異常値を含むデータに対する文脈情報統合の有効性に関する研究

― 株価予測を題材とした検証 ―

内田陽太・浦崎華瑠

有明工業高等専門学校 情報システムコース
Gauthier Lovic 研究室

2026 年 1 月 27 日

An Empirical Study on the Effectiveness of Contextual Information Integration for Data with Outliers

― Evidence from Stock Price Prediction ―

Hinata Uchida・Haru Urasaki

Course of Information Systems, Department of Creative Engineering, National Institute of Technology (KOSEN), Ariake College

January 27, 2026

概要

時系列予測は金融・需要予測・設備保全など多様な分野で重要である一方，数値特徴量のみの学習では，外部要因に起因する急変動（文脈依存の異常局面）を十分に説明しにくい．本研究は株価日足を題材に，ニュース（文脈情報）を特徴量として統合することが汎化性能，とりわけ急変動日での性能に与える影響を検証する．

まず数値特徴量のみで 4 モデル（Logistic Regression / LightGBM / LSTM / Transformer）を比較し，次に GDELT 由来のニュース特徴量を追加して同一条件で再評価する．ニュースは全期間を GKG（日次ファイル）で取得し，取引日単位に記事数・ソース多様性・トーン統計および V2Tone 成分等を日次集計し，短期 rolling と急増度（surprise）を付与する．公開時刻の不整合による将来情報混入を避けるため，引け後・非取引日のニュースは翌取引日に繰り上げ，GKG 日次で時刻情報が粗い場合も安全側に翌取引日扱いとする．

評価は翌営業日方向性の三値分類（上昇・横ばい・下落）とし，時系列リークを抑える Purged Walk-Forward Embargo（PWFE）分割を用いる．さらに，将来リターンの絶対値 $|r_H|$ 上位 10% を急変動日集合（subset）として定義し，通常日（main）と併せて macro-F1，balanced accuracy，混同行列を報告する．実験の結果，main では Logistic Regression と LightGBM で macro-F1 が約 +0.02 改善した一方，LSTM と Transformer では改善が見られなかった．subset では LightGBM が相対的に改善した一方，他モデルでは低下または改善が限定的であり，日次集計型ニュース特徴量は木モデルで活用されやすい可能性が示唆された．

目次

1　序論

本章では，研究動機（背景）→ 課題設定 → 目的 → 位置づけ → 論文構成の順に道筋を示す．以降の章では，この課題設定に対応する形で，手法（データ・特徴量・分割・学習）と実験結果を順に提示する．

1.1　背景

時系列データの将来予測は，金融取引，需給予測，設備保全，異常検知など多様な分野で基盤技術として位置づけられる．特に株価は公開データが豊富であり，日足 OHLCV（Open/High/Low/ Close/Volume）に加えて指数・為替・ボラティリティ指数などの市場要因も同時に観測できるため，時系列予測の研究題材として重要である．一方で株価系列は，非定常性，急激なレジーム変化，外生ショックの影響を受けやすく，単一の統計的仮定の下で安定した予測則を得ることが難しい．

株価の短期変動は，自己相関やボラティリティの時間変動（いわゆるクラスタリング），市場全体のリスク要因，需給・流動性など複数要因の重ね合わせとして観測される．実務・研究の双方で，これらを代理する特徴量として移動平均乖離，モメンタム，ボラティリティ指標（ATR，BB 幅等），出来高指標などが広く用いられてきた．しかし，決算，政策金利，地政学リスク，災害，企業不祥事といったイベントが引き金となる急変動は，価格系列だけからは原因を特定しづらく，数値特徴量のみでの学習では「外部要因に起因する変動」を十分に説明できない場合がある．

この点に対し，ニュースや SNS 等の文脈情報を予測に統合する試みがなされている．ただし，文脈情報は非構造であり（テキスト・メタデータの混在），取得・整形・欠損処理に加えて，公開時刻と市場（営業日・引け時刻・タイムゾーン）の整合が不可欠である．とりわけ公開時刻の扱いを誤ると，検証対象日の後に公開されたニュースが当日特徴量へ混入し，将来情報を使ってしまう look-ahead（時系列リーク）を招く．したがって，ニュース統合の有効性を議論するには，「情報を安全に当該取引日へアラインする」前処理設計と，通常局面だけでなく急変動局面に焦点を当てた評価設計が重要となる．

近年，GDELT に代表される大規模ニュースデータが整備され，記事数・ソース多様性・トーン統計など，日次レベルで扱える構造化信号も利用可能になった．本研究はこれらを活用し，株価日足の三値分類という設定の下で，数値特徴量のみの場合とニュース特徴量を統合した場合を同一条件で比較し，特に急変動日での汎化性能がどのように変化するかを検証する．

以上の背景から，本研究で本質的に難しい点は「ニュースを使うこと」そのものではなく，**(i) ニュースを安全に取引日へ対応付ける（リークを避ける）こと**と，**(ii) 急変動局面で本当に効いたかを測れる評価枠組みを**用意することにある．そこで次項では，この 2 点を中心に課題を明確化する．

1.2　課題

株価の短期変動は，自己相関やボラティリティの時間変動（クラスタリング），市場全体要因（指数・為替・VIX 等）に加え，決算・政策・災害・不祥事といった外部イベントに強く影響される．そのためニュースに代表される文脈情報を特徴量として統合できれば，とくに急変動局面での性能改善が期待される．しかし，文脈情報の統合には以下の課題がある．

■(1) ニュースの時刻整合と時系列リーク（look-ahead）　ニュースは公開時刻が取引所の営業日・引け時刻と一致せず，週末・祝日もまたぐ．さらに GDELT GKG では時刻情報が粗く，UTC 00:00 に丸められたレコードが多い．このとき，ニュースを「発生日の特徴量」として安易に当日へ結合すると，実際には取引終了後（あるいは翌日の取引開始前）に公開された情報が当日の入力へ混入し，将来情報を用いた過大評価（look-ahead bias）を招きうる．従って，ニュースを取引日へ写像する規則（引け後・非取引日の繰り上げ，タイムゾーン処理）を明示し，安全にアラインする必要がある．

■(2) 評価設計：急変動日での有効性をどう測るか　平均的な性能（main）だけでは，急変動局面での有効性を見落とす可能性がある．一方，急変動日を例えば将来リターンの絶対値上位 $p\%$ で定義すると，クラス分布が偏り（中立クラスが欠損する等），macro-F1 や balanced accuracy の解釈が難しくなる場合がある．したがって，通常日（main）と急変動日（subset）を分離して評価し，指標の限界を踏まえつつ混同行列等も併用して誤分類の傾向を確認する枠組みが必要である．

以上より，ニュース統合を検証するには，(1) 時刻整合を安全側に設計してリークを抑え，(2) 通常局面と急変動局面を分離した評価を行う，という 2 点を同時に満たすことが課題となる．

次項では，上記の課題 (1)(2) を**どのような実験設計で解決し，何をもって「効果があった」と結論づけるのか**を，研究目的として具体化する．

1.3　目的

本研究の目的は，株価日足の三値分類（上昇・横ばい・下落）を題材として，ニュースに代表される文脈情報を数値特徴量へ統合することが予測性能に与える影響を，時系列リークを抑えた条件で定量的に検証することである．具体的には，(i) 数値特徴量のみ（フェーズ 1）と，(ii) GDELT 由来のニュース特徴量を日次集計して追加した条件（フェーズ 2）を同一の分割・評価手順で比較し，4 モデル（Logistic Regression / LightGBM / LSTM / Transformer）における性能変化を明らかにする．

さらに，平均的な局面のみでは文脈情報の効果を

見落とし得るため，検証集合全体（main）に加えて，将来リターンの絶対値 $|r_H|$ が大きい急変動日集合（subset）を抽出し，通常局面と異常局面で効果が異なるかを評価する．評価は macro-F1，balanced accuracy，混同行列により行い，ニュース統合が「全体の汎化性能」と「急変動局面での頑健性」に与える影響を検証する．

ここまでで「何を検証したいか（目的）」を定めた．次に，その目的に到達するために本研究が採る設計上の要点（評価の切り分け，リーク対策，特徴量化，比較の仕方）を整理し，本研究の位置づけとしてまとめる．

1.4 本研究の位置づけ

本研究の位置づけを以下にまとめる．

1. **異常局面を明示した評価設計**：株価日足の方向性予測を三値分類として定式化し，ボラティリティに基づく閾値 $\tau_t(H)$ でラベルを定義することで，通常局面（main）に加えて急変動日（subset）を分離して性能を評価できる枠組みを与える．
2. **時系列リークを抑えた検証手順**：時系列評価における将来情報混入（look-ahead）を避けるため，Purged Walk-Forward Embargo（PWFE）分割（検証区間直前の embargo 除外）を採用し，評価の信頼性を高める．
3. **長期バックテストに適したニュース特徴量化**：2013–2025 年の全期間を同一仕様で扱うため，GDELT の GKG（日次ファイル）を用いてニュースを取得し，記事数・ソース多様性・トーン統計・V2Tone 成分等を取引日単位で日次集計して特徴量化する．また公開時刻の不整合や非取引日に起因するリークを避けるため，引け後・非取引日のニュースは翌取引日に繰り上げる安全側のアラインを実装する．
4. **ニュース統合効果の系統的比較**：上記の設定を共通条件とし，ニュースなし（フェーズ 1）とニュース統合（フェーズ 2）を同一条件で比較することで，4 モデル（Logistic Regression / LightGBM / LSTM / Transformer）における文脈情報統合の有効性を main/subset の二系統で検証する構成とした．

1.5 論文構成

本論文の構成を以下に示す．まず Section 2 で株価予測, ニュース統合, 異常局面評価, および時系列リーク対策に関する関連研究を整理する．次に Section 3 でデータ取得，数値・ニュース特徴量の設計，ラベル定義（main/subset），および PWFE 分割を含む評価手順を述べる．Section 4 では比較対象とする 4 モデルの学習手順と前処理をまとめ，Section 5 で実験設定（期間，ハイパーパラメータ，指標）を明示する．Section 6 でフェーズ 1（数値のみ）とフェーズ 2（ニュース統合）の結果を main および subset の両面から提示し，Section 7 で結果の解釈，限界，および今後の課題を議論する．

特に本研究では，課題 (1)「リークを避けたニュースのアライン」と，課題 (2)「急変動局面での評価」を手法章（Section 3）と結果章（Section 6）で対応付けて示す．以降，関連研究の整理を通じて，本研究がどこに新規性（評価設計とリーク耐性の同時満足）を置くかを明確にする．

2 関連研究

序論で示した課題は，「予測モデルの表現力」だけでなく，「文脈情報の取り込み方」と「時系列評価の信頼性」にまたがる．そこで本章では，(i) 数値系列のみでの株価予測，(ii) ニュース・センチメント統合，(iii) リーク（look-ahead）を避ける評価手法の 3 軸で先行研究を整理し，本研究がどの論点を継承し，どこに工夫を置くかを位置づける．この位置づけを踏まえ，次章では本研究の手法（特徴量化・アライン規則・PWFE 分割）を具体化する．

2.1 株価予測と時系列モデル

金融時系列の予測は古くから研究されており，統計モデルから機械学習・深層学習まで多様な手法が提案されている．近年は深層学習（LSTM 等）や Transformer により，非線形性や長期依存を表現しようとする試みが増えている．LSTM は長期依存の学習に有効な RNN として提案され [14]，Transformer は自己注意により並列計算と長距離依存の表現を可能にした [15]．

2.2 ニュース・センチメントの統合

ニュースや SNS から抽出したセンチメントを金融予測に統合する研究がある．例えば，イベント（出来事）を抽出し株価変動と結び付けるイベント駆動型のアプローチが報告されている [19]．

2.3 時系列リーク対策（Purging / Embargo）

時系列データでは，標準的な K-fold 分割（ランダムなシャッフル）を行うと，未来の情報を含むサンプルが訓練側へ混入し，評価が過大推定されうる（look-ahead bias）．L'opez de Prado は金融 ML における purging/embargo の重要性を整理し [20]，Purged K-fold 等の枠組みを提示している．本研究ではこれを簡略化した Purged Walk-Forward Embargo（PWFE）を用い，検証区間直前の学習データを embargo として除外することでリークを抑える．

以上の関連研究整理から，本研究は「モデル自体の新規性」よりも，**(a) ニュースをリークなく日次特徴量へ落とす実務的な前処理設計**と，**(b) 急変動局面を明示した評価枠組み**に焦点を当てる．次章では，この焦点を具体的なデータ処理・特徴量・分割手順として定式化する．

3　検証手法

本章では, データ取得 → 数値特徴量設計 → ニュース取得とアライン → ラベル定義（main/subset）→ 分割（PWFE）→ 学習・評価の順に,「どこでリークを防ぎ，どこで急変動局面を評価するか」を明示する.

3.1　研究全体の流れ

本研究は,「ニュース（文脈情報）を統合することが汎化性能に与える影響」を切り分けて評価するため，同一のデータ期間・同一の分割法（PWFE）・同一の評価指標の下で，**入力特徴量のみ**を切り替える 2 フェーズ比較として設計する．評価は検証集合全体（main）に加え，将来リターンの絶対値 $|r_H|$ が大きい日を抽出した部分集合（subset）でも行い，急変動局面での挙動を明示的に確認する（subset は実装では上位 $p = 0.10$ を用いる）.

フェーズ 1（数値特徴量のみ）価格・出来高・テクニカル指標および市場要因から構成される数値特徴量のみを用い，4 モデル（Logistic Regression / LightGBM / LSTM / Transformer）を同一条件で学習・評価し，ベースライン性能を確立する.

フェーズ 2（ニュース統合）フェーズ 1 の数値特徴量に加えて，GDELT（GKG 日次ファイル）由来のニュース特徴量を統合する．ニュースは取引日へのアライン（引け後・非取引日の繰り上げ）を行ったうえで，記事数・ソース多様性・トーン統計・V2Tone 成分等を日次集計し，さらに短期 rolling および急増度（surprise）を付与した特徴量としてモデルへ入力する．フェーズ 1 と同一の分割・指標で再評価し，main および subset で性能の変化を比較する.

3.2　データ取得と前処理

3.2.1　株価・市場関連データ

対象銘柄の日次 OHLCV（Open/High/Low/Close/Volume）は `yfinance` を用いて取得する [4]．実装では `yfinance.download` において `auto_adjust=False` とし，株式分割・配当などの調整を行わない価格系列を用いる（以降のリターン計算は主に `Close` に基づく）．対象銘柄の例としてトヨタ自動車（`7203.T`）を扱う.

市場要因として，日経平均（`^N225`），S&P500（`^GSPC`），VIX（`^VIX`）を同様に取得し，対象銘柄の取引日に整列した上で，各系列のリターンやボラティリティ推定値（例：20 日ローリング標準偏差）を特徴量化する（Section 3.3）．TOPIX については，`yfinance` で指数ティッカー（`^TOPX`）が取得不能となる場合があるため，実装既定では TOPIX ETF（`1306.T`）を参照系列として用いる．また市場系列の欠損処理は系列ごとに異なる．TOPIX（取得失敗時は 1306.T）系列のみ対象銘柄の取引日に `reindex` した上で最大 5 日まで前方補完（`ffill(limit=5)`）し，一方で, 日経平均（`^N225`），S&P500（`^GSPC`），VIX（`^VIX`）は日付キーで左結合するのみで，欠損が残る場合は後段の `dropna` により当該行が学習・評価から除外される.

3.2.2　技術的指標の算出

取得した OHLCV から，`ta` 等のライブラリを用いてボリンジャーバンド幅（`bb_width`），ATR（`atr`），RSI（`rsi14`），MACD（`macd`, `macd_sig`），ストキャスティクス（`stoch_k`, `stoch_d`），OBV 系（`obv_ratio`）などを算出する [5]．

3.2.3　欠損処理と有効期間

ローリング計算を含むため，期間先頭付近には欠損が生じる．実装では特徴量作成後に無限大を欠損へ置換し，特徴量・ラベルに欠損が残る行は学習・評価から除外する．また市場系列の欠損は，TOPIX（または代替 ETF）についてのみ最大 5 日まで前方補完する．それ以外の市場系列で残る欠損は，特徴量・ラベル整列後の `dropna` により除外される.

3.3　数値特徴量設計

日足 OHLCV から，価格水準そのものに依存しにくい**比率・変化率（リターン）・標準化量（Z スコア）**を中心に特徴量を設計した．目的は, (i) 終値水準の違いによるスケール差を抑え，(ii) トレンド・ボラティリティ・出来高などの状態量を分解して表現し，(iii) 急変動や局所的な異常を捉えやすくすることである.

■(1) **リターン（変化率）**　終値の**複数期間リターン**を用いる．具体例として, `ret1`，`ret5`，`ret10`，`ret20`（それぞれ 1/5/10/20 営業日の `pct_change`）を作成する．これにより短期〜中期の変化方向と大きさを同時に表現する.

■(2) **トレンド乖離（移動平均からの距離）**　移動平均に対する乖離率として，`devMA5`，`devMA20`，`devMA60`（例：$\mathrm{Close}/\mathrm{MA} - 1$）を用いる．水準ではなく「平均からどれだけ離れているか」を使うことで，水準差の影響を受けにくくする.

■(3) **ボラティリティ・レンジ**　ローリング標準偏差（`sigma5`，`sigma20`）やその比（`sigma_ratio`），ボリンジャーバンド幅（`bb_width`），ATR 系（`atr`，`atr_ratio`，`atr_diff`），日中レンジの移動平均比（`range_ma_ratio`）を用い，変動の大きさ（不確実性）を特徴量化する.

■(4) **テクニカル指標（モメンタム等）**　`rsi14`，`macd`，`macd_sig`，`stoch_k`，`stoch_d` に加え, モメンタム（`mom10`，`mom20`）や短期 EMA 差（`ema_diff`）を作成し，過熱感・トレンド転換の兆候を捉える.

■(5) **ローソク足形状**　実体・ヒゲを比率で表し，`body_ratio`，`upper_shadow_ratio`，`lower_shadow_ratio` を用いる．これにより，値幅の方向性（実体優位か，ヒゲ優位か）をスケール不変に表現する.

■(6) **トレンド構造（回帰ベース）** 価格系列に対し短期窓での線形回帰を行い，傾き（slope30，slope60）と残差を得る．残差はローリングで標準化し，resid30_z，resid60_z として局所的な逸脱を表現する．

■(7) **出来高・フロー** 出来高の変化率（vol_chg），移動平均比（vol_ratio），週次相当の比（vol_wk_ratio），および OBV 比（obv_ratio）を用い，需給の変化を数値化する．

■(8) **ギャップと異常度** 前日終値と当日始値の差分として gap を作成し，ローリング平均・分散に基づく異常度（gap_anom，price_anom，vol_anom）を付与することで，急変の「どれくらい珍しいか」を表現する．

■(9) **市場要因と相対強度** 市場全体の影響を入れるため，指数・VIX 等に対してもリターン・ボラを作成し，nikkei_ret1，nikkei_sigma20，sp500_ret1，sp500_sigma20，vix_ret1 等を入力へ含める（実装に基づく）．さらに銘柄と市場（TOPIX 代理系列を含む）の差分として，10 日累積の相対強度 rel_strength10 を作成する．なお TOPIX 系列は取得失敗時に代替系列を用いるため，利用可否に応じて入力列が自動的に調整される（Section 5）．

■(10) **スケール差の緩和（$\log(1+x)$）** 正の値を取りやすい一部の変数については $\log(1+x)$ 変換を行い，*_log1p 列として追加する（例：sigma20_log1p，bb_width_log1p，atr_log1p 等）．

最終的にモデル入力へ採用する列名の全一覧は付録 Section 付録 B（表 B.1）に集約する．本文では冗長な全列挙を避け，設計意図とカテゴリ構造（どの情報をどの形で表現しているか）を中心に述べた．

3.4 ニュース（文脈情報）の取得

3.4.1 GDELT（DOC / GKG）の概要

GDELT は世界中のニュースを大規模に収集・構造化したデータベースであり，API および日次ファイルとして提供される [1)]．本研究で利用し得る取得手段は大きく以下の 2 系統である．

- **DOC API（Article List）**：記事検索のための API であり，直近の記事取得に適する一方，取得可能期間は概ね過去 3 か月程度に制約されるため，長期バックテスト全区間を同一仕様で収集する用途には不向きである [2)]．
- **GKG（日次ファイル）**：日次 ZIP（YYYYMMDD.gkg.csv.zip）として過去分が公開されており，長期間（年単位）の履歴を遡って取得できるが,title や text などの文章情報は提供されていない.[3)]．

本研究では予測対象期間（2013–2025 年）を一貫した方式でカバーするため，**全期間にわたり GKG（日次ファイル）を用いてニュース情報を取得**した．DOC API は取得可能期間の制約により長期バックテストの全区間を同一仕様で収集できないため，本研究の実験設定では用いていない [2, 3)]．

3.4.2 クエリ設計

銘柄ごとのニュース抽出にはクエリマップ（JSON）を用いる．本実装で使用した 7203.T には「トヨタ，Toyota Motor Corporation」等の同義語（日本語/英語）を設定し，GKG レコード中の文字列フィールド（URL やメタ情報）に対してキーワード一致により関連記事を抽出する．

Listing 1: 7203.T のクエリマップ

```
{
  "7203.T": [
    "トヨタ",
    "トヨタ自動車",
    "Toyota",
    "TOYOTA",
    "\"Toyota Motor\"",
    "\"Toyota Motor Corporation\""
  ]
}
```

3.4.3 GKG で得られる主なフィールドと V2Tone

GKG レコードにはトーンやテーマ等の構造化情報が含まれる．本研究では特に V2Tone（Tone，Positive，Negative，Polarity，ActivityRefDensity，SelfGroupRefDensity，WordCount）を用いる [3)]．V2Tone は記事テキストを感情辞書で走査して得られる指標であり，Tone は文書全体の平均トーン（−100〜+100，一般的には −10〜+10 付近が多い）で，Positive Score（ポジティブ語に該当した語の割合 [%]）から Negative Score（ネガティブ語に該当した語の割合 [%]）を差し引いて算出される．また Polarity はトーン辞書にマッチした語の割合 [%] であり，Tone がほぼ 0 でも Polarity が高い場合は「肯定語と否定語が拮抗して感情的に強い」状況を示唆する．ActivityRefDensity は能動的（active）な語の割合 [%] でテキストの「能動性」の粗い代理指標，SelfGroupRefDensity は代名詞の割合 [%] で自己/集団参照の強さを表す．WordCount は文書の総語数である（GDELT 公式コードブック参照）[3)]．

3.5 ニュース特徴量化

3.5.1 取引日へのアライン（リーク対策）

ニュースの公開時刻 published_at は UTC/JST の不整合や非取引日が存在するため，そのまま当日特徴量にすると将来情報が混入しうる．本研究では以下の規則でニュースを「有効取引日」へ写像する．

- 非取引日（休日・週末）のニュースは次の取引日に繰り上げる．
- 取引日であっても，引け後に公開されたニュースは翌取引日に繰り上げる（実装では境界時刻を JST 15:30 に設定）．

- GKG は時刻情報が粗く UTC 00:00 に丸められたレコードとして取得するため, すべてのニュースを「引け後扱い（翌営業日に繰り上げ）」としてアラインし，look-ahead を抑制する．

3.5.2 日次集計特徴量

各有効取引日 d について，ニュース群を集計して特徴量ベクトル $\boldsymbol{z}_d$ を作る．本研究のニュース特徴量は，(i) **量（volume）**，(ii) **極性/強度（tone, V2Tone）**，(iii) **多様性（source）**，(iv) **文字列フィールドの粗い要約（themes/organizations 等）**を，取引日 d に対して日次で集計したものである．さらに短期窓（例：3/5 営業日）の rolling 集計と，長期窓（例：20 営業日）に対する急増度（surprise）を加え，「ニュースが多い/少ない」や「急に増えた」といった状態変化を数値化する．

なお，本研究のニュース特徴量は GKG 由来の数値シグナル（`tone`, V2Tone 成分, `gkg_numarts` 等）を**取引日単位に集計**するものであり，title/body を用いたテキスト特徴量（埋め込み等）は作成しない．また, 欠損日は「ニュースなし」とみなして 0 補完する．

ニュース特徴量の列名一覧は付録 Section 付録 C（表 C.1）にまとめる．

3.6 ラベル定義（三値分類）

本研究は翌営業日方向性の三値分類を行う．取引日インデックスを t とし，終値 P_t に対して予測ホライズン H 取引日先の将来リターンを

$$r_H(t) = \frac{P_{t+H}}{P_t} - 1 \tag{1}$$

と定義する（実装では $H = 1$）．また, 当日ボラティリティ推定として，1 取引日リターン $\mathrm{ret1}_t = \frac{P_t}{P_{t-1}} - 1$ の 20 日ローリング標準偏差

$$\sigma_{20}(t) = \mathrm{Std}\left(\mathrm{ret1}_{t-19}, \ldots, \mathrm{ret1}_t\right) \tag{2}$$

を計算し，閾値を

$$\tau_t(H) = k_\tau \, \sigma_{20}(t) \sqrt{H} \tag{3}$$

と定義する（実装では $k_\tau = 0.3$）．このときクラス $y_t \in \{0, 1, 2\}$ を

$$y_t = \begin{cases} 0 & (r_H(t) < -\tau_t(H)) \\ 1 & (|r_H(t)| \le \tau_t(H)) \\ 2 & (r_H(t) > \tau_t(H)) \end{cases} \tag{4}$$

で定義する．ここで $y = 1$ は「横ばい（中立）」を意味する．

さらに異常局面の強度指標として

$$a_t = |r_H(t)| \tag{5}$$

を併せて保持し，検証区間内で a_t が上位 p%（実装では $p = 0.10$，すなわち上位 10%）の集合を subset として評価に用いる（Section 5.4）．

3.7 入力窓とデータセット化

前処理後の有効取引日を $t = 0, 1, \ldots, N-1$ とし，各取引日 t の特徴量ベクトルを $\boldsymbol{x}_t \in \mathbb{R}^F$ とする．ラベル y_t は Section 3.6 で定義した三値分類ラベルであり，H 取引日先リターン $r_H(t)$（実装では $H = 1$）に基づく．なお $r_H(t)$ は将来値 $t + H$ を参照するため末尾 H 行は欠損となり，実装では `dropna` により除外される．

時系列入力は窓幅 $T = 60$ の固定長系列である．実装では，

$$\boldsymbol{X}_t = [\boldsymbol{x}_{t-T}, \ldots, \boldsymbol{x}_{t-1}] \in \mathbb{R}^{T \times F}, \qquad t = T, \ldots, N-1 \tag{6}$$

と定義する．すなわち，**ラベルが付与された取引日 t 当日の特徴 $\boldsymbol{x}_t$ は入力に含めず**，直前 T 取引日分のみから y_t を予測する．この定義では先頭 T 取引日は系列を構成できないため，有効サンプル数は $N - T$ となる（末尾側の H 日分は前処理段階で既に除外済み）．

モデルごとの入力は以下のとおりである．時系列モデル（LSTM/Transformer）は $\boldsymbol{X}_t \in \mathbb{R}^{T \times F}$ をそのまま入力し，系列依存を内部で学習する．一方，Logistic Regression および LightGBM は，窓方向にプーリングして $\boldsymbol{z}_t \in \mathbb{R}^F$ を作り，$\boldsymbol{z}_t$ を入力として分類する．プーリングは設定により

$$\text{avg:} \quad \boldsymbol{z}_t = \frac{1}{T} \sum_{i=1}^{T} \boldsymbol{x}_{t-i}, \tag{7}$$

$$\text{last:} \quad \boldsymbol{z}_t = \boldsymbol{x}_{t-1} \tag{8}$$

のいずれかを用いる（既定は `last`）．LightGBM はこの $\boldsymbol{z}_t$ を前処理段階で生成して学習し，Logistic Regression はモデル内で同等のプーリングを行った上で線形分類する．

3.8 分割法：Purged Walk-Forward Embargo（PWFE）

金融時系列の評価では，データを無作為に分割する交差検証（例：シャッフルを伴う K-fold）を用いると，時間順序が崩れて将来情報が訓練側へ混入し，性能が過大推定される（look-ahead bias）おそれがある．そこで本研究では，時系列順を保ったまま訓練・検証を分割し，さらに境界近傍の混入を抑えるため，Purged Walk-Forward Embargo（PWFE）により外側検証を行う [20]．

■分割の考え方（図 1・Algorithm1） データを時系列順に $K = n_{\text{splits}}$ 個の連続ブロックへ分割し，fold k ではブロック k を検証区間（VAL）とする．訓練区間（TRAIN）は「VAL より過去の全データ」から構成するが，VAL 開始点 s の直前 $E =$ embargo 日分は訓練から除外する（embargo）．これは Algorithm1 における $t_{\text{end}} = \max(0, s - E)$ の計算に対応し，学習インデックスを $\{0, \ldots, t_{\text{end}} - 1\}$ として切り詰める

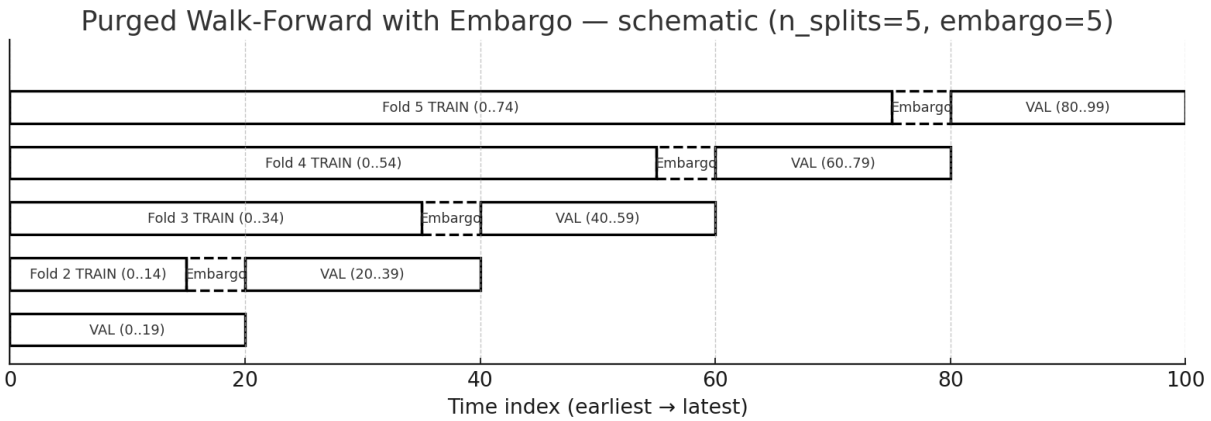

図 1: PWFE 分割の模式図（$n_{\text{splits}} = 5$, embargo=5）.

Algorithm 1. PWFE 分割（実装に基づく概略）

Input: サンプル数 N，分割数 K，embargo$E$
Output: fold ごとの $(\mathcal{I}_{\text{tr}}^{(k)}, \mathcal{I}_{\text{va}}^{(k)})$
$S \leftarrow \lfloor N/K \rfloor$ `// base fold size`
$r \leftarrow N \bmod K$ `// remainder`
for $i = 0, \ldots, K-1$ **do**
  fold_sizes$[i] \leftarrow S$
  `// fold_sizes` $= [N//K] \times K$
end
for $i = 0, \ldots, r-1$ **do**
  fold_sizes$[i] \leftarrow$ fold_sizes$[i] + 1$ `//` **余り** $N\%K$ **を先頭** `fold` **へ配分**
end
$s \leftarrow 0$ `//` **検証区間の開始位置（累積）**
for $k = 0, \ldots, K-1$ **do**
  $e \leftarrow \min(s + \text{fold\_sizes}[k], N)$
  $\mathcal{I}_{\text{va}}^{(k)} \leftarrow \{s, \ldots, e-1\}$
  $t_{\text{end}} \leftarrow \max(0, s - E)$ `//` `embargo` **を除外**
  $\mathcal{I}_{\text{tr}}^{(k)} \leftarrow \{0, \ldots, t_{\text{end}} - 1\}$
  if $|\mathcal{I}_{\text{tr}}^{(k)}| <$ min_train ***or*** $|\mathcal{I}_{\text{va}}^{(k)}| <$ min_val **then**
    skip fold
  end
  $s \leftarrow e$ `//` **次** `fold` **へ進める**
end

ことで実現している．図 1 の点線部がこの embargo に対応し，例えば Fold 3 では VAL が (40..59) のブロックであるのに対し，TRAIN は (0..34) までで，(35..39) の 5 日間を学習から除外している．

■embargo を入れる理由（リークの具体例） 本研究ではラベルが「$H = 1$ 日先リターン」に基づくため (Section 3.6)，境界近傍のサンプルでは「訓練側ラベルの計算に用いる将来価格」が検証区間と時間的に近接しやすい．また特徴量も，移動平均や分散などのローリング計算を含むため，境界直前のデータを訓練に含めると，検証直前の局所的な情報（直近の急変動など）を通じて評価が楽観的になるおそれがある．そこで VAL の直前 E 日分を訓練から除外し，境界近傍の依存（時間的近接による過大評価）を抑制する．特に本研究は $H = 1$ であり，$E = 5$ は安全側の設定として機能する．

■本研究での具体的な設定と fold 数 実装の既定値は

$$n_{\text{splits}} = 5,\ \text{embargo} = 5,\ \text{min\_train} = 252,\ \text{min\_val} = 63$$

である．有効行数 $N = 2778$ を $K = 5$ に分割すると，各ブロック長は概ね $S = \lfloor N/K \rfloor \approx 555$ となる．ただし最初の fold では，VAL より過去に十分な訓練データが存在しないため，手順 1 の条件 ($|\mathcal{I}_{\text{tr}}^{(k)}| <$ min_train) によりスキップされる．結果として，本研究では 4 fold で外側検証を行った (Section 5.2)．

■実装上の追加のリーク対策（前処理の fit 範囲） 分割後の前処理は各 fold で独立に行い，`StandardScaler` の `fit` や，Mutual Information (MI) による特徴量選択は訓練区間のみに基づいて推定し，検証区間には `transform` のみを適用する．これにより，「統計量の推定に検証データが混入する」という別種のリークも抑制している．

3.9 特徴量選択（Mutual Information）

高次元の入力特徴量に対し，学習データのみに基づいて相互情報量（Mutual Information; MI）によるフィルタ型の特徴量選択を行う．実装では `mutual_info_classif`（離散ラベルに対する MI 推定）を用い [8]，fold 内の訓練データをさらに内部分割し，**内側訓練（inner train）のみ**で MI を推定して，

各特徴量について推定した MI が閾値 $\epsilon = 10^{-4}$ を超えるもののみを採用する．

4 使用モデル

前章で定義したデータセット（窓化入力・ラベル・PWFE 分割）に対し，本章では比較対象 4 モデルの学習方法をまとめる．ここでは「モデルごとに何が違い，何が共通か」を先に宣言するため，まず共通の入力形式・前処理・学習手順を述べ，その後に各モデル固有の構造へ進む．

4.1 共通：入力形式と学習手順

本研究では，各時刻 t に対し過去 T 日分の特徴量系列 $\boldsymbol{X}_t \in \mathbb{R}^{T\times F}$（Section 3）を構成し，PWFE の各 fold で学習・評価する．前処理として，`StandardScaler` は outer train を時間順に **inner train / inner validation** へ分割したうえで，**inner train のみに** `fit` し，inner validation および outer validation へは `transform` のみを適用する [9]．さらに特徴量選択（MI）を用いる場合も，**学習データのみに基づいて**採否を決める（リーク防止）．

また本実装では，学習データ（outer train）を時間順に **inner train / inner validation** へ分割し，末尾 20%（最小 64，最大は学習データの半分まで）を inner validation として EarlyStopping 等に用いる [17]．

入力形式はモデルにより異なる．LSTM/Transformer は $\boldsymbol{X}_t$（$T \times F$）をそのまま入力する一方，Logistic Regression/LightGBM は系列方向を

$$\boldsymbol{z}_t = \mathrm{Pool}(\boldsymbol{X}_t) \in \mathbb{R}^F$$

で集約してから入力する．ここで $\mathrm{Pool}(\cdot)$ は実装上 `avg`（時系列平均）または `last`（最終時刻）であり，既定は `last` である．

4.2 Logistic Regression（線形ソフトマックス分類器）

本研究の「Logistic Regression」は，時系列窓 $\boldsymbol{X}_t$ を $\boldsymbol{z}_t = \mathrm{Pool}(\boldsymbol{X}_t)$ で集約し，線形写像＋ softmax により三値分類を行う（多項ロジスティック回帰に相当）．

$$\boldsymbol{p}_t = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{z}_t + \boldsymbol{b}) \tag{9}$$

推定クラスは $\hat{y}_t = \arg\max_c p_{t,c}$ とする．実装では，$\boldsymbol{z}_t$ に対して Dropout は挿入せず（線形＋ softmax の 1 層），出力層を学習し，最適化は Adam，損失は（label smoothing 無効時）`SparseCategoricalCrossentropy` を用いる．またクラス不均衡には，学習データから推定したクラス重み（`class_weight`）で対処する．

4.3 LightGBM（GBDT）

LightGBM は勾配ブースティング決定木（GBDT）を高速化した手法であり [10]，多クラス目的（`multiclass`）で学習する [13]．本研究の実装では，時系列窓 $\boldsymbol{X}_t$ をそのままフラット化するのではなく，$\boldsymbol{z}_t = \mathrm{Pool}(\boldsymbol{X}_t) \in \mathbb{R}^F$ へ集約してから入力する．学習は `LGBMClassifier` で行い，inner validation に対する `multi_logloss` を監視して early stopping を適用する．またクラス不均衡には，学習データから推定したクラス重みを `sample_weight` として反映する．

4.4 LSTM

LSTM はゲート機構により長期依存を学習する RNN である [14]．本研究では双方向 LSTM を 2 層重ね，出力系列を `last` で集約し，Dropout の後に Dense（softmax）1 層で三値分類する．損失は交差エントロピー，最適化は Adam[16] を用い，EarlyStopping および学習率スケジューラを用いる [17, 18]．

4.5 Transformer

Transformer は自己注意（Self-Attention）により系列間相関を捉える [15]．本研究では固定の位置エンコーディングを加えた後，Transformer Encoder block（LayerNorm ＋ MHA ＋残差，LayerNorm ＋ FFN ＋残差）を複数段積層し，系列表現を `last` で集約して，MLP（出力層）で三値分類する．

4.6 ハイパーパラメータ

本文ではモデルの要点を中心に述べ，主要ハイパーパラメータの一覧は付録 Section 付録 A（表 A.1）にまとめる．

5 実験設定

本章では，結果を解釈する際に前提となる実験条件をまとめて固定する．具体的には，(i) データ期間・対象銘柄，(ii) 前処理後の有効行数と特徴量次元，(iii) 分割法（PWFE）および subset の定義，(iv) 評価指標を整理する．これにより，次章で示すフェーズ 1/2 の比較結果（表 3, 表 4）を，「同一条件下で入力のみを変えた差分」として一貫して読み取れるようにする．

5.1 データ期間・対象銘柄

本研究では対象銘柄としてトヨタ自動車（`7203.T`）を用い，株価・市場要因は `yfinance` から日足データを取得した．取得期間は `start=2013-04-01`, `end=2025-12-31` を指定しており，取得された OHLCV の最終取引日は `2025-12-30` であった．期間内の取引日数は 3140 日である（ニュース特徴量の結合対象日数）．

特徴量算出に必要なローリング窓（例：σ_{20} 等）および欠損除去後，分割に投入できる「有効行」は 2778 行となり，範囲は 2013-09-19 から 2025-12-29 である．以降の PWFE 分割・学習・評価はこの 2778 行を用いる．(付録 Section 付録 A（表 A.2）)

5.2 データの基本統計

本研究のデータ基本統計を付録 Section 付録 A(表 A.2) に示す．ニュース特徴量は取引日単位で結合され，結合前の特徴量列数 78 に対し，ニュース由来 53 列が追加され，結合後は 131 列となった．

その後，ラベルや中間列を除いた学習入力の特徴

表 1: MI 特徴量選択におけるニュース特徴量の採用状況（transformer の実行ログ）

fold	selected(F)	news_selected	news_ratio(%)
1	66	28	52.8
2	68	30	56.6
3	62	28	52.8
4	53	28	52.8
平均	62.25	28.5	53.75

量次元は $d = 107$ となる．さらに，各 fold の inner train に基づき Mutual Information（MI）による特徴量選択を行い，各モデルの入力特徴量を fold ごとに削減する．表 1 は Transformer 実行ログに基づく代表例であり，選択次元は fold により変動した．

PWFE 分割は $n_{\text{splits}} = 5$, embargo=5, min_train=252, min_val=63 とした. 有効行数 2778 に対し，先頭 fold は学習データが不足（train<252）となるためスキップされ，最終的に 4 fold で評価した．また，時系列モデル（LSTM/Transformer）では入力窓幅 $T = 60$ を用いるため，各検証ブロック長 L に対して有効な窓サンプル数は $L - T$ となり，例えば $L = 556$ の場合は 496 サンプルとなる（表 A.3）．

上記の数値（取引日数・有効行数・特徴量次元・有効 fold 数など）の一覧は，本文の流れを優先して付録 Section 付録 A（表 A.2）に集約する．本文では以降，結果の解釈に必要な要点（有効行数 $N = 2778$, 有効 fold 数 4，窓幅 $T = 60$）のみを用いる．

PWFE の fold ごとの訓練/検証サンプル数および NN 内部分割（inner validation）の詳細は，再現性情報として付録 Section 付録 A（表 A.3）に示す．

5.3 評価指標

本研究では以下を用いる [6, 7]．

- **macro-F1**：各クラスの F1 を平均.
- **balanced accuracy**：各クラス再現率の平均 [6].
- **混同行列**：誤分類の内訳を可視化.

5.4 main 評価と subset 評価

main は各検証区間の全サンプルで評価する．**subset** は検証区間内の異常日集合で評価するが，時系列窓化を行うモデルでは検証ブロック先頭の T サンプルが窓を構成できず除外されるため，subset の抽出母集団は窓化後の検証集合 $\tilde{\mathcal{I}}_{\text{va}}$（サイズ $|\mathcal{I}_{\text{va}}| - T$）とする．

$$\mathcal{S}_p = \left\{ t \in \tilde{\mathcal{I}}_{\text{va}} : \ a_t \text{が上位 } 10\% \right\}, \quad a_t = |r_H(t)| \tag{10}$$

実装では $k = \max(1, \lfloor p|\tilde{\mathcal{I}}_{\text{va}}| \rfloor)$ 個を選ぶ.

5.5 検証集合におけるラベル分布（main / subset）

評価に用いた外側検証集合（PWFE の validation）におけるラベル分布を表 2 に示す．main（検証集合の全サンプル）では 3 クラスが存在する一方，subset（$|r_H|$ 上位 10%）では中立クラス（$y = 1$）が 0 件となった．これは subset が「絶対変動の大きい日」を抽出する定義であるため，閾値内に収まる中立ラベルが選ばれにくいことによる．このとき macro-F1 や balanced accuracy はクラス欠損の影響を受けるため，混同行列と併せて解釈する必要がある．

5.6 フェーズ 1 ／フェーズ 2 の比較手順

1. フェーズ 1：ニュース特徴量を用いず，4 モデルを評価する．
2. フェーズ 2：ニュース特徴量を統合し，同条件で再評価する．

6 結果

本節では，フェーズ 1（数値のみ）とフェーズ 2（ニュース統合）の性能を，同一の分割・前処理・評価指標の下で比較する．以降の結果は，PWFE 外側検証で得た各 fold の予測を結合した **pooled** 集計を主として示し，fold 間のばらつきをならした全体傾向を確認することを目的とする．序論の課題 (2)（急変動局面での有効性）に対応して，まず main（全検証サンプル）で平均的な汎化性能の変化を示し，次に subset（異常日）で急変動局面に限った挙動の変化を示す．

さらに，指標だけでは見えにくい誤分類傾向や特徴の寄与を**補助的に把握する**ため，混同行列と Permutation Importance（MDA）による特徴量重要度を**代表例**として提示する．

6.1 総合結果（main）

表 3 に記載.

6.2 subset 結果（異常日）

表 4 に記載.

6.3 混同行列（代表例：数式による提示）

本研究では，精度指標（macro-F1, balanced accuracy）に加えて，誤分類の内訳を混同行列で確認する実装を行った．ただし，全 fold・全モデルの混同行列を掲載すると分量が過大となるため，本節では**代表的な 1 例**のみを示す．以下の混同行列は，Transformer（ニュースあり）の **pooled main**（外側検証 fold の予測を結合）で得られた混同行列である．各行の和が表 2 の main（pooled, $N = 1982$）に一致することからも，pooled 集計であることが確認できる．クラス順は {down(y=0), neutral(y=1), up(y=2)} とし，行が真のラベル，列が予測ラベルを表す．

$$\mathbf{C}^{(\text{rep})}_{\text{main}} = \begin{pmatrix} 235 & 205 & 235 \\ 198 & 176 & 192 \\ 238 & 250 & 253 \end{pmatrix}, \tag{11}$$

表 2: 検証集合のラベル分布

集合	down($y=0$)	neutral($y=1$)	up($y=2$)
main (pooled,$N=1982$)	675 (34.1%)	566 (28.6%)	741 (37.4%)
subset top10% (pooled,$N=196$)	92 (46.9%)	0 (0.0%)	104 (53.1%)

表 3: main 結果

モデル	フェーズ 1（数値のみ）		フェーズ 2（ニュース統合）	
	macro-F1	bal. acc.	macro-F1	bal. acc.
Logistic Regression	0.332172	0.339152	0.353197	0.356976
LightGBM	0.329829	0.330392	0.349798	0.350153
LSTM	0.334890	0.338991	0.325760	0.336064
Transformer	0.341312	0.343847	0.333113	0.333511

表 4: subset 結果（$|r_H|$ 上位 10%）

モデル	フェーズ 1（数値のみ）		フェーズ 2（ニュース統合）	
	macro-F1	bal. acc.	macro-F1	bal. acc.
Logistic Regression	0.312461	0.429139	0.284436	0.357232
LightGBM	0.295167	0.372910	0.318668	0.410117
LSTM	0.256121	0.321279	0.219676	0.259615
Transformer	0.286105	0.378344	0.296148	0.380644

ここで c_{ij} は「真のクラス i が予測クラス j に分類された件数」である．

6.4 特徴量重要度

ニュース統合の効果を「精度」だけでなく，モデルが依存した特徴の傾向として可視化するため，Permutation Importance（MDA: Mean Decrease in Accuracy）を算出・保存する実装を行った．MDA は学習済みモデルを固定したまま，検証データ上で特定の特徴量のみをランダムに入れ替え，評価指標がどれだけ低下するかで重要度を測る**モデル非依存**の手法である．

■図 2 の読み方 横軸は baseline からの macro-F1 低下量（Score drop vs baseline）である．本例（Transformer・ニュースあり・Fold4）では `slope30` や `news_v2tone_self_3bd_mean`, `bb_width_log1p` の低下量が相対的に大きく，fold4 の検証データ上でこれらが予測に効いていることを示す．またニュース由来特徴（`news_v2tone_*`, `news_gkg_numarts_*` 等）が上位に含まれており，少なくとも当該 fold ではニュース特徴がモデル出力に影響していることが確認できる．ただし MDA は相関の強い特徴同士で重要度が分散し得るため，「単一特徴の因果効果」ではなく「モデルが依存している度合い」として解釈する必要がある．図中の誤差棒は，反復試行におけるスコア低下量のばらつき（標準偏差）であり，値が大きいほど重要度の推定が不安定であることを表す．

ただし，これも全 fold・全モデル分を掲載すると冗長となるため，本節では**代表的な 1 例**のみ図として掲載する．

7 考察

本章では，結果章で得られた性能差を解釈する．ただし本研究では，差分が小さい場合に実行ばらつきの影響を受け得ること，また subset ではクラス欠損により指標が不安定になり得ることに留意する必要がある．これらの前提を確認したうえで，main で改善が限定的だった理由，ニュース統合でモデル間の精度が分かれた理由を順に考察し，最後にニュースアラインのトレードオフと今後の課題を述べる．

7.1 実験結果の要約

表 3（main）および表 4（subset）より，ニュース統合（フェーズ 2）の効果はモデルと評価集合によって異なった．main では Logistic Regression と LightGBM で指標が改善した一方，LSTM と Transformer

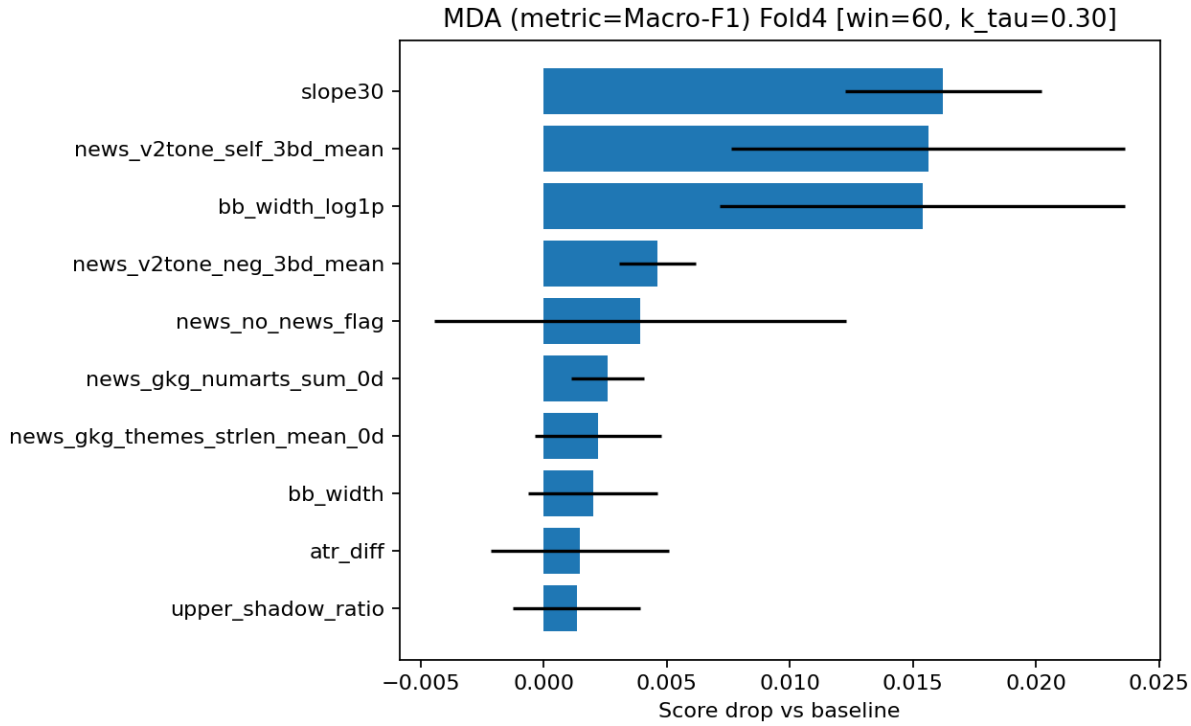

図 2: 特徴量重要度（transformer(ニュースあり) の Fold4 の場合）

では改善が見られなかった．具体的には，main の macro-F1 は Logistic Regression で +0.021，LightGBM で +0.020 程度の増加であったが，LSTM は −0.009，Transformer は −0.008 程度の減少であった．balanced accuracy も同様に，Logistic Regression と LightGBM では増加，LSTM と Transformer では横ばい〜減少となった．

subset ($|r_H|$ 上位 10%) では LightGBM が macro-F1 で +0.024，balanced accuracy で +0.037 と相対的に改善した一方，Logistic Regression と LSTM は大きく低下した．Transformer は macro-F1 がわずかに改善したが，balanced accuracy の改善は限定的であった．以上から，**日次集計型のニュース特徴量は，非線形な分岐・相互作用を表現できる木モデル（LightGBM）で効果が出やすい一方，時系列 NN（LSTM/Transformer）では入力次元増加や信号希釈の影響により性能が悪化し得る**ことが示唆される．また subset はサンプル数が小さく（$N = 196$），指標の解釈には一層の注意が必要である（Section 7.3）．

7.2 再現性（乱数固定下でも残るばらつき）

本研究では乱数シードを固定して学習を行ったが，深層学習フレームワーク内部の非決定的計算（並列化・演算順序・バックエンド最適化等）や学習の早期終了点の差により，同一設定でも実行ごとに結果がわずかに変動することがある．実際，本研究の実行環境では，全モデルで**指標が概ね ±1% 程度**の範囲で揺らぐ．したがって，表 3 および表 4 に示した差分が小さい場合は，誤差の範囲内であると主張する．

7.3 subset 評価における指標解釈の注意

subset は $|r_H|$ 上位 10% という定義のため，検証集合において neutral クラスが 0 件となった（表 2）．このとき macro-F1 や balanced accuracy は，**クラス欠損の影響で値が不安定になり得る**．例えば，予測が neutral に寄ると「存在しないクラスへの誤分類」が増え，指標が急激に悪化する．従って subset では，(i) down/up の二値に落として評価する，(ii) subset の定義を「上位 p%」ではなく「$|r_H| > \tau$」等にして neutral が完全に消えないよう調整する，(iii) 混同行列（誤分類の方向）を主として解釈する，といった補助が必要である．

本研究の結果では，subset で LightGBM が改善した一方で Logistic Regression と LSTM が大きく悪化した．これは，subset が「方向が明確な大変動日（down/up 中心）」で構成されるため，**非線形な相互作用（例：記事数急増＋市場ボラ上昇）**を拾えるモデルが有利になり，線形モデルや学習が不安定になりやすい NN では性能が崩れた可能性がある．

7.4 main での改善が限定的だった理由

main の指標は全体として 0.33 〜 0.36 程度に留まり，検証集合のラベル分布（表 2）は up が約 37%，down が約 34%，neutral が約 29% で，クラス不均衡は極端ではない．この条件において，価格系列のテクニカル指標と市場要因（指数・為替等）がすでに一定の説明力を持つため，ニュースを加えても平均的（main）な改善が小さくなりやすいのだと考えられる．

加えて，本研究のニュースは GKG 由来の数値シグナル（V2Tone 等）を日次に集計しており，「単発イベントの意味内容」を直接表現していない．そのため，通常局面（main）ではニュースが持つ追加情報が薄まり，記事数やトーン統計が**安定した予測信号としては弱い**可能性がある．

さらに，ニュース統合が有効に働きにくい仮説として，**市場がニュース公開より先に情報を織り込んでいる（先行織り込み仮説）**が考えられる．実務的には，機関投資家や海外投資家を含む参加者が，企業開示の予兆，需給，関連市場（ADR，先物，為替），あるいは速報・噂レベルの情報を通じてポジション調整を行い，**記事として観測されるタイミング以前に**価格へ反映されることがある．この場合，ニュース特徴量（特に日次集計のトーンや記事数）は「原因」ではなく「結果（既に動いた後の報道）」に近くなり，$H = 1$ の短期方向性予測に対しては追加情報として機能しにくい．この仮説は，ニュースの公開時刻と価格変動の先後関係をより高頻度データで検証することにより，将来の課題として検討できる．

7.5 ニュース統合によるモデル別の精度

■Logistic Regression　main では改善したが subset では大きく低下した．線形分類器は高次元特徴の追加に対して比較的頑健である一方，subset のようなレジーム（極端局面）では「ニュース × 市場 × 直近ボラ」などの非線形相互作用を表現できず，誤分類が増えたと考えられる．加えて本実装では L2 正則化を用いておらず（付録 Table A.1），ニュース特徴量がノイズとして作用した場合に性能が崩れやすい可能性がある．

■LightGBM　main と subset の両方で相対的に改善が見られた．LightGBM は非線形性と特徴間の交互作用を表現でき，日次集計のニュース特徴量（記事数，トーン統計，V2Tone 成分，GKG 文字列要約）が持つ**非線形な信号**を活用しやすい．また欠損や 0 埋めが混じる特徴に対しても比較的扱いやすい点が寄与した可能性がある．

■LSTM / Transformer　ニュース統合で性能が改善しなかった．原因として，(i) サンプル数に対して入力次元が増え過学習しやすい，(ii) ニュース特徴量は「日次の集計」であり，60 日窓に積んでも系列パターンとしての情報が薄い，(iii) ニュースのアラインを安全側に倒した結果，情報が翌日に繰り越され信号が希釈した，などが考えられる．特に Transformer では MI によりニュース特徴量が一定割合採用されているが（表 1），これは「訓練データで相関があった」ことを示すに留まり，**外側検証での汎化に寄与するとは限らない**．また本研究の窓定義では，ラベル日 t 当日の特徴 $\boldsymbol{x}_t$ を入力へ含めず直前 T 日のみで予測する（Equation (6)）ため，当日ニュースの即時効果を捉えにくい点も影響し得る．今後は，次元削減（より厳しい MI 閾値等），正則化の強化，あるいは複数銘柄で学習してデータ量を増やすことが必要である．

7.6 ニュースアライン（リーク対策）と信号希釈のトレードオフ

本研究ではリークを避けるため，引け後ニュースを翌営業日に繰り上げ，さらに GKG の時刻粒度が粗いため, 翌営業日扱いとした．この設計は look-ahead を抑える一方で，「実際には引け前に市場へ織り込まれたニュース」も翌日に回すため，ニュースが持つ即時性の信号を弱める可能性がある．したがって，リーク耐性と信号の強さにはトレードオフがある．今後は，(i)GKG 以外の手段で, ニュースの時刻粒度が細かいデータを取得する, (ii) 取引時間内のニュース密度などの, より強固な時間特徴を追加する，などの改善が考えられる．

7.7 今後の課題

本研究には以下の課題がある．

- 単一銘柄での検証であり，ニュース統合の効果が銘柄依存である可能性を排除できない．
- subset の定義上，neutral クラスが欠損し，macro-F1 や balanced accuracy の解釈が難しい．
- GKG 由来の数値シグナル中心であり，記事本文の意味内容（埋め込み等）を利用していない．
- 改善幅が小さいため，fold ごとのばらつき（平均 ± 標準偏差）や有意性の議論が必要である．
- ニュースのアラインを安全側に倒しているため（Section 3.5），即時性のある信号が希釈される可能性がある．
- 乱数固定下でも内部計算の非決定性により指標が揺らぐ可能性があり，差分が小さい場合の解釈には注意を要する．

今後は，(i) 複数銘柄での学習・評価，(ii) subset をイベント種別（決算，政策，災害等）で分解した分析，(iii) ニュースの意味表現（見出し/本文の埋め込み）による特徴量化，(iv) subset 評価の指標設計（二値化等）を含む評価の再設計，(v) ニュース時刻と価格変動の先後関係を検証し，先行織り込み仮説を定量的に評価, を進めることで，文脈情報統合の効果をより明確に検証できると考える．

付録 A　実験設定の詳細

本文では議論の流れを優先し，実装既定値のハイパーパラメータや, 使用した具体的なデータ数を本付録に集約する．

付録 B　数値特徴量一覧

本付録は，実装で**作成・使用**する数値特徴量の列名を列挙する．

付録 C　ニュース特徴量一覧

本付録は，日次で集計してモデル入力へ追加するニュース特徴量（全列）を列挙する．

付録 D　ソースコード（GitHub リポジトリ）

本研究で用いた実装コード一式は GitHub リポジトリとして公開している．論文中にソースコード全文を掲載すると冗長となるため，本付録では参照先のみを明示する．

- GitHub リポジトリ：`https://github.com/59GauthierLab/Uchida_Research`

表 A.1: 主なハイパーパラメータ（実装既定値）

モデル	主な設定	値（既定）
共通（データ）	窓幅/ホライズン	$T = 60$, $H = 1$
	閾値係数/異常日割合	$k_\tau = 0.3$, $p = 0.10$
	分割（PWFE）	$n_{\text{splits}} = 5$, embargo=5, min_train=252, min_val=63
Logistic Regression	Dropout / L2	なし / なし
	学習率 / epoch	Cosine LR: base_lr=1e-4, alpha=0.07 / 80（early stop）
LightGBM	objective / metric	multiclass / multi_logloss
	num_leaves / max_depth	31 / -1
	subsample / colsample	0.9 / 0.8
	reg_alpha / reg_lambda	0 / 0
	num_boost_round / early stop	400 / 50
LSTM	BiLSTM ユニット数	128（2 層）
	Dropout	0.2
	学習率 / epoch	Cosine LR: base_lr=1e-4, alpha=0.07 / 80（early stop）
Transformer	block 数 / head 数	2 / 4
	d_model / ff_dim	F（MI 後の次元；fold 依存）/ 128
	Dropout	0.2
	学習率 / epoch	Cosine LR: base_lr=1e-4, alpha=0.07 / 80

表 A.2: データ概要

項目	値
ニュース結合対象の取引日数（2013-04-01..2025-12-30）	3140
最終的に分割へ投入できる有効行（2013-09-19..2025-12-29）	2778
ニュース結合前の列数 → 結合後の列数	78 → 131
ニュース追加列数（added_cols）	53
学習入力の特徴量次元（MI 前）	$d = 107$
PWFE の有効 fold 数（先頭 fold は train 不足で skip）	4

表 A.3: PWFE（外側）および NN 内部分割のサンプル数（窓幅 $T = 60$）

fold	tr_raw	iv_raw	va_raw	tr_win	iv_win	va_win
fold 1	441	110	556	381	50	496
fold 2	886	221	556	826	161	496
fold 3	1331	332	555	1271	272	495
fold 4	1775	443	555	1715	383	495

表 B.1: 数値特徴量一覧

カテゴリ	列名（モデル入力）
リターン	ret1, ret5, ret10, ret20
トレンド乖離	devMA5, devMA20, devMA60
ボラ/レンジ	sigma5, sigma20, sigma_ratio, bb_width, atr, atr_ratio, atr_diff, range_ma_ratio
テクニカル/モメンタム	rsi14, macd, macd_sig, stoch_k, stoch_d, mom10, mom20, ema_diff
ローソク足形状	body_ratio, upper_shadow_ratio, lower_shadow_ratio
トレンド構造	slope30, slope60, resid30_z, resid60_z
出来高/フロー	vol_chg, vol_ratio, vol_wk_ratio, obv_ratio
ギャップ/異常度	gap, gap_anom, price_anom, vol_anom
市場要因	topix_ret1, topix_sigma20, nikkei_ret1, nikkei_sigma20, sp500_ret1, sp500_sigma20, vix_ret1, rel_strength10
$\log(1+x)$ 追加列	sigma5_log1p, sigma20_log1p, bb_width_log1p, atr_log1p, atr_ratio_log1p, range_ma_ratio_log1p, topix_sigma20_log1p, nikkei_sigma20_log1p, sp500_sigma20_log1p, vol_wk_ratio_log1p, vix_log1p

表 C.1: ニュース特徴量一覧

カテゴリ	列名（モデル入力）
記事数・多様性/アライン比率	news_count_0d, news_source_nunique_0d, news_after_close_ratio_0d, news_no_news_flag
記事数（rolling / surprise）	news_count_3bd_sum, news_count_5bd_sum, news_count_surprise_20bd
GKG numarts（強度）	news_gkg_numarts_sum_0d, news_gkg_numarts_mean_0d, news_gkg_numarts_3bd_sum, news_gkg_numarts_5bd_sum, news_gkg_numarts_surprise_20bd
GKG themes 要約	news_gkg_themes_items_sum_0d, news_gkg_themes_items_nunique_0d, news_gkg_themes_strlen_mean_0d
GKG organizations 要約	news_gkg_organizations_items_sum_0d, news_gkg_organizations_items_nunique_0d, news_gkg_organizations_strlen_mean_0d
tone 統計（0d/rolling）	news_tone_mean_0d, news_tone_sum_0d, news_tone_min_0d, news_tone_max_0d, news_tone_abs_mean_0d, news_tone_valid_count_0d, news_tone_valid_ratio_0d, news_tone_pos_ratio_0d, news_tone_neg_ratio_0d, news_tone_3bd_mean, news_tone_5bd_mean
V2Tone pos	news_v2tone_pos_mean_0d, news_v2tone_pos_sum_0d, news_v2tone_pos_3bd_mean, news_v2tone_pos_5bd_mean
V2Tone neg	news_v2tone_neg_mean_0d, news_v2tone_neg_sum_0d, news_v2tone_neg_3bd_mean, news_v2tone_neg_5bd_mean
V2Tone pol	news_v2tone_pol_mean_0d, news_v2tone_pol_sum_0d, news_v2tone_pol_3bd_mean, news_v2tone_pol_5bd_mean
V2Tone act	news_v2tone_act_mean_0d, news_v2tone_act_sum_0d, news_v2tone_act_3bd_mean, news_v2tone_act_5bd_mean
V2Tone self	news_v2tone_self_mean_0d, news_v2tone_self_sum_0d, news_v2tone_self_3bd_mean, news_v2tone_self_5bd_mean
V2Tone wc	news_v2tone_wc_mean_0d, news_v2tone_wc_sum_0d, news_v2tone_wc_3bd_mean, news_v2tone_wc_5bd_mean

参考文献

1) GDELT Project, *The GDELT Project (official site)*. https://www.gdeltproject.org/（最終アクセス：2026/01/16）.
2) GDELT Project, *GDELT DOC 2.0/2.1 API*（公式ブログ/ドキュメント）. https://blog.gdeltproject.org/gdelt-doc-2-0-api-supports-jsonfeed/（最終アクセス：2026/01/16）.
3) GDELT Project, *GKG / V2Tone Codebook*（公式コードブック）. https://www.gdeltproject.org/data.html（最終アクセス：2026/01/16）.
4) R. Ran Aroussi, *yfinance: Yahoo! Finance market data downloader (GitHub)*. https://github.com/ranaroussi/yfinance（最終アクセス：2025/09/18）.
5) D. B. Santos, *Technical Analysis Library in Python (ta)*. https://github.com/bukosabino/ta（最終アクセス：2025/09/18）.
6) scikit-learn developers, *sklearn.metrics.balanced_accuracy_score*. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html（最終アクセス：2025/09/18）.
7) scikit-learn developers, *scikit-learn metrics API reference*. https://scikit-learn.org/stable/api/sklearn.metrics.html（最終アクセス：2025/09/18）.
8) scikit-learn developers, *mutual_info_classif (Feature selection)*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html（最終アクセス：2025/09/18）.
9) scikit-learn developers, *StandardScaler*. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html（最終アクセス：2025/09/18）.
10) G. Ke, Q. Meng, T. Finley, *et al.*, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in *Proc. NeurIPS*, 2017.
11) LightGBM developers, *Python API: lightgbm.train*. https://lightgbm.readthedocs.io/en/v3.2.1/pythonapi/lightgbm.train.html（最終アクセス：2025/09/18）.
12) LightGBM developers, *Python API: lightgbm.early_stopping*. https://lightgbm.readthedocs.io/en/v3.2.1/pythonapi/lightgbm.early_stopping.html（最終アクセス：2025/09/18）.
13) LightGBM developers, *Parameters (LightGBM documentation)*. https://lightgbm.readthedocs.io/（最終アクセス：2025/09/18）.
14) S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
15) A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention Is All You Need,” in *Proc. NeurIPS*, 2017.
16) D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv:1412.6980*, 2014.
17) TensorFlow developers, *tf.keras.callbacks.EarlyStopping*. https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping（最終アクセス：2025/09/18）.
18) TensorFlow developers, *tf.keras.callbacks.LearningRateScheduler*. https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/LearningRateScheduler（最終アクセス：2025/09/18）.
19) X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep Learning for Event-Driven Stock Prediction,” in *Proc. IJCAI*, 2015.
20) M. López de Prado, *Advances in Financial Machine Learning*. Wiley, 2018.