE), Equation (7).

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{7}$$

3. Root Mean Squared Logarithmic Error (RMSLE), Equation (8).

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log(1+\hat{y}_i) - \log(1+y_i)\right)^2} \tag{8}$$

4. Coefficient of Determination (R^2), Equation (9).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{9}$$

These metrics are complementary: RMSE and RMSLE penalise large errors more heavily than small ones, MAE gives a linear average of the error magnitude, and R^2 is the fraction of target variance explained by the model. By sharing retrieved context between the k-NN baseline and the LLM path, and by reporting the same metrics on the three paths, we can attribute any gain to the retrieval step or to the LLM step rather than conflating the two.

3 Results

Building the dataset was a non-trivial step in itself: ten years of intervention records had to be paired with the corresponding vigilance bulletins, and the bulletins vectorized with two embedding models (Mistral-Embed and OpenAI `text-embedding-3-small`) to enable a side-by-side comparison. Once the models had been trained and evaluated, their predictive performance was measured on the held-out test set; the aggregate results appear in Table 2, for both embedding spaces.

Table 2: Prediction results using OpenAI and Mistral embeddings. The k-`NN (retrieval)` row is the retrieval baseline averaging the intervention counts of the top-k semantically most similar training bulletins, without any LLM reasoning step. It therefore quantifies how much of the final LLM+RAG performance is already achievable by retrieval alone.

Model	Input	RMSE	MAE	RMSLE	R^2
`Mistral`	OpenAI-Vec	3.22	2.92	0.40	0.71
	Mistral-Vec	2.92	2.61	0.36	0.75
`GPT`	OpenAI-Vec	2.45	2.28	0.34	0.79
	Mistral-Vec	**2.30**	**2.10**	**0.21**	**0.84**
`Random forest`	OpenAI-Vec	4.65	3.85	0.81	0.60
	Mistral-Vec	4.71	3.95	0.85	0.58
`Linear regression`	OpenAI-Vec	5.25	4.25	0.95	0.50
	Mistral-Vec	5.32	4.35	1.03	0.44
`Support vector regression`	OpenAI-Vec	7.07	6.12	1.23	0.15
	Mistral-Vec	8.15	7.22	1.50	0.06
k-`NN (retrieval)`	OpenAI-Vec	3.55	2.95	0.48	0.69
	Mistral-Vec	3.42	2.85	0.44	0.72

Three patterns emerge from Table 2. First, embedding-only ML models perform poorly without access to retrieved neighbours: SVR stays at or below $R^2 = 0.15$, while RF and LR reach at best $R^2 \in [0.44, 0.60]$. Second, the k-NN retrieval baseline, which has no learned parameters, already reaches $R^2 = 0.72$ (Mistral-Vec), outperforming the best embedding-only model (RF, $R^2 = 0.60$) by $\Delta R^2 = 0.12$. Third, on top of the same retrieved context, GPT adds a further $\Delta R^2 \approx 0.12$ over k-NN (from 0.72 to 0.84) and Mistral adds $\Delta R^2 \approx 0.03$ (from 0.72 to 0.75). Figures 4 and 5 break these results down by meteorological

risk category and by year, respectively; the ranking between model families is stable across years and across five of the six risk categories, the exception being extreme heat, for which Mistral ties or slightly exceeds GPT.

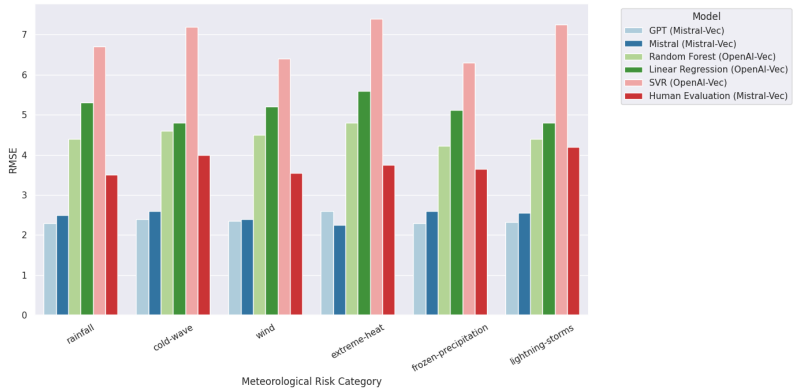


Fig. 4: Average RMSE by risk category and model. For each model, the embedding choice (Mistral-Vec or OpenAI-Vec) yielding the lowest overall RMSE on the test set is shown: Mistral-Vec for LLMs and the k-NN baseline, OpenAI-Vec for the embedding-only ML models. The ranking between model families is qualitatively preserved if a single embedding is fixed for all models.

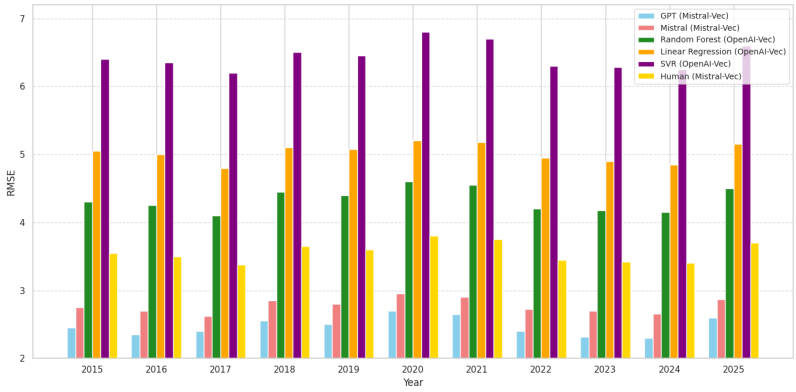


Fig. 5: Yearly RMSE scores per model (best-performing embedding per model, same convention as Figure 4).

Figure 6 shows the average prediction error by month. GPT with Mistral embeddings has the lowest error on most months, with a slight advantage for

Mistral during summer (July and August), a pattern we return to in the discussion.

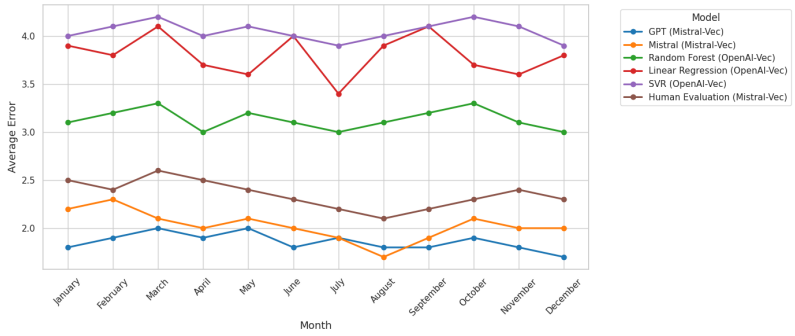
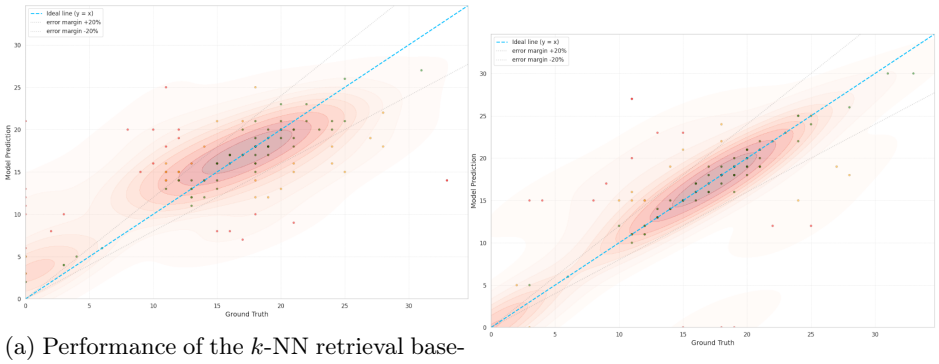


Fig. 6: Monthly average prediction error by model (best-performing embedding per model, same convention as Figure 4).

Finally, Figures 7-(a) and 7-(b) compare the best LLM configuration (GPT with Mistral embeddings) against the k-NN retrieval baseline, both sharing the same retrieved context. Most predictions fall within a $\pm 20\%$ error margin of the ideal $y = x$ line, with GPT producing a tighter prediction cloud and fewer systematic biases at high intervention counts. Statistical significance of the differences between models is not assessed here and is left for future work; the comparison should therefore be read as descriptive rather than inferential.



(a) Performance of the k-NN retrieval baseline

(b) Performance of GPT-based LLM

Fig. 7: Predicted vs. ground-truth intervention counts on the test set: the k-NN retrieval baseline (a) and GPT+RAG (b), both using Mistral embeddings and the same retrieved context. Dotted lines indicate $\pm 20\%$ error margins around the ideal $y = x$ line (dashed).

4 Discussion

This study compares three families of predictors on a decade of fire service interventions (2015–2025) coupled with French weather vigilance bulletins: embedding-only ML, a k-NN retrieval baseline, and LLMs conditioned via RAG on the same retrieved context, each evaluated on two embedding spaces (Mistral and OpenAI). We organise the discussion around three questions: where does the predictive signal come from, how do the models behave across risk categories and seasons, and what does the LLM add beyond numerical accuracy.

Where does the signal come from? The ranking in Table 2 can be read as a three-step ablation. Embedding-only ML models (RF, LR, SVR) reach $R^2 \in [0.06, 0.60]$, confirming that raw embeddings alone carry only limited signal for this task. Simply averaging the targets of the top-k semantically similar past bulletins (k-NN baseline, no learned parameters) already reaches $R^2 = 0.72$, which suggests that the dominant source of predictive signal is the similarity structure of the bulletin space combined with the stability of SDIS activity under recurring meteorological regimes. The additional gain from LLM reasoning on top of retrieval is $\Delta R^2 \approx 0.12$ for GPT and $\Delta R^2 \approx 0.03$ for Mistral: non-negligible, but much smaller than the retrieval gain itself. We emphasise this decomposition because a naïve reading of "LLMs outperform classical ML" could over-credit LLM reasoning.

Choice of embedding. Both the LLMs and the k-NN baseline benefit from Mistral-Embed, likely because of its stronger fit to French-language inputs; conversely, the embedding-only ML models tend to perform better with OpenAI embeddings. Reporting, in Figures 4 to 6, each model with its best-performing embedding is a form of cherry-picking and is acknowledged as such; the same ranking holds qualitatively if a single embedding is fixed for all models (see Table 2).

Category and seasonal behaviour. A category-level analysis (Figure 4) shows that GPT+RAG leads on five out of six risk categories, with Mistral+RAG overtaking GPT on extreme heat. A month-level analysis (Figure 6) shows that GPT+RAG has the lowest error on most months, while Mistral+RAG is best during summer. These observations are consistent with the hypothesis that Mistral's French-centric training provides a small advantage on the more domain-specific descriptions (e.g., summer heat episodes). Whether this observation generalises beyond the present dataset, or is partly noise driven by a single test split, cannot be decided without repeated splits and confidence intervals, which we do not provide here.

Textual justifications. Figure 7 shows that, on the same retrieved context, GPT produces a tighter prediction cloud than the k-NN baseline. Beyond the point prediction, the LLM produces a short structured justification (Figure 8) listing risk type, level and predicted impacts. Such textual output could in principle support operator interpretation and auditability, but we do not evaluate its

utility with end-users here; the claim is therefore limited to "the LLM emits structured explanations" and must not be read as "the LLM helps operators".

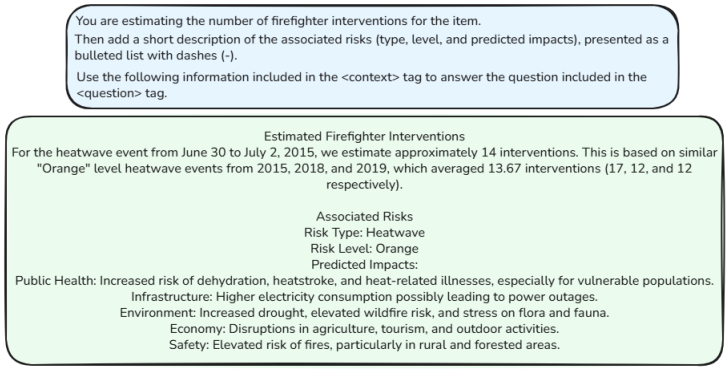


Fig. 8: Example of a prompt (top) and of the structured LLM response (bottom), for a June–July 2015 orange-level heatwave. The LLM is instructed to estimate a number of interventions and to produce a short bulleted description of the associated risks. Accuracy of the estimate is evaluated in Section 3; the factual content of the explanation is not evaluated in this paper.

4.1 Limitations and threats to validity

Several limitations should be kept in mind when interpreting the results reported above.

Split strategy. The 80/20 train/test partition is not explicitly time-aware. If the split is uniformly random across the 2015–2025 period, the test set may contain bulletins that are temporally interleaved with the training set, which on this task is favourable to retrieval-based approaches (a similar event in the training set is often very close in time to its test counterpart). A strictly chronological split (e.g., training on 2015–2023, testing on 2024–2025) would be more faithful to the prospective use case of forecasting future interventions; we do not report such a split here and cannot therefore claim that the observed R^2 values transfer to a truly prospective setting.

Retrieval and leakage control. The k-NN and LLM+RAG predictors share the same retrieved context. Any leakage between the Milvus index and the query set would inflate both baselines symmetrically, but would also inflate them relative to the embedding-only ML models. In our protocol, only training bulletins are indexed and exact query duplicates are filtered from the retrieval; however, near-duplicates due to multi-day vigilance episodes may still be retrieved, and this may contribute to the strong k-NN baseline. A stricter temporal buffer around each query would offer a more conservative evaluation.

Statistical uncertainty. We report point estimates of RMSE, MAE, RMSLE and R^2 on a single test split. Neither confidence intervals (e.g., via bootstrap) nor pairwise significance tests (e.g., Wilcoxon signed-rank on absolute errors, or Diebold-Mariano on forecasts) are provided. The LLMs used here are also stochastic, and no run-to-run variance is reported. Quantitative differences close to the magnitude of a plausible confidence interval should therefore not be over-interpreted.

Target variable construction. The target is the rounded *mean daily number of interventions* during the event, with zero assigned when no intervention occurred. This design choice mixes events of very different durations and obscures the distinction between "no intervention observed" and "no intervention attributable to this event", and it may deserve re-examination in future work.

Cost, latency and deployment constraints. Inference cost and latency differ by orders of magnitude between RF/LR/SVR (sub-millisecond, local), the k-NN baseline (index lookup, local), and the LLMs (seconds per call, external proprietary API). For an actual SDIS deployment, dependency on a proprietary API raises questions of cost, latency, data sovereignty, GDPR compliance and long-term availability of specific model versions, none of which are addressed in this study.

Baselines not explored. Several stronger or more standard baselines are not evaluated: gradient boosting (e.g., XGBoost, LightGBM) on embeddings, count-appropriate regressors (Poisson or zero-inflated Poisson), calendar baselines (historical mean by month and event category) and hybrid models that feed k-NN retrieved targets as additional features to RF or LR. Their inclusion would strengthen future comparisons.

Interpretability. The textual justifications produced by the LLMs (Figure 8) are evaluated here only through a qualitative example. No user study, no assessment of factual correctness of the justifications, and no analysis of failure cases are provided; claims about operational interpretability should therefore be treated as hypotheses to validate, not as established results.

5 Conclusion

This study compares three families of predictors for fire service activity from French weather vigilance bulletins over a ten-year period (2015–2025): embedding-only ML models, a k-NN retrieval baseline, and LLMs conditioned via RAG on the same retrieved context. Using a decade-long dataset of 335,820 interventions from the Doubs Fire and Rescue Service (SDIS 25), we observe the following.

First, most of the gap between embedding-only ML and LLM+RAG is captured by the k-NN retrieval step itself, which reaches $R^2 = 0.72$ with no learned parameters. Second, LLM reasoning on top of the same retrieved context adds

a further $\Delta R^2 \approx 0.12$ for GPT and ≈ 0.03 for Mistral, producing the best overall configuration (GPT with Mistral embeddings: RMSE = 2.30, MAE = 2.10, $R^2 = 0.84$). Third, LLMs emit structured textual justifications for their predictions, the operational usefulness of which remains to be validated with end-users.

These observations support a measured conclusion: on this benchmark, retrieval-augmented generation is a promising avenue for predicting fire service activity from textual weather bulletins, *provided* that (i) the evaluation protocol controls for retrieval leakage and uses a chronologically faithful split, (ii) statistical uncertainty is quantified, and (iii) operational trade-offs (cost, latency, dependency on proprietary APIs) are explicitly taken into account. We detail these caveats in Section 4.1.

Future work will (i) re-run the benchmark under a strictly chronological split with bootstrap confidence intervals and pairwise significance tests; (ii) extend the predictor set to gradient boosting and count-appropriate regressors as additional baselines; (iii) integrate complementary contextual features (media reports, road accident data) to move toward a more holistic model of emergency activity; and (iv) study the interpretability and failure modes of the LLM justifications with SDIS operators.

This work has been achieved in the frame of the EIPHI Graduate school (contract "ANR-17-EURE-0002").

References

1. Raiaan, et al. *A review on large Language Models: Architectures, applications, taxonomies, open issues and challenges*, *IEEE Access*, 2024, Publisher: IEEE.
2. Xu, et al. *Mental-llm: Leveraging large language models for mental health prediction via online text data*, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 8, n° 1, pp. 1–32, 2024, Publisher: ACM New York, NY, USA.
3. Geng, Binzong and Huan, Zhaoxin and Zhang, Xiaolu and He, Yong and Zhang, Liang and Yuan, Fajie and Zhou, Jun and Mo, Linjian, *Breaking the length barrier: Llm-enhanced CTR prediction in long textual user behaviors*, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2311–2315, 2024, Publisher: ACM New York, NY, USA.
4. Schoenegger, Philipp and Park, Peter S and Karger, Ezra and Trott, Sean and Tetlock, Philip E, *Ai-augmented predictions: Llm assistants improve human forecasting accuracy*, *ACM Transactions on Interactive Intelligent Systems*, vol. 15 ,n° 1, pp. 1–25, 2025, Publisher: ACM New York, NY.
5. Peng, Mingxing and Guo, Xusen and Chen, Xianda and Chen, Kehua and Zhu, Meixin and Chen, Long and Wang, Fei-Yue, *LC-LLM: Explainable lane-change intention and trajectory predictions with large language models*, *Communications in Transportation Research*, vol. 5, p. 100170, 2025, Publisher: Elsevier.
6. Cerna, Selene and Guyeux, Christophe and Laiymani, David, *The usefulness of NLP techniques for predicting peaks in firefighter interventions due to rare events.* In: *Neural Computing and Applications*, vol. 34, pp. 10117–10132, 2022, Publisher: Springer.

7. Mallouhy, Roxane and Sirri, Naoufal and Nahvi, Irum and Guyeux, Christophe, *AI's Current Impact and Future Potential in Emergency Services: A Comprehensive Review and Analysis*. In: *International Journal of Intelligent Systems and Applications*, vol. 14, no. 6, pp. 1–19, 2024, Publisher: MECS Press. DOI: 10.5815/ijisa.2024.06.01.
8. Guyeux, Christophe, *Integrating Vector Stores and Information Retrieval for Analysis Literature Data on Mycobacterium tuberculosis: experience feedback*. In: *5th IEEE Middle East & North Africa COMMunications Conference (MENACOMM'25)*, Byblos, Lebanon, February 2025, IEEE.
9. Sirri, Guyeux, *Traditional vs. AI-generated meteorological risks for emergency predictions*. In: *Frontiers in Artificial Intelligence*, vol. 8, p. 1545851, 2025, Publisher: Frontiers Media SA.
10. Sirri, Guyeux, *Firefighter Intervention Predictive Modeling: Reliability Assessment*. In: *International Conference on Circuit, Systems and Communication*, IEEE , 2024.
11. Sirri, Guyeux, *Reliability of Firefighter Prediction Models Across Time and Peaks*. In: *International Conference on Advanced Information Networking and Applications*, Springer, 2025.
12. Sirri, Guyeux, *Air Quality Impact on Firefighter Interventions: Factors Analysis*. In: *The 7th International Conference on Big Data and Internet of Things*, Springer, 2024.
13. Sirri, Guyeux, *Solar activity Impact on Firefighter Interventions: Factors Analysis*. In: *The 5th International Conference on Deep Learning and Applications*, Springer, 2024.
14. Sirri, Guyeux, *River Levels Affecting Firefighter Interventions: Factor Analysis*. In: *Proceedings of the 28th International Database Engineered Applications Symposium (IDEAS 2024)*, pp. 394–410, ACM, 2024.
15. Sirri, Guyeux, *Assessing the Influence of Epidemics on Firefighter Services: A Factor Analysis Approach*. In: *International Conference on Multi-disciplinary Trends in Artificial Intelligence*, Springer, 2024.
16. Sirri, Guyeux, *Factor Analysis of Weather Conditions Impact on Firefighter Interventions*. In: *The 11th International Conference on Future Data and Security Engineering*, Springer, 2024.
17. Gao, et al. *Retrieval-augmented generation for large language models: A survey*. arXiv preprint arXiv:2312.10997, 2023.
18. Vigilance-France. https://vigilance.meteofrance.fr/fr. Accessed on 20 July 2025.
19. Mistral Embeddings. https://docs.mistral.ai/capabilities/embeddings/overview/. Accessed on 20 July 2025.
20. Maaten, Laurens van der and Hinton, Geoffrey, *Visualizing data using t-SNE. Journal of machine learning research*, vol. 9, pp. 2579–2605, 2008.
21. Brown, et al. *Language models are few-shot learners*. In: *Advances in neural information processing systems*, volume 33, pp. 1877–1901, 2020.
22. Jiang, et al. *Mixtral of experts*. arXiv preprint arXiv:2401.04088, 2024.
23. Breiman, Leo. *Random forests*. In: *Machine learning*, volume 45, pp. 5–32, 2001, Publisher:Springer.
24. Su, Xiaogang and Yan, Xin and Tsai, Chih-Ling *Linear regression*. In: *Wiley Interdisciplinary Reviews: Computational Statistics*, volume 4, pp. 275–294, 2012, Publisher:Wiley Online Library.
25. Cortes, Corinna and Vapnik, Vladimir. *Support-Vector Networks*. In: *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. DOI: 10.1007/BF00994018.