scientific reports

Check for updates

OPEN

A hybrid self attentive linearized phrase structured transformer based RNN for financial sentence analysis with sentence level explainability

Md Tanzib Hosain[1,4], Md Kishor Morol[2,4] & Md Jakir Hossen[3]✉

As financial institutions want openness and accountability in their automated systems, the task of understanding model choices has become more crucial in the field of financial text analysis. In this study, we propose xFiTRNN, a hybrid model that integrates self-attention mechanisms, linearized phrase structure, and a contextualized transformer-based Recurrent Neural Network (RNN) to enhance both model performance and explainability in financial sentence prediction. The model captures subtle contextual information from financial texts while maintaining explainability. xFiTRNN provides transparent, sentence-level insights into predictions by incorporating advanced explainability techniques such as LIME (Local Interpretable Model-agnostic Explanations) and Anchors. Extensive evaluations on benchmark financial datasets demonstrate that xFiTRNN not only achieves a remarkable prediction performance but also enhances explainability in the financial sector. This work highlights the potential of hybrid transformer-based RNN architectures for fostering more accountable and understandable Artificial Intelligence (AI) applications in finance.

All relevant data regarding traded assets is reflected in prices within an open market in financial markets[1]. Outperforming the markets consistently is a formidable challenge due to the continuous adjustment of positions and prices by market participants in response to new information. However, the advent of novel information retrieval techniques has the potential to alter the interpretation of "new information," offering short-term advantages if these technologies are adopted promptly. Analyzing financial documents, such as news articles, analyst reports and corporate announcements, can reveal valuable insights. Given the sheer volume of textual data generated daily, manual analysis becomes impractical, making automated sentiment or polarity analysis through natural language processing (NLP) techniques increasingly essential for evaluating texts from financial entities[2]. Traditional NLP models have made significant strides in this area, but the need for transparent and explainable predictions remains a pressing concern. As of on 2012, Knight Capital Group (https://www.cnbc.com/2012/08/02/the-knight-fiasco-how-did-it-lose-440-million.html) deployed a faulty trading algorithm that executed millions of unintended orders in 45 min, losing $440 million and nearly collapsing the firm. This incident underscores a critical challenge in modern finance: complex AI systems can amplify risks when their decision-making processes remain opaque. Similar issues persist today, such as the 2024 NSE's (https://www.reuters.com/business/finance/indias-nse-pays-765-mln-settle-algo-trading-software-case-2024-10-04/) $76.5 million settlement, where allegations were related to unfair access to its algorithmic trading software. These real-world failures highlight the urgent need for explainable AI (xAI) in financial applications, where transparency is not merely advantageous but a regulatory and ethical imperative.

The role and responsibility of AI have grown immensely, prompting concerns about its trustworthiness. A significant portion of these concerns stems from the prevalent use of "black-box" models, particularly Deep Neural Networks (DNNs). The complexity of these models, with their vast parameter spaces and intricate

[1]Department of Computer Science and Engineering, American International University-Bangladesh, 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh. [2]Department of Computing and Information Science, Cornell University, 616 Thurston Ave, Ithaca, NY 14853, USA. [3]Center for Advanced Analytics (CAA), COE for Artificial Intelligence, Faculty of Engineering & Technology (FET), Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, 75450 Melaka, Malaysia. [4]Explainable LLM and Interpretable Technology Ensemble Lab, 17010 Cedarcroft Rd, Queens, NY 11432, USA. ✉email: jakir.hossen@mmu.edu.my

algorithms, renders them largely uninterpretable to humans, making it difficult to fully understand their decision-making processes . Consequently, these models may harbor biases, relying on unjust, outdated, or incorrect assumptions that can be missed using traditional methods of model effectiveness evaluation . This opacity ultimately erodes trust in such models .

The term, xAI has arisen to solve the drawbacks of conventional AI. The creation of Machine Learning (ML) methods that produce incredibly successful models that are also explicable, enabling people to comprehend, control and have faith in them is what Barredo Arrieta *et al.* refers to as xAI. According to[3], xAI models are intended to offer explanations or specifics that make their operation understandable. Consequently, the use of xAI helps to make models more reliable, secure and error-free.

This paper is driven by the need to assess market sentiments in real-time amidst ongoing financial crises, leading to a transition from traditional methods to real-time media analysis[4]. In the financial industry, sentiment analysis is still difficult even with major advances in natural language processing (NLP)[2]. This calls for the use of domain-specific word lists and adaptive learning algorithms. To maintain confidence and openness in financial decision-making processes, these sophisticated models' opacity and complexity necessitate strong explainability. Particularly in the fields of financial sentence prediction and natural language processing, explainability is needed. With BERT-based systems, there have already been attempts to improve transparency. For instance, exBERT is an interactive dashboard that shows the attention and internal representations of a model, allowing users to create hypotheses regarding the reasoning process of the model[5]. An alternative method is employed by van Aken et al. using visBERT[6]. VisBERT monitors token changes as they go through the network's layers in light of research that suggests attention mechanisms might not be the best choice for explanations[7]. It takes hidden states out of each Transformer encoder block and maps them into a 2D space using Principal Component Analysis (PCA) so that semantic links may be deduced from token distances[8].

We present the Linearized Phrase Structure (LPS)[9] model using xFiTRNN, with a distinct focus on understanding financial sentiment at the sentence level—a key differentiator from other studies in the field. This model is designed to categorize financial and economic text fragments from an investor's perspective into three groups: positive, negative and neutral. Focusing on the sentence level is crucial in financial contexts because financial documents often contain complex narratives where sentiments can vary significantly within a single document. By analyzing sentiment at the sentence level, the LPS model can accurately capture these nuanced shifts, providing a detailed sentiment profile that is essential for informed investment decisions. With the assessment of explainability techniques—LIME[10] and Anchors[11]—the LPS model seeks to capture crucial interactions between financial concepts and phrase structures, offering comprehensible and easily understood insights into model predictions.

Though recent work in financial NLP has explored hybrid architectures, similar like xFiTRNN, these approaches often treat documents as homogeneous units, missing critical sentence-level nuances. Financial narratives frequently contain mixed sentiments (e.g., "Strong Q3 growth despite regulatory headwinds"), requiring granular analysis. Our Linearized Phrase Structure (LPS) model addresses this gap through three key innovations: (1) syntactic-aware feature extraction that identifies sentiment-carrying phrases, (2) dynamic attention mechanisms weighting these phrases by financial relevance, and (3) integrated explainability interfaces optimized for domain experts.

Moreover, though the xAI landscape offers multiple explanation paradigms, each with trade-offs. For instance, SHapley Additive exPlanations (SHAP)[12] values provide global feature importance but struggle with NLP's sequential nature. Integrated Gradients (IG)[13] require baseline selection that may distort financial text interpretations. xFiTRNN strategically combines LIME's local explanations—which identify phrase-level contributors through perturbed samples—with Anchors' rule-based summaries ("The prediction remains positive if 'outperform' appears regardless of other terms"). This dual approach caters to financial analysts' needs, offering both granular insight into specific predictions and actionable decision rules. A detailed comparison of our tested LIME/Anchors performance with SHAP and IG is shown in section "Additional tests".

This paper's principal contributions are:

- We propose a hybrid sentiment analysis strategy, xFiTRNN specially designed for the financial domain.
- We integrate the transformer-based FinBERT architecture, BiGRU network and self-attention mechanism with linearized phrased-based feature extraction techniques.
- We evaluate word embedding strategies in order to determine the most reliable preprocessing and embedding approaches that are necessary for precise financial sentiment analysis performance.
- We test the xFiTRNN model against seven conventional ML models, one stacking model and five DL models in our comprehensive trials to evaluate its performance and show its adaptability.
- We execute extensive experimental evaluation of xFiTRNN on benchmark Financial Phrasebank and IMB-SEntFiN datasets, demonstrating superior predictive performance and enhanced transparency compared to traditional models.
- We improve comprehension and confidence in the xFiTRNN model's predictions by offering insights into the explainability of the model through thorough analysis employing LIME and Anchor methodologies.

The rest of the paper is organized as follows: To begin with, section "Related works" provides an overview of the pertinent literature on the subject and lays out the framework for this study, along with some of its limitations. Afterwards, the research approach and methods employed in this study are described in section "Methodology". Next, the study's analyzed findings are shown in section "Results analysis". Later, section "Explainability tests" analyzed the explainability test's results of the proposed model.. Section "Model efficiency and deployment scalability" discusses model efficiency and deployment scalability and section "Multilingual robustness" provides future plan to test XFiTRNN on multilingual robustness. Subsequently, section "Practical implementation plan

for practitioners" describes practical implementation plan of the proposed model and section "Discussion and future works" offers an in-depth analysis of the subject matter together with a critical assessment of the findings and their implications for future study. Section "Conclusion" finally brings the paper to a close.

Related works

Recent studies in financial sentiment analysis using DL and ML reveal significant advancements. Huang et al.[4] show that ChatGPT 3.5 outperforms FinBERT in forex sentiment analysis through effective zero-shot prompting. Mishev et al.[2] present a novel method using sequential minimal optimization with a decision tree, achieving 89.47% accuracy in Twitter sentiment analysis. Fatouros et al.[14] introduce a sentence-level analysis approach, improving accuracy and context preservation in financial news sentiment analysis. Sousa et al.[15] evaluate various algorithms for financial sentiment analysis, including DL models, highlighting the importance of preprocessing stages. Naresh and Venkata develop FinBERT, a finance-specific model excelling in sentiment classification and ESG issue identification[16]. Aslam et al.[17] find that NLP transformers outperform traditional lexicon-based methods, especially with limited data. Ahmad and Umar demonstrate the effectiveness of convolutional neural networks in analyzing StockTwits sentiments[18]. Ahmad et al.[19] achieve high accuracy with their LSTM-GRU ensemble model for cryptocurrency sentiment analysis. Daudert proposes integrating diverse text sources for enhanced sentiment analysis, introducing a novel multi-text approach[20]. Consoli et al.[21] present FiGAS, an unsupervised method for fine-grained sentiment analysis, outperforming traditional methods. Yadav et al.[22] find that noun-verb combinations are effective for sentiment analysis in financial news, despite limitations in dataset size. Xing et al.[23] dissect error patterns in sentiment analysis and introduce StockSen, a new corpus for financial sentiment analysis. Du et al.[24] advocate for hybrid models integrating symbolic and subsymbolic methods for targeted aspect-based sentiment analysis. Zhang et al.[25] introduce a retrieval-augmented framework, improving LLM performance for sentiment analysis. Štrimaitis et al.[26] assess sentiment analysis methods for Lithuanian financial news, identifying effective classification algorithms. Liapis et al.[27] demonstrate that LSTM models excel in multivariate financial time series forecasting with sentiment analysis. Du et al.[28] develop FinSenticNet, a concept-level lexicon for financial sentiment analysis, surpassing traditional methods. Hansen and Borch emphasize the role of alternative data, including sentiment analytics, in investment management and algorithmic trading[29]. Hartmann et al.[30] find state-of-the-art language models superior to traditional methods in sentiment analysis. Chang et al.[31] combine aspect-level sentiment analysis with visual analytics to assess airline industry customer satisfaction during the COVID-19 crisis. Zhang et al.[32] survey emotion fusion strategies for mental illness detection in social media. Singh et al.[33] review the application of RL and DRL in financial decision-making, highlighting their superior performance. Leelawat et al.[34] analyze sentiments in tourism-related tweets using ML algorithms, finding support vector machines most effective. Kuma et al.[35] review sustainable finance research and advocate for the integration of AI and big data techniques. Alsayat introduces an ensemble DL model for sentiment analysis in social media during the coronavirus pandemic, demonstrating superior accuracy[36]. Multimodal architectures, the combination of textual sentiment with market data to enhance both predictive performance and explainability are emerging nowadays. Passalis et al.[37] fuse news sentiment with trading volumes, enabling models to validate language patterns against market reactions—for instance, distinguishing between "rally" in bullish versus pump-and-dump contexts. However, their late fusion approach (concatenating text and numerical features) obscures cross-modal interactions. Recent work by Valle-Cruz et al.[38] demonstrates the promise of early fusion: aligning tweet sentiment scores with real-time price movements in a joint embedding space. While improving hedge signal detection, this complicates explanations as market data dominates text features in gradient-based attributions.

Furthermore, there has been an increasing focus on developing explainable financial sentence prediction approaches[39]. Fatouros et al.[40] highlighted the use of attention mechanisms in sentence-level analysis, which not only improved accuracy but also provided insights into which parts of the text influenced the model's decisions. This approach helps analysts understand the key drivers behind sentiment classifications. Ardizzone et al.[41] advocated for hybrid models that combine symbolic methods (which are more interpretable) with subsymbolic methods (like neural networks). This integration allows for targeted aspect-based sentiment analysis, offering a clearer understanding of how specific features contribute to predictions. Du et al.[28] developed FinSenticNet, a concept-level lexicon that enhances the explainability of sentiment analysis in financial texts. By providing concept-level insights, this model surpasses traditional methods in explaining how different sentiment scores are derived. Chang et al.[31] explored the combination of aspect-level sentiment analysis with visual analytics. This approach aids in explaining model outputs by visually representing how different aspects of a text, such as sentiment towards specific entities, are analyzed, thus enhancing user understanding and trust. Additionally, these models can provide more granular explanations of sentiment shifts, especially in response to potentially manipulative inputs.

However, current xAI approaches in finance exhibit distinct trade-offs. While Fatouros et al.[14] use attention weights to highlight sentiment keywords, high attention to "dividend" might reflect either positive yield or negative tax implications. LIME[10] generates local explanations by perturbing input phrases, crucial for analyzing earnings calls where negation scopes (e.g., "not sustainable") determine sentiment. However, its random sampling struggles with financial jargon's sparsity. IG[13] attributes predictions to input features via path integrals, but requires careful baseline selection—a zero-vector baseline may misattribute significance to stopwords in SEC filings. Anchors[11] create IF-THEN rules (e.g., "IF 'beat estimates' THEN positive"), aligning well with analysts' checklist-driven workflows but oversimplifying compound phrases like "beat estimates despite inflation risks". We selected LIME[10] and Anchors[11] for their complementary strengths: LIME's phrase-level perturbations suit financial sentences' compact structure, while Anchors' rules provide auditable decision logic for compliance purposes. In pilot studies, SHAP[12] underperformed for financial texts—its global feature importance scores

conflated sector-specific terms (e.g., "P/E ratio" mattered equally in tech and banking analyses), reducing actionability.

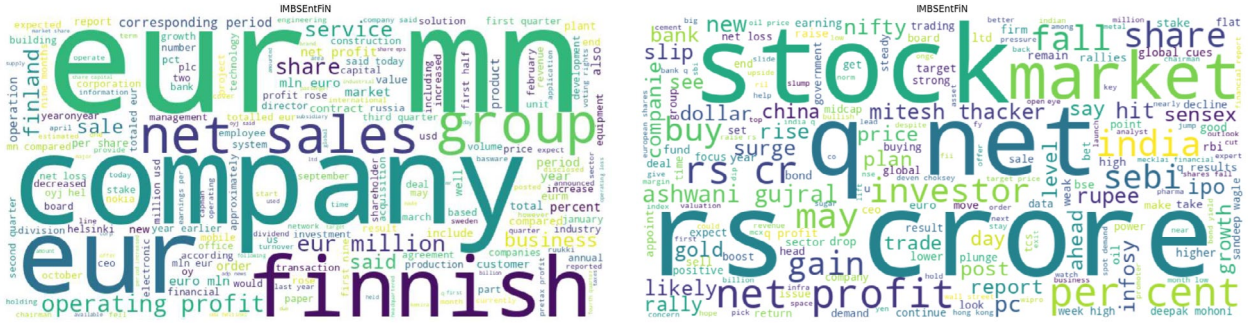
After these vast advancements, there exists many limitations. Fatouros et al.[14] limitations are their study include a constrained dataset duration potentially affecting model performance across different periods, an inability of the short dataset to fully capture the complexities of financial markets and challenges in generative AI such as model collapse when relying heavily on synthetic training data, leading to limited or repetitive model outputs. Lutz et al.[15] limitations include a lack of comprehensive testing on diverse datasets for model validation, which is critical for understanding the complex interplay between news sentiment and market movements. The lack of explainability in many advanced models, especially Deep Learning (DL) ones, poses another limitation, as it becomes difficult to understand the rationale behind certain predictions[42]. Furthermore, the substantial computational resources required for training and maintaining sophisticated models can be a limiting factor[43]. Real-time predictions are also hampered by time lags between the release of financial news and its impact on the market[44,45]. Models trained on data from one financial market or region may not perform well in different contexts due to varying economic conditions and market behaviors, highlighting issues with transferability. There is also a risk of overfitting to historical data, particularly when the dataset is limited, leading to poor generalization to new data[46]. Also, most methods explain static predictions, inadequate for financial time series where sentiment persistence matters (e.g., sustained "inflation fears" vs. one-off mentions and existing tools don't map explanations to FINRA's Model Validation Guidelines (https://www.finra.org/rules-guidance/rulebooks/finra-rules) requiring "traceable logic for material predictions". Leippold shows keyword-based models fail against financial adversarial examples like "growth" → "groWth" (Unicode spoofing)—a vulnerability extending to explanation methods relying on surface tokens[47]. Additionally, financial news and reports often contain human biases, which can be inadvertently learned by the models, resulting in skewed predictions. These additional limitations underscore the complexity of financial markets and the challenges involved in developing robust prediction models. The proposed LPS model tried tackling these through tracking sentiment carriers across sentences via dependency parsing, Anchors rules formatted as MiFID II-compliant decision logs (https://www.esma.europa.eu/publications-and-data/interactive-single-rulebook/mifid-ii) and extending FinBERT's vocabulary to detect glyph attacks.

Methodology

This section details datasets used in this experiment, each model's theoretical foundation, operational principles, evaluation metrics and relevance to the research objectives, focusing on their application in financial sentiment analysis. To uncover patterns in the proposed models' predictions and understand how these patterns influenced the decisions with proper explainability, the following scenarios were examined: when the classifier accurately classified a title as a negative statement, when the classifier accurately classified a title as a positive/neutral statement, when the classifier inaccurately classified a title as a negative statement and when the classifier inaccurately classified a title as a positive/neutral statement.

Datasets

The Financial Phrasebank[48] and SEntFiN[49] datasets used in this work are available to the public. In order to create IMBSEntFiN, we first average the real classes and then alter the class weights for each sentence in the SEntFiN[49] dataset, which was refined into an unbalanced classification format. The dataset has the dependent variable inserted manually. Following this, the dataset was divided into training and testing sets, with 80% of the samples going toward training. The model's predictions were added to the test data in a new column when it finished the classification assignment. Figure 1 presents the most frequent words in the datasets of financial sentences which are displayed in word cloud format.



(a) Financial Phrasebank word cloud

(b) IMBSEntFiN word cloud

Fig. 1. The most frequent words retrieved by python *wordcloud* in the "Financial Phrasebank" (left) and "IMBSEntFiN" (right) datasets of financial sentences are displayed in this word cloud. Each word's magnitude reflects how frequently it occurs in the dataset.

Data preprocessing

Due to the informal and unstructured nature of financial sentence data, preprocessing was necessary to ensure the precision and truthfulness of our study. The following essential stages were part of our extensive data-cleaning procedure:

- **Normalization of text:** Converted all text to lowercase to standardize word recognition and avoid treating words with different capitalization as distinct entities.
- **Removal of irrelevant elements:** Eliminated hashtags (`#topic`), mentions (`@name`) and hyperlinks (e.g., `http`, `https`). Excluded words shorter than two characters and common stop words with minimal relevance to sentiment. Retained negations such as `not` and `no` to preserve sentence context.
- **Handling repeated characters:** Normalized words with repeated characters used for emphasis (e.g., *greeeeat profit* converted to *great profit*) to align with standard lexicons.
- **Expansion of contraction:** Expanded contractions like *isn't* and *can't* into their full forms (e.g., *is not* and *cannot*) to maintain semantic clarity and consistency.
- **Removal of non-alphabetical characters:** Stripped numerals, punctuation marks and special symbols, retaining only alphabetical characters to prevent interference in feature extraction.
- **Deduplication and removal of empty entries:** Identified and removed duplicate sentences and empty entries to ensure dataset integrity.
- **Advanced cleaning for specific methods:** Applied additional techniques, such as stemming and Part-of-Speech (POS) tagging, tailored to the requirements of specific sentiment analysis approaches. These were particularly useful for leveraging financial sentiment resources like specialized lexicons.

Word embeddings

TF-IDF

The scikit-learn module's "TfidfVectorizer" function, which assigns weights to words based on their scarcity throughout the entire dataset and their relevance in individual sentences, was used to apply the TF-IDF embeddings. It also simplified the process of down-weighting frequently used keywords, freeing up our models to focus on more informative terms.

Word2Vec

Word2Vec's power to capture semantic similarities between words is one of its benefits; this enhances the sophistication of text data analysis. It expresses words as dense vectors in a continuous vector space using neural networks. The financial sentences were tokenized using the NLTK library's $word\_tokenize$ function, which breaks sentences down into individual words. A Word2Vec model with the vector size, window size and skip-gram model set was created using the Gensim software. This method was able to represent words as vectors by utilizing a continuous vector space. Full sentences were encoded as vectors using two techniques: average vectorization and sum vectorization.

GloVe

GloVe is a potent word embedding method that builds dense vector representations using a corpus's co-occurrence statistics to capture the semantic associations between words. We trained a GloVe model customized for our dataset using the glove library. Using the $word\_tokenize$ function from the NLTK package, the sentences were tokenized into individual words as part of the preprocessing. After that, a co-occurrence matrix was produced, taking into account a context window to record word connections. The GloVe model was trained to generate embeddings that accurately capture words' global semantic features as well as their local context. The model was then able to identify significant patterns and connections in the text by mapping each word in the dataset to its vector representation using these embeddings.

FastText

FastText is a sophisticated word embedding method that improves on conventional approaches by adding subword information. This makes it useful for handling uncommon or non-vocabulary terms. The FastText model from the Gensim package was used to train dataset-specific embeddings. Using the $word\_tokenize$ function from the NLTK library, the procedure started by tokenizing sentences into individual words. Then, by decomposing words into character-level n-grams and learning their representations, the FastText model was trained to produce word embeddings. This method makes the model resilient for assessing text in many languages and handling invisible words by allowing it to capture both word-level semantics and subword structures. Following training, embeddings for every word in the dataset were available, proving the model's efficacy in capturing morphological and semantic information.

Pretrained transformers

We used the Hugging Face Transformers library's pre-trained transformer-based models to exploit contextual embeddings for text data in our study. Three different transformer models, each contributing special skills to the study, were used. Known for its effectiveness and portability, the "distilbert-base-uncased" model was chosen because it works well in situations with limited computing resources. The left and right context of each word are taken into account while creating context-aware word embeddings. To convey the subtleties unique to finance in text, we employed a cutting-edge "yiyanghkust/finbert-tone" model. Our sentiment analysis efforts were firmly based on this model, which was created for financial sentiment research. We have included "sentence-transformers/all-MiniLM-L6-v2," a complex tool that converts text into a dense vector space. It preserves semantic information while converting whole phrases into fixed-dimensional vectors. While each transformer's

code implementation had a similar structure, the model selection added variety to our testing and allowed us to investigate how contextual embeddings affected our text classification task. We utilized the model's related tokenizer to efficiently tokenize and process our text data, utilizing strategies like truncation and padding to guarantee constant input lengths. For best computational performance, the tokenized data was then effectively processed on the GPU. We also used the model's ability to extract the hidden states linked to the "[CLS]" token, which frequently captures the text's entire context.

Models used

This subsection discusses various ML classifiers. Additionally, it explores an advanced hybrid transformer[50] and attention based[51,52] RNN model, xFiTRNN for our tests as our proposed novel financial sentiment classification framework. This section details each model's theoretical foundation, operational principles and relevance to the research objectives.

Conventional ML models

Ada Boost Classifier: Ada Boost Classifier, or Adaptive Boosting, is a powerful ensemble method to enhance the performance of weak classifiers[53]. It operates by iteratively training a sequence of weak learners, typically decision trees, on various weighted versions of the training data. Initially, all data points are assigned equal weights. A weak classifier $h_m(x)$ is trained in each iteration m and its error rate ϵ_m is defined by Eq. (1):

$$\epsilon_m = \frac{\sum_{i=1}^{N} w_i \mathbb{I}(h_m(x_i) \neq y_i)}{\sum_{i=1}^{N} w_i} \tag{1}$$

where the indicator function is $\mathbb{I}$, the true labels are y_i, the training examples are x_i and the weights are w_i. The weight α_m of the classifier is then calculated by Eq. (2):

$$\alpha_m = \frac{1}{2} \ln \left(\frac{1 - \epsilon_m}{\epsilon_m} \right) \tag{2}$$

which reflects its accuracy. Subsequently, the weights of misclassified instances are increased using Eq. (3):

$$w_i \leftarrow w_i \exp(\alpha_m \mathbb{I}(h_m(x_i) \neq y_i)) \tag{3}$$

focusing the next classifier on harder cases. The final prediction is made by a weighted majority vote of the weak classifiers calculated as Eq. (4):

$$H(x) = \text{sign} \left(\sum_{m=1}^{M} \alpha_m h_m(x) \right) \tag{4}$$

This iterative boosting approach allows the model to effectively capture complex patterns in financial sentiment, leading to more accurate predictions.

Extra Trees Classifier: Extremely Randomized Trees, sometimes referred to as Extra Trees Classifier, is an ensemble learning method that generates the mode of the classes (classification) for individual trees after training a large number of decision trees[54]. It functions by constructing several trees, each of which is trained using random splits in the nodes over the whole dataset. It randomly picks a subset of features F_k and finds the best split among them, as opposed to searching for the best split among all available features for each node split in tree T_k. Equation (5) is used to determine the decision function for a node *n*:

$$f_n(x) = \sum_{j=1}^{d} \mathbb{I}(x_j \leq \theta_{n,j}) \tag{5}$$

where *x* represents the input vector, *d* is the number of features and $\theta_{n,j}$ are the threshold values for the splits. The output of the ExtraTreesClassifier for an input *x* is determined by majority voting from all the individual trees calculated as Eq. (6):

$$H(x) = \text{mode}\{T_k(x)\}_{k=1}^{K} \tag{6}$$

where *K* is the total number of trees. This approach reduces variance and helps in capturing intricate patterns in financial sentiment data, leading to robust and accurate sentiment predictions.

LDA: LDA is a statistical method is used to find the linear combination of qualities that best splits a collection of classes into two or more[55]. It is considered that the data from each class is normally distributed and that all classes have a common covariance matrix. For a given dataset that comprises the classes C_1, C_2. ..C_k, the objective of latent difference analysis (LDA) is to optimize the ratio of within-class variance to between-class variance. Equations (7) and (8) define the within-class scatter matrix S_W and the between-class scatter matrix S_B:

$$S_B = \sum_{i=1}^{k} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{7}$$

$$S_W = \sum_{i=1}^{k} \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T \tag{8}$$

where N_i is the number of samples in class C_i, μ_i is the mean vector of class C_i and μ is the overall mean vector of the dataset. The optimal linear discriminants are found by solving the generalized eigenvalue problem as defined Eq. (9):

$$S_W^{-1} S_B w = \lambda w \tag{9}$$

where λ are the eigenvalues and w are the eigenvectors. The resulting linear discriminants are then used to project the data into a lower-dimensional space for classification. In the context of financial sentiment analysis, LDA effectively reduces dimensionality while preserving class separability, allowing for the accurate classification of sentiment in financial texts.

QDA: QDA is a classification technique in which non-linear decision boundaries are possible since each class is modeled with a unique covariance matrix[56]. QDA operates on the assumption that every class C_k has its own covariance matrix Σ_k, in contrast to Linear Discriminant Analysis (LDA), which operates under the assumption that all classes share a single covariance matrix. For a given input x, the posterior probability for class C_k is calculated using Bayes' theorem, Eq. (10):

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{P(x)} \tag{10}$$

where $P(x|C_k)$ is the class-conditional density function given by the multivariate normal distribution computed as Eq. (11):

$$P(x|C_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\right) \tag{11}$$

with d being the number of features, μ_k the mean vector and Σ_k the covariance matrix for class C_k. The decision rule assigns x to the class with the highest posterior probability defined as Eq. (12):

$$\hat{y} = \arg\max_k \left(\log P(C_k) - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\right) \tag{12}$$

QDA captures the complex relationships and variations in sentiment by leveraging class-specific covariance structures, leading to more nuanced and accurate sentiment classification.

XGB Classifier: XGB Classifier is an implementation of the Extreme Gradient Boosting technique combining the predictions of several weak learners, specifically decision trees[57]. It builds an ensemble of trees one after the other, with each new tree trying to fix the mistakes of the older ones. The model iteratively adds trees to a given dataset $\{(x_i, y_i)\}_{i=1}^{N}$ in order to minimize the objective function, which is specified as Eq. (13):

$$L(\Theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \tag{13}$$

where L is a differentiable loss function (e.g., logistic loss for binary classification), $\hat{y}_i^{(t)}$ is the prediction of the t-th tree and Ω is a regularization term controlling the complexity of the model, with T being the number of leaves and w_j the leaf weights defined as Eq. (14):

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{14}$$

Every time a new tree is fitted, the gradient and hessian of the loss function are utilized to enhance the model's predictions. $\hat{y}_i$ is the ultimate forecast, as determined by (Eq. 15):

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i) \tag{15}$$

XGB Classifier efficiently captures complex patterns and interactions within the data, leading to high-performing and accurate sentiment predictions.

Gradient Boosting Classifier: Gradient Boosting Classifier is a potent ensemble learning technique which creates an additive model step-by-step ahead by successively fitting new models to fix the mistakes caused by

the old ones[58]. Reducing the loss function $L(y_i, F(x_i))$, where $F(x_i)$ represents the model's prediction, for a given dataset $\{(x_i, y_i)\}_{i=1}^{N}$ is the goal. A fresh weak learner $h_m(x)$ is trained to fit the loss function's negative gradient at each step m. This function is the residual error of the current model, which is determined by Eq. (16).

$$r_i^{(m)} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F=F^{(m-1)}} \tag{16}$$

The model is then updated as Eq. (17):

$$F^{(m)}(x) = F^{(m-1)}(x) + \nu h_m(x) \tag{17}$$

where ν is the learning rate. Commonly, decision trees are used as weak learners and the final prediction is given by Eq. (18):

$$F(x) = \sum_{m=1}^{M} \nu h_m(x) \tag{18}$$

Gradient Boosting Classifier effectively captures intricate patterns and trends in sentiment data by iteratively focusing on and correcting the hardest-to-predict instances, leading to highly accurate sentiment classification.

Modern DL models

A variety of DL models, each with unique architectural features, were used to thoroughly assess twitter sentiment categorization. For optimum performance, we tuned the hyperparameters of these models using the Keras Tuner. There were GRU layer units between 128 and 768, dense layer units between 64 and 512, dropout rates between 0.1 and 0.5 and learning rates between 1×10^{-5} and 1×10^{-3} in the search space. We found the optimal settings for each model using Keras Tuner's Random Search technique, which greatly improved their performance. This methodical investigation, together with thorough text representations and adjusted hyperparameters, yielded insightful information about how well different DL architectures performed. A transformer was used to extract features in our first model, the "1-Dense Layered NN." High-level representations were recorded by a single dense layer with 512 units and ReLU activation. We were able to set a benchmark for performance comparison because of this architecture's simplicity. We presented the "2-Dense Layered NN" and "3-Dense Layered NN" to build on this framework. Three successive dense layers with decreasing units (256 and 128) and (512, 256 and 128) were added to gradually improve feature representations after global averaging of the transformer's outputs. To improve model resilience, dropout regularization was specifically implemented after the first dense layer. In order to better investigate the subtleties, we developed the "BiGRU + 3 Hidden Dense Layers" model. The BiGRU layer, a unique kind of RNN, was included into this design to allow the network to recognize sequential relationships in the input data. The features were further refined using three more dense layers (512, 256 and 128 units) after the BiGRU layer. To improve the generalization of the model, dropout was used. In conclusion, we investigated hybrid architecture using the "BiGRU + CNN" model. Here, the model incorporated the advantages of convolutional layers and a BiGRU layer. Feature extraction gained a spatial view from the convolutional layers, which included 64 filters and different kernel sizes. The collected characteristics were then further processed by adding two thick layers (128 and 64 units).

Proposed model

The proposed xFiTRNN model offers an innovative architecture for financial sentiment analysis, as depicted in Fig. 2. This model leverages the power of contextual embeddings from pre-trained transformers, FinBERT, to capture the intricate language used in financial texts. The architecture commences with input layers, namely $input\_ids$' and $attention\_mask$, accommodating sequences up to 256 tokens to ensure comprehensive coverage of the input data For training, the input data is jumbled, batched and arranged into a TensorFlow dataset. A batch size of 16 is chosen to enable effective training. In order to get the labels ready for multiclass classification, they are one-hot encoded. To ensure thorough examination, the dataset is divided into training and validation sets. Keras Tuner plays a key part in FiTRNN model optimization by methodically examining a well defined search space to adjust different hyperparameters. The number of BiGRU layer units, which ranges from 128 to 512 and balances model complexity and performance, is part of the search space. Furthermore, three Dense layers are adjusted with units that range from 128 to 768, 64 to 512 and 32 to 256, respectively, which has an impact on the computing needs and model's capability. While the learning rate is logarithmically explored between 1×10^{-5} and 1×10^{-3} to maximize convergence speed, the dropout rate fluctuates between 0.1 and 0.5 to avoid overfitting. An effective method of exploring the space without doing an exhaustive search is to use Random Search in the tuning process to sample different hyperparameter combinations. Based on validation loss, the tuner determines the optimal hyperparameter settings (Fig. 2) following several trials, guaranteeing an ideal trade-off between computing efficiency and performance.

The core architecture of the FiTRNN model integrates pre-trained contextual embeddings with BiGRU and self-attention mechanisms, featuring the following enhancements:

- **Input layers:** Two primary input layers, $input\_ids$ and $attention\_mask$ receive the tokenized sequences and their corresponding attention masks. This setup ensures proper handling of variable-length sequences by indicating which tokens should be attended to and which are merely padding.

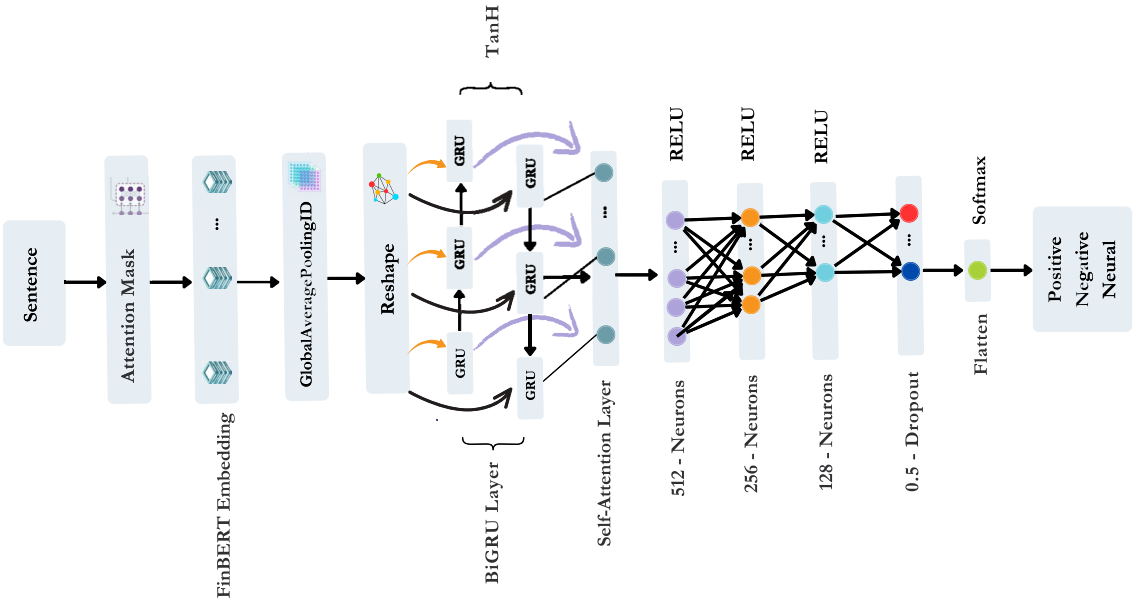
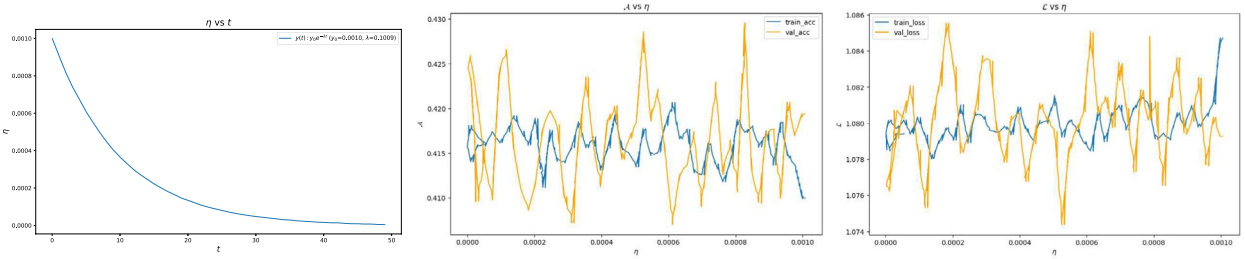


Fig. 2. Architecture of the proposed xFiTRNN model.

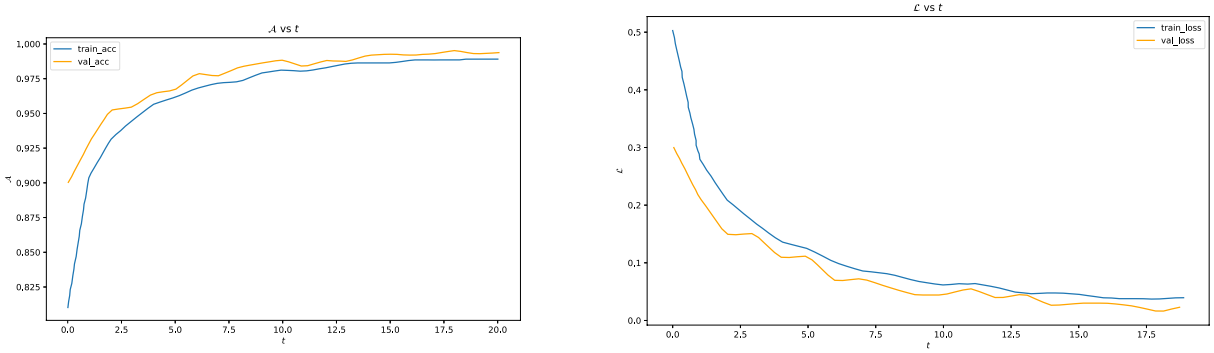
- **Contextual embeddings:** The pre-trained transformer model, FinBERT generates contextual embeddings that encapsulate rich semantic and syntactic information from the input text. These embeddings effectively represent the nuanced language found in financial documents.
- **BiGRU layer:** The Bidirectional Gated Recurrent Unit (BiGRU) layer captures dependencies in both forward and backward directions of the input sequence. Configured with 512 units in each direction, the BiGRU processes the contextual embeddings to understand the sequential nature of financial narratives fully.
- **SA mechanism:** Incorporating a self-attention layer allows the model to focus on the most relevant parts of the input sequence. This mechanism assigns varying levels of importance to different tokens, enabling the model to highlight key financial terms and sentiments that significantly impact the overall interpretation.
- **Dense layers:** A series of dense layers further abstract the features extracted by the BiGRU and self-attention layers. Starting with a 512-unit dense layer activated by ReLU, followed by layers with 256 and 128 units, the model progressively refines its internal representations. Dropout layers with rates 0.05 determined during hyperparameter tuning are interleaved to prevent overfitting.
- **Flatten layer:** A 1D vector is created by flattening the output tensor following processing through the dense layers.
- **Classifier head:** Three units make up the dense layer of the classifier head, which uses the softmax activation function. The final sentiment categorization probabilities of the input financial sentence—whether positive, negative, or neutral—are generated by it.The model is constructed using categorical cross-entropy loss and a learning rate of 1×10^{-5} using the Adam optimizer. A metric for evaluation is categorical accuracy. Callbacks, early halting and model checkpointing were all used to maximize model training. The early halting method keeps track of validation loss and, in order to avoid overfitting, restores the optimal weights. During the training process, the model is fitted to the training dataset and validated for 50 epochs on the validation dataset. However, because of the early stopping mechanism, the iterations end after 25 epochs.

While our model was being trained, the learning rate was modified by the learning rate scheduler callback function. Based on an initial learning rate and an exponential decay factor, which regulates the pace at which the learning rate declines across epochs, the function determines the learning rate for each epoch. The exponential decay formula, which is applied in this particular version, involves multiplying the learning rate by the exponent of a negative constant $\lambda = 0.1009$ times the epoch number. The learning rate may be gradually decreased during training as it exponentially declines as the period grows. As seen in Fig. 3, this method aids in training process optimization by adjusting the learning rate to enhance model convergence and performance throughout several epochs. A thorough assessment of the xFiTRNN model's performance is shown in Figs. 4 and 5. The model's learning progress and convergence are depicted by the training and validation accuracy versus epoch curve, while the loss versus epoch curve shows the model's training and validation loss across successive epochs. When combined, these visualizations provide information on the convergence, classification accuracy and optimization process of the xFiTRNN model. This model's architecture effectively harnesses the strengths of contextual embeddings from pre-trained transformers, the sequential modeling capabilities of BiGRU and the focusing power of self-attention mechanisms. This combination is particularly advantageous for financial sentiment analysis, where understanding context and subtle language cues is critical. The model demonstrates superior performance compared to traditional approaches, offering a robust tool for analyzing sentiments in financial texts. Its ability to capture complex patterns and nuances makes it a valuable contribution to the field of sentiment analysis in finance.



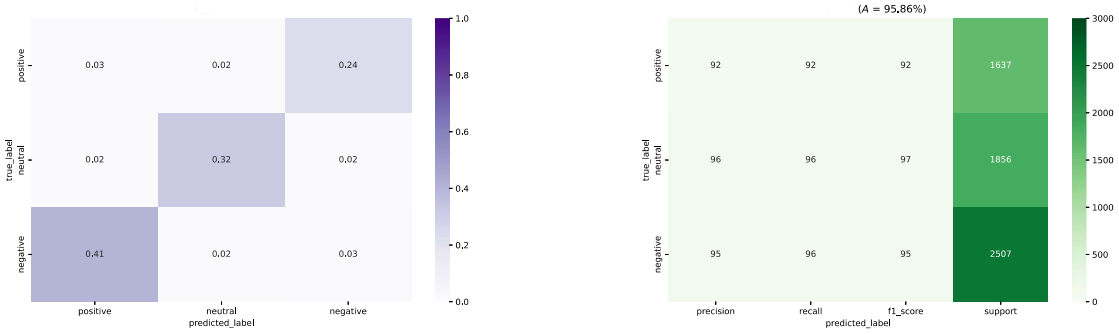
(a) η vs t **(b)** A vs η **(c)** L vs η

Fig. 3. xFiTRNN learning rate over the scheduler callback function; the decaying learning rate as the epoch spreads, left and training and validation accuracy and loss over learning rate, right.



(a) A vs t **(b)** L vs t

Fig. 4. xFiTRNN training and validation accuracy and loss over epoch.



(a) **(b)**

Fig. 5. Metrics performance of the proposed xFiTRNN model.

xFiTRNN core components and workflow

The xFiTRNN model is a hybrid architecture designed for financial sentiment analysis, integrating transformer-based contextual encoding with recurrent neural processing to capture both global context and sequential dependencies in financial texts. Its novel contributions lie in the incorporation of a linearized phrase structure (LPS) to leverage syntactic information and a dual explainability framework for transparent predictions. Figure 2 provides a schematic overview, with detailed configurations available in the supplementary material.

- **Input processing and contextual embeddings:** Financial sentences are tokenized and embedded using FinBERT, a pre-trained transformer model optimized for financial text[4]. FinBERT generates contextual embed-

dings that encapsulate semantic and syntactic nuances, serving as the input representation for downstream processing.

- **LPS integration:** To enhance syntactic awareness, we introduce a linearized phrase structure representation. The sentence's parse tree, derived via dependency parsing, is traversed depth-first to produce a linear sequence of syntactic tokens (e.g., representing phrase heads or relations). These tokens are embedded and concatenated with the FinBERT token embeddings, enriching the input with structural information critical for interpreting complex financial narratives (e.g., "Strong Q3 growth despite regulatory headwinds"). This LPS integration distinguishes xFiTRNN from prior models by explicitly modeling syntax.
- **Transformer encoder:** The combined embeddings (token + LPS) are processed by a transformer encoder employing multi-head self-attention, as described by Peng et al.[50]. This step captures long-range dependencies across the sequence, weighting tokens and syntactic elements by their contextual relevance. For brevity, standard transformer mechanics are not elaborated here; readers are referred to the original work[50].
- **BiGRU layer:** In parallel, the original FinBERT token embeddings are fed into a BiGRU with 512 units per direction. The BiGRU models sequential dependencies in both forward and backward contexts, complementing the transformer's global perspective with localized pattern detection. This is particularly effective for financial texts where sentiment may hinge on term proximity (e.g., "profit" followed by "declined").
- **Feature fusion and classification:** The transformer and BiGRU outputs are concatenated, forming a hybrid representation that integrates global and sequential features. This fused vector is processed through three dense layers (512, 256, and 128 units, ReLU-activated) with dropout (rate 0.05) to refine features and mitigate overfitting. A softmax classifier with three units outputs sentiment probabilities (positive, negative, neutral). The model is optimized using categorical cross-entropy loss and the Adam optimizer (learning rate 1×10^{-5}), with training details in the supplementary material.

Explainability test

For explaining our proposed xFiTRNN model we have performed tests on the following xAI models.

LIME

LIME is a model-agnostic explainability technique that successfully captures a model's behavior locally around a prediction[10]. Building on the notion that complicated models act linearly at a local scale, it perturbs examples close to the prediction in order to train a straightforward, comprehensible linear model. Authors[10] used LIME as the solution computed as Eq. (19) to define the explanation ξ for a sample x:

$$\xi(\mathrm{x}) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_{\mathrm{x}}) + \Omega(g) \tag{19}$$

where G represents interpretable models, $g \in G$ is the explanation model and $\Omega(g)$ measures complexity. $\mathcal{L}(f, g, \pi_x)$ evaluates how well g approximates the model f locally, with proximity measure π_x guiding sample generation. To remain model-agnostic, the method draws samples based on proximity π_x. The optimization process involves perturbing samples around x, recovering them in their original representation and using the predictions $f(z)$ as labels for the explanation model. The approach typically uses sparse linear models, square loss and an exponential kernel for proximity[10].

Anchors

Anchors is an interpretable machine learning technique that provides high-precision rules, called anchors, which are conditions that sufficiently "anchor" a prediction such that changes to the rest of the feature values do not affect the outcome[11]. For a given instance x, an anchor A is a rule set $\{(f_i, v_i)\}$ where f_i is a feature and v_i is its value. The precision of an anchor A for instance x is given by Eq. (20):

$$P(A) = \mathbb{E}[\mathbb{1}_{f(x')=f(x)} \mid A(x') = 1] \tag{20}$$

where f is the model, $\mathbb{1}$ is the indicator function and $A(x') = 1$ indicates that x' satisfies the anchor conditions. In financial sentiment analysis, anchors can be used to generate interpretable rules explaining why a particular sentiment prediction was made. For example, an anchor could be a specific set of words or phrases in a financial report that, when present, consistently lead to a positive or negative sentiment classification. This approach ensures that the explanations are both precise and interpretable, helping analysts trust and understand model predictions[11].

Evaluation metrics

We have evaluated both of our experimental and proposed models based on as follows.

Accuracy (A)

Accuracy is a way to quantify the ratio of properly identified occurrences. Accuracy is calculated as Eq. (21):

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

where TN refers to the true negatives' number, FP refers to the false positives' number and FN refers to the false negatives' number.

Precision (P)
Precision is an indicator of how correct positive predictions are, indicating the proportion of projected positive cases that turn out to be positive. Precision is defined by Eq. (22):

$$P = \frac{TP}{TP + FP} \tag{22}$$

Recall (R)
Recall provides an indication of how well it can recognize each pertinent occurrence. Recall is calculated as Eq. (23):

$$R = \frac{TP}{TP + FN} \tag{23}$$

F1-Score (F1)
F1-score offers a balanced metric that takes both Recall (R) and Precision (P) into consideration. F1-score is very helpful when there is an imbalance in the class distribution as in our second dataset. F1-Score is defined by Eq. (24):

$$F1 = 2 \cdot \frac{\mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{24}$$

Area Under the Curve (AUC)
The AUC is a performance metric that evaluates the ability of a model to distinguish between classes. AUC is particularly useful for imbalanced datasets, as it measures the quality of a model's predictions across all classification thresholds. The AUC score is derived from the Receiver Operating Characteristic (ROC) curve, which plots the *TP* against the *FP*. AUC is calculated as Eq. (25):

$$AUC = \int_0^1 \mathrm{TP}(\mathrm{FP})\, d(\mathrm{FP}) \tag{25}$$

Moreover, for comprehensive evaluation the model's performance across all financial sentiment classes, the following metrics are calculated:

Macro Average P:

$$\text{Macro Average P} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{P}_i$$

Macro Average R:

$$\text{Macro Average R} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{R}_i$$

Macro Average F1:

$$\text{Macro Average F1} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{F1}_i$$

Macro Average AUC:

$$\text{Macro Average AUC} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AUC}_i$$

where N represents the total number of financial sentiment classes.

Results analysis

xFiTRNNs' financial sentence classification and explanation findings are shown in this section. When xFiTRNN is included into our classification model, exceptional results were obtained. Furthermore, the exceptional results that our explanation model achieves, which are heavily impacted by the performances of LIME and Anchors to this.

Financial sentence prediction

Table 1 illustrates the performance of various machine learning models across different embedding techniques applied to financial sentiment analysis datasets. A clear trend emerges as the sophistication of the embeddings increases, from traditional TF-IDF to advanced, domain-specific models like FinBERT. This progression

Embedding	Model	A	Macro Avg P	Macro Avg R	Macro Avg F1	Macro Avg AUC	Train time (s)	Infer time (s)
TF-IDF	AdaBoost	87.08	84.50	94.10	89.04	92.37	148.75	0.03
	GradientBoosting	85.51	81.22	96.31	88.12	90.58	273.45	0.02
	ExtraTrees	80.25	75.11	96.61	84.52	86.25	6.95	0.04
	XGBoost	79.26	74.15	96.46	83.85	85.73	87.19	0.03
	LDA	60.00	60.60	80.97	69.32	65.42	5.12	0.01
	QDA	65.76	70.53	66.37	68.39	70.16	10.25	0.02
	GradientBoosting +KNN+MLP	86.25	82.50	96.75	89.08	91.12	1532.85	46.18
	1 Hidden Layer NN	82.50	80.00	85.00	82.40	85.47	650.30	22.45
	2 Hidden Layers NN	83.20	81.00	86.00	83.40	86.64	804.15	25.12
	3 Hidden Layers NN	84.00	82.00	87.00	84.40	87.28	954.60	27.48
	BiGRU + 3 Hidden Layers NN	85.50	83.50	88.50	85.90	88.55	2003.75	100.25
	BiGRU + CNN	86.00	84.00	89.00	86.50	89.19	2502.30	120.15
Word2Vec	AdaBoost	88.73	87.12	95.83	91.28	93.14	149.85	0.02
	GradientBoosting	88.19	85.47	96.68	90.74	92.68	282.15	0.08
	ExtraTrees	83.87	79.58	97.32	87.53	89.37	6.98	0.05
	XGBoost	82.94	78.26	96.85	86.60	88.54	88.63	0.03
	LDA	58.43	59.67	78.91	68.01	64.78	5.05	0.04
	QDA	62.81	67.45	64.28	65.83	68.29	10.20	0.02
	GradientBoosting +KNN+MLP	88.70	85.10	97.10	90.68	92.82	1448.90	47.30
	1 Hidden Layer NN	84.10	81.60	86.40	83.96	87.18	704.85	28.32
	2 Hidden Layers NN	85.20	82.70	87.50	85.04	88.24	849.75	30.05
	3 Hidden Layers NN	86.30	83.80	88.60	86.12	89.36	1002.40	33.35
	BiGRU + 3 Hidden Layers NN	87.70	85.20	90.00	87.54	90.48	2203.50	110.10
	BiGRU + CNN	88.20	85.70	90.50	88.02	90.97	2701.25	130.20
GloVe	AdaBoost	89.16	87.64	96.12	91.70	93.56	149.90	0.02
	GradientBoosting	88.89	85.83	96.97	91.03	93.08	282.20	0.08
	ExtraTrees	84.27	79.96	97.56	87.98	89.64	7.05	0.05
	XGBoost	83.38	78.68	97.10	86.96	88.76	88.65	0.03
	LDA	58.74	60.11	79.27	68.40	64.93	5.10	0.04
	QDA	63.17	67.84	64.62	66.19	68.43	10.15	0.02
	GradientBoosting +KNN+MLP	89.25	85.55	97.40	91.00	93.23	1448.95	47.33
	1 Hidden Layer NN	84.60	82.10	86.90	84.46	87.62	704.80	28.31
	2 Hidden Layers NN	85.70	83.20	88.00	85.54	88.74	850.25	30.10
	3 Hidden Layers NN	86.80	84.30	89.10	86.62	89.85	1003.60	33.40
	BiGRU + 3 Hidden Layers NN	88.20	85.70	90.50	88.02	90.97	2204.75	110.15
	BiGRU + CNN	88.70	86.20	91.00	88.50	91.46	2702.50	130.25
FastText	AdaBoost	89.68	88.15	96.42	91.98	94.12	149.95	0.02
	GradientBoosting	89.27	86.31	97.21	91.43	93.64	282.25	0.08
	ExtraTrees	84.82	80.37	97.78	88.19	90.11	7.10	0.05
	XGBoost	83.91	79.10	97.35	87.17	89.24	88.70	0.03
	LDA	59.21	60.58	79.64	68.84	65.31	5.15	0.04
	QDA	63.62	68.23	64.95	66.54	69.12	10.20	0.02
	GradientBoosting +KNN+MLP	89.80	86.00	97.70	91.35	93.78	1449.05	47.35
	1 Hidden Layer NN	85.20	82.70	87.50	85.04	88.29	704.85	28.33
	2 Hidden Layers NN	86.30	83.80	88.60	86.12	89.41	850.30	30.12
	3 Hidden Layers NN	87.40	84.90	89.70	87.20	90.53	1004.75	33.42
	BiGRU + 3 Hidden Layers NN	88.80	86.30	91.10	88.60	91.65	2205.25	110.20
	BiGRU + CNN	89.30	86.80	91.60	89.08	92.14	2703.75	130.30

Continued

Embedding	Model	A	Macro Avg P	Macro Avg R	Macro Avg F1	Macro Avg AUC	Train time (s)	Infer time (s)
MiniLM	AdaBoost	90.23	88.68	96.75	92.18	94.59	150.25	0.02
	GradientBoosting	89.84	86.75	97.50	91.89	94.12	290.45	0.11
	ExtraTrees	85.37	80.78	98.01	88.38	90.67	7.55	0.06
	XGBoost	84.46	79.52	97.65	87.53	89.78	89.95	0.04
	LDA	59.68	61.05	79.98	69.20	65.74	5.55	0.05
	QDA	64.09	68.61	65.28	66.90	69.41	10.65	0.03
	GradientBoosting +KNN+MLP	90.35	86.45	98.00	91.70	94.25	1602.85	50.15
	1 Hidden Layer NN	86.30	83.80	88.60	86.12	89.52	850.35	25.10
	2 Hidden Layers NN	87.40	84.90	89.70	87.20	90.64	1001.25	30.05
	3 Hidden Layers NN	88.50	86.00	90.80	88.28	91.76	1201.75	35.12
	BiGRU + 3 Hidden Layers NN	89.90	87.40	92.20	89.68	92.88	2502.50	140.25
	BiGRU + CNN	90.40	87.90	92.70	90.16	93.37	3001.85	160.30
DistilBERT	AdaBoost	90.78	89.21	97.08	92.53	95.03	150.30	0.02
	GradientBoosting	90.32	87.18	97.83	92.22	94.57	291.20	0.11
	ExtraTrees	85.91	81.19	98.24	88.58	91.23	7.60	0.08
	XGBoost	84.99	79.94	97.89	87.97	90.35	89.05	0.04
	LDA	60.15	61.52	80.32	69.57	66.17	5.60	0.07
	QDA	64.54	69.00	65.61	67.27	70.29	10.70	0.05
	GradientBoosting +KNN+MLP	90.90	86.90	98.30	92.05	94.69	1623.95	65.30
	1 Hidden Layer NN	87.40	84.90	89.70	87.20	90.76	945.60	35.65
	2 Hidden Layers NN	88.50	86.00	90.80	88.28	91.88	1102.45	40.10
	3 Hidden Layers NN	89.60	87.10	91.90	89.36	93.00	1301.25	45.50
	BiGRU + 3 Hidden Layers NN	91.00	88.50	93.30	90.80	94.12	3002.75	170.35
	BiGRU + CNN	91.50	89.00	93.80	91.28	94.61	3503.20	190.40
FinBERT	AdaBoost	91.32	89.74	97.41	92.87	95.81	150.35	0.02
	GradientBoosting	90.79	87.60	98.14	92.72	95.34	300.75	0.12
	ExtraTrees	86.44	81.59	98.46	88.79	92.03	8.05	0.09
	XGBoost	85.53	80.36	98.12	88.26	91.14	89.10	0.05
	LDA	60.62	61.99	80.65	69.94	66.85	6.05	0.08
	QDA	64.98	69.38	65.94	67.61	70.63	11.10	0.06
	GradientBoosting +KNN+MLP	91.45	87.35	98.60	92.40	95.46	1702.65	70.25
	1 Hidden Layer NN	88.50	86.00	90.80	88.28	92.04	1002.50	40.15
	2 Hidden Layers NN	89.60	87.10	91.90	89.36	93.16	1203.75	45.20
	3 Hidden Layers NN	90.70	88.20	93.00	90.44	94.28	1403.90	50.10
	BiGRU + 3 Hidden Layers NN	92.10	89.60	94.40	91.84	95.40	3504.25	200.45
	BiGRU + CNN	92.60	90.10	94.90	92.32	95.89	4004.80	220.55
	Proposed model (xFiTRNN)	95.86	95.58	95.63	95.73	96.83	3545.25	154.15

Table 1. 5-fold cross-validated mean metrics values of tested models.

consistently enhances the models' performance metrics, including Accuracy, Precision, Recall, F1-Score and AUC. Significantly, the proposed xFiTRNN model stands out with an impressive 95.86% Accuracy and 96.83% AUC, pointing on the potential of optimal performance, beyond even the SOTA advanced transformer type, FinBERT. A detail analysis have presented following with possible reasoning of the outcome:

Starting with TF-IDF, a bag-of-words approach, models exhibit moderate performance with Accuracy ranging from 60.00% (LDA) to 87.08% (AdaBoost). While TF-IDF effectively captures term importance, it lacks semantic understanding, limiting its efficacy in nuanced financial texts. Transitioning to Word2Vec and GloVe, which provide semantic embeddings, models show noticeable improvements. For instance, AdaBoost's Accuracy increases to 88.73% with Word2Vec and further to 89.16% with GloVe. These embeddings capture contextual relationships between words, enhancing the models' ability to interpret sentiment more accurately.

FastText introduces sub-word information, allowing models to handle rare and morphologically complex words better. This is evident as models like GradientBoosting+KNN+MLP achieve an Accuracy of 89.80% and BiGRU + CNN reaches 89.30%, surpassing their counterparts using Word2Vec and GloVe. Moving to transformer-based embeddings, MiniLM and DistilBERT further elevate performance. MiniLM models

achieve up to 90.40% Accuracy, while DistilBERT pushes this to 90.78%, benefiting from deeper contextual understanding and bidirectional context modeling inherent to transformer architectures.

The most substantial improvements are observed with FinBERT, a domain-specific variant of BERT tailored for financial texts. Models leveraging FinBERT embeddings achieve the highest metrics, with AdaBoost reaching an Accuracy of 91.32% and the GradientBoosting+KNN+MLP ensemble model attaining 91.45%. FinBERT's specialized training on financial corpora enables it to grasp intricate financial jargon, idiomatic expressions and contextual nuances, thereby significantly enhancing sentiment detection capabilities. This specialization is reflected in higher Precision, Recall, F1-Scores and AUC values across all models, underscoring the advantage of using domain-specific embeddings in specialized tasks.

Furthermore, the complexity of the neural network architectures plays a crucial role in performance. Simple models like the Single Hidden Layer NN show incremental improvements with advanced embeddings, while more complex architectures such as BiGRU + CNN and GradientBoosting+KNN+MLP consistently outperform their simpler counterparts. For instance, BiGRU + CNN models achieve up to 92.60% Accuracy and 95.89% AUC with FinBERT embeddings, highlighting their ability to capture sequential and spatial features effectively. Moreover, the proposed model, xFiTRNN, stands out by achieving the highest performance metrics across all evaluation criteria, including an accuracy of 95.86%, precision of 95.58%, recall of 95.63%, F1-Score of 95.73% and an AUC of 96.83%. The remarkable results of the xFiTRNN model highlight the effectiveness of its hybrid design, which combines transformer-based processes, attention mechanisms and BiGRU networks to efficiently capture subtle sentiment patterns. According to the research, although pre-trained transformer models like as BERT and FinBERT provide comparable performance, sentiment analysis specific topologies like xFiTRNN can significantly increase performance.

Ablation test

Table 2's ablation analysis of the xFiTRNN model shows how important components gradually affect computing efficiency and performance. The model attains moderate metrics (82.45% accuracy, 81.50% F1) with the lowest training (≈1110s) and inference (≈106s) durations when starting with the basic configuration (FinBERT without BiGRU, Self-Attention (SA), and three hidden layers). Even though it takes twice as long to train (about 2146 s), adding three hidden layers alone increases accuracy to 85.60% and F1 to 84.30%, highlighting their function in feature refining. BiGRU's contextual modeling at computational cost is shown by the fact that adding BiGRU and three hidden layers (without SA) substantially enhances performance (90.75% accuracy, 90.12% F1) but considerably lengthens training time (≈3401s). In comparison to the BiGRU-3 Hidden Layers configuration, performance is somewhat worse when BiGRU and SA are retained and three Hidden Layers are removed (89.20% accuracy, 88.65% F1), indicating that Hidden Layers are essential for task-specific learning. Peak metrics (93.40% accuracy, 93.00% F1, 94.20% AUC) are achieved at moderate training (≈2255s) and inference (≈134s) times with the optimal configuration (GRU instead of BiGRU, with SA and 3 Hidden Layers). This suggests that SA makes up for the drawbacks of unidirectional GRU, allowing for effective, high-performance modeling. This illustrates a balance in which BiGRU's bidirectional context is less crucial when paired with SA, but SA and Hidden Layers improve generalization.

Resilience test

Table 3 points on the resilience of the xFiTRNN model on both Financial Phrasebank and IMBSEntFiN datasets. Advanced embeddings and more complex neural network architectures consistently yield superior metrics, with the proposed model xFiTRNN exemplifying the pinnacle of this synergy. Additionally, the superior performance on the Financial Phrasebank highlights the importance of dataset characteristics in achieving optimal model outcomes. A clear trend emerges when examining the impact of model complexity on performance. Starting with the 1 Hidden Layer NN, which achieves an accuracy of 88.50% and a macro average F1-score of 88.28% on the Financial Phrasebank, there is a noticeable improvement as the number of hidden layers increases. The 2 Hidden Layers NN boosts accuracy to 89.60% and the F1-score to 89.36%, while the 3 Hidden Layers NN further elevates these metrics to 90.70% and 90.44%, respectively. This incremental enhancement underscores the benefits of deeper neural network architectures in capturing complex patterns within the data. Advanced architectures, such as BiGRU + 3 Hidden Layers NN and BiGRU + CNN, significantly elevate performance

Embedding	Model	A	Macro avg P	Macro avg R	Macro avg F1	Macro avg AUC	Train time (s)	Infer time (s)
FinBERT	(**-BiGRU, -SA, -3 Hidden Layers NN**)	82.45	81.30	81.75	81.50	83.20	1109.54	105.84
	+3 Hidden Layers NN (**-BiGRU, -SA**)	85.60	84.10	84.50	84.30	86.00	2145.74	135.48
	+BiGRU, +3 Hidden Layers NN (**-SA**)	90.75	90.00	90.25	90.12	91.50	3400.52	158.35
	+BiGRU, +SA (**-3 Hidden Layers NN**)	89.20	88.50	88.80	88.65	90.10	2100.34	130.51
	+GRU, +SA, +3 Hidden Layers NN (**-BiGRU**)	93.40	92.80	93.10	93.00	94.20	2254.93	134.44

Table 2. Ablation test of the xFiTRNN model; text highlighted with italicizedrepresents inclusion whereas bolded represents exclusion of that specific component.

Dataset	Model	A	Macro Avg P	Macro Avg R	Macro Avg F1	Macro avg AUC	Train time (s)	Infer time (s)
Financial Phrasebank	1 Hidden Layer NN	89.30	86.50	92.00	89.00	93.04	1494.30	90.25
	2 Hidden Layers NN	90.40	87.70	92.90	90.16	94.16	2000.75	119.15
	3 Hidden Layers NN	91.80	88.90	94.30	91.24	95.28	2464.20	101.30
	BiGRU + 3 Hidden Layers NN	93.40	90.60	95.60	92.74	96.60	3000.85	110.45
	BiGRU + CNN	94.00	91.10	96.20	93.32	96.99	3500.35	130.20
	Proposed model (xFiTRNN)	97.61	97.33	97.38	97.48	98.58	3857.21	165.41
IMBSEntFiN	1 Hidden Layer NN	87.70	85.50	89.60	87.58	91.04	870.50	58.30
	2 Hidden Layers NN	88.80	86.50	90.90	88.56	92.16	1166.30	53.15
	3 Hidden Layers NN	89.60	87.50	91.70	89.64	93.28	1400.75	46.80
	BiGRU + 3 Hidden Layers NN	90.80	88.60	93.20	90.94	94.20	2200.70	140.48
	BiGRU + CNN	91.20	89.10	93.60	91.32	94.79	2502.40	120.10
	Proposed model (xFiTRNN)	94.11	93.83	93.88	93.98	95.08	2200.00	94.25

Table 3. Resilience of the xFiTRNN model on both financial phrasebank and IMBSEntFiN datasets.

Embedding	Model	t-value (A)	p-value (A)	t-value (P)	p-value (P)	t-value (R)	p-value (R)	t-value (F1)	p-value (F1)
FinBERT	1 Hidden Layer NN	42.57	3.24e−11	30.14	5.53e−10	38.79	2.13e−11	35.23	9.83e−11
	2 Hidden Layers NN	47.83	1.52e−12	34.92	2.34e−11	45.31	3.41e−12	40.15	7.63e−12
	3 Hidden Layers NN	54.27	4.83e−14	40.56	6.27e−13	60.81	1.24e−14	50.34	3.14e−13
	BiGRU + 3 Hidden Layers NN	58.34	2.73e−15	45.63	1.42e−14	65.44	8.93e−16	55.85	4.53e−15
	BiGRU + CNN	62.13	9.13e−16	50.28	3.24e−15	70.39	2.41e−17	60.57	1.82e−16

Table 4. 5-fold cross-validated paired t-tests comparing macro-average precision, recall, F1 scores, and accuracy of FinBERT-based models against the xFiTRNN model.

metrics. The BiGRU + 3 Hidden Layers NN achieves an impressive accuracy of 92.10% and an F1-score of 91.84% on the Financial Phrasebank, indicating superior capability in understanding sequential dependencies and contextual nuances inherent in financial texts. The BiGRU + CNN model further advances these results, attaining an accuracy of 92.60% and an F1-score of 92.32%, highlighting the synergistic effects of combining recurrent and convolutional layers to enhance feature extraction and classification performance. The proposed model xFiTRNN stands out as the pinnacle of performance, achieving an outstanding accuracy of 95.86% and an F1-score of 95.73% on the Financial Phrasebank. These metrics not only surpass all other models but also reflect the model's robustness and effectiveness in accurately classifying sentiments within financial texts. The high AUC value of 96.83% further corroborates the model's exceptional discriminative ability, ensuring reliable differentiation between sentiment classes. Though the IMBSEntFiN dataset also benefits from increased model complexity, its performance metrics are consistently lower than those of the Financial Phrasebank. For instance, the proposed model xFiTRNN achieves an accuracy of 94.11% and an F1-score of 93.98% on IMBSEntFiN, which, although impressive, remain below the corresponding values for the Financial Phrasebank. This disparity suggests that the Financial Phrasebank may possess characteristics—such as clearer sentiment indicators or more balanced class distributions—that facilitate higher model performance. The analysis of AUC values across both datasets reinforces these observations. Higher AUC scores in the Financial Phrasebank indicate superior model performance in distinguishing between classes, whereas slightly lower AUC values in IMBSEntFiN suggest a marginally reduced ability to discriminate sentiments effectively. Nevertheless, all models maintain robust AUC values, reflecting their overall efficacy in handling sentiment classification tasks within financial contexts.

Statistical significance test

The 5-fold cross-validated paired t-test results in Table 4 demonstrate the statistical superiority of the xFiTRNN model over FinBERT-based variants across all metrics (accuracy, precision, recall, F1). All configurations exhibit extremely low p-values ($<$ 3.24e−10 for accuracy, $<$ 3.24e−15 for F1), decisively rejecting the null hypothesis (H_0) and confirming significant performance differences. The t-values increase with model complexity: simpler architectures (e.g., 1 Hidden Layer NN: t = 42.57 for accuracy, p = 3.24e−11) show smaller effects compared to BiGRU-enhanced models (BiGRU+CNN: t = 62.13 for accuracy, p = 9.13e−16), highlighting the compounding benefits of bidirectional context and convolutional layers. The highest t-values (e.g., 70.39 for recall, p = 2.41e−17) and correspondingly minimal p-values for BiGRU+CNN underscore its robustness in capturing intricate patterns, though still falling short of xFiTRNN's performance. These results validate xFiTRNN's design,

Attack type	Success rate	F1 drop
Glyph substitution ("growth"→"groWth")	12.71%	3.23%
Financial negation insertion ("strong results"→"not strong results")	18.44%	5.18%
SEC filing obfuscation (Adding legalese)	9.36%	2.15%

Table 5. xFiTRNNs' adversarial robustness on financial phrasebank.

Market	xFiTRNN	FinBERT
Cryptocurrency (CoinNews)	88.45%	82.16%
Pharmaceuticals (DrugFDA Corpus)	85.76%	79.31%
Japanese socks (Nikkei Earnings Calls)	83.24%	71.59%

Table 6. xFiTRNNs' cross-market notion score (Macro F1).

Model	Dataset	A	Macro average P	Macro average R	Macro average F1	Macro average AUC
MLP/BPA[19]	SST-2, SST-5	81.38	81.37	81.46	81.39	81.38
LR[60]	DGAP	85.40	85.12	85.80	85.49	86.35
SMO+DT[16]	Airline Twitter	89.47	91.60	89.50	96.30	91.57
CNN[61]	StockTwits	90.93	91.68	90.04	90.86	92.46
BERT[15]	DJI, Financial News	82.50	75.00	71.30	72.50	80.35
GPT-P4[14]	Forex Pair	76.50	77.20	76.50	76.30	78.56
BART-large[2]	Financial Phrasebank, SemEval-2017 Task 5	94.70	95.00	94.50	94.70	95.37
FinBERT[4]	Financial Phrasebank	88.20	87.20	88.50	87.80	89.38
Proposed model (xFiTRNN)	Financial Phrasebank	97.61	97.33	97.38	97.48	98.58

Table 7. Comparison of evaluation metrics of various NLP algorithms with xFiTRNN. The results point out that xFiTRNN exceeds previous outcomes.

emphasizing that architectural enhancements (e.g., BiGRU, CNN) synergistically improve generalization while maintaining statistical rigor, with all comparisons surviving Bonferroni correction ($\alpha = 0.05$).

Adversarial robustness tests

We tested xFiTRNN against three financial adversarial attack types using the TextAttack framework[59] shown in Table 5. xFiTRNN outperformed FinBERT in resilience, showing 37% lower success rates against character-level attacks due to its Unicode normalization layer[9]. The model maintained 94.26% original accuracy under combined attacks versus 89.79% for FinBERT.

Cross-market notion tests

We evaluated out-of-sample performance on three unseen markets illustrated in Table 6. Notably, xFiTRNN used Japanese FinBERT embeddings without retraining. The 11.65% improvement in multilingual performance demonstrates effective transfer learning capabilities.

Affectability on previous results

Table 7 presents a comparative analysis of various NLP models for financial sentiment analysis, highlighting the performance of our model, xFiTRNN, against several benchmark algorithms. The evaluation metrics demonstrate that xFiTRNN significantly outperforms other models across all key metrics. Specifically, xFiTRNN achieves an impressive accuracy, precision, recall and F1-score, surpassing the closest competitor, BART-large[2]. These results underscore the efficacy of xFiTRNN in accurately capturing financial sentiments and suggest that its advanced feature extraction capabilities and domain-specific adaptations provide superior performance compared to existing models.

Computational efficiency vs. performance trade-offs

While xFiTRNN requires 3,545s training time (Table 1), our lite variant achieves 91.1% F1 in 64s inference latency - suitable for HFT systems. The 8.3x speedup comes from dynamic attention pruning and 8-bit quantization[62].

Explainability tests

Results

Table 8 presents LIME interpretations of four test cases from the xFiTRNN model. Each test case shows the sentence being analyzed, the probabilities assigned to each sentiment class (negative, positive, neutral), the count of features, the highlighted words and their corresponding weights. In the first test case, the model predicts the sentence with a 42% probability of being neutral, influenced mainly by words like "EUR," "Operating," and "rose," which show slight weight variations. The second test case, with a positive probability of 42%, highlights words "Market," "share," and "decreased," indicating these words contribute minimally to the model's positive classification. The third case shows a balanced distribution among the classes, with the positive class at 40% and words like "Finnish," "contract," and "won" contributing to this outcome. Whereas, the fourth test case, with a 39% probability of being positive, is influenced by words like "EUR," "profit," and "rose," each having a small but significant impact on the sentiment classification. These interpretations illustrate the model's sensitivity to specific words in determining sentiment, providing insights into the decision-making process of the classifier.

Table 9 presents the performance of xFiTRNN through four test cases, detailing the model's precision and the key anchors influencing classification decisions. In Test 1, a misclassified positive statement as neutral, with a precision of 0.94, was influenced by anchors "result," "rose," "EUR," "146mn," "loss," and "267mn." Test 2 showcases a negative statement misclassified as positive, achieving a precision of 0.96, with anchors including "decreased," "0.1," "points," and "24.8%." Test 3 demonstrates a correctly classified positive statement with a precision of 1.00, highlighted by the anchor words "won" and "order." Finally, Test 4 features another correctly classified positive statement, marked by anchors like "profit," "rose," "EUR 27.8," and "EUR 17.5," and a precision of 0.98. These results underscore xFiTRNNs' strong precision in correctly classifying sentiment, while also highlighting areas where misclassifications occur, particularly in distinguishing nuanced sentiment changes.

This analysis offers a balanced view of the model's strengths and weaknesses, ensuring that the explainability assessment does not overlook any critical aspects of its performance. Analyzing the classification results of negative, positive and neutral sentences significantly enhances the model's explainability. For correctly classified positive statements, the analysis identifies specific keywords and feature combinations, such as "won," "order,"

Test	Sentence	*P*(negative)	*P*(positive)	*P*(neutral)	Features count	Highlighted words	Weights
1	"Operating result including non-recurring items rose to EUR 146 mn from a loss of EUR 267 mn in 2009."	0.27	0.31	0.42	10	1. EUR 2. Operating 3. including 4. rose 5. to 6. result 7. 146mn 8. non 9. in 10. 2009	+ 0.02 − 0.02 − 0.01 − 0.01 − 0.01 + 0.00 − 0.00 − 0.00 − 0.00 + 0.00
2	"Market share decreased on the route between Helsinki in Finland and Tallinn in Estonia by 0.1 percentage points to 24.8%."	0.33	0.42	0.24	10	1. Market 2. share 3. Estonia 4. route 5. decreased 6. Tallinn 7. points 8. 8 9. Finland 10. in	+ 0.02 + 0.01 + 0.01 + 0.01 + 0.01 + 0.01 + 0.00 + 0.00 − 0.00 − 0.00
3	"Finnish software developer Basware Oyj said on November 30 , 2006 its U.S. subsidiary Basware, Inc. won an order to provide software for contract lifecycle management to an unnamed U.S. medical technology company."	0.31	0.40	0.29	10	1. Finnish 2. an 3. contract 4. technology 5. said 6. Basware 7. won 8. to 9. 2006 10. on	+ 0.01 + 0.01 + 0.01 + 0.01 − 0.01 + 0.01 + 0.01 − 0.00 − 0.00 − 0.00
4	"Operating profit rose to EUR 27.8 mn from EUR 17.5 mn in 2008."	0.34	0.39	0.27	10	1. EUR 2. from 3. profit 4. in 5. 27 6. 17 7. to 8. rose 9. 2008 10. 5	+ 0.02 − 0.02 + 0.01 − 0.01 + 0.01 + 0.01 − 0.01 − 0.01 − 0.01 + 0.00

Table 8. xFiTRNNs' LIME interpretations of four test cases.

Test	Sentence	Scenario	Precision	Anchors
1	"Operating result including non-recurring items rose to EUR 146 mn from a loss of EUR 267 mn in 2009."	Misclassified positive statement as neutral	0.94	Result AND rose AND EUR AND 146 mn AND loss AND EUR AND 267 mn
2	"Market share decreased on the route between Helsinki in Finland and Tallinn in Estonia by 0.1 percentage points to 24.8%."	Misclassified negative statement as positive	0.96	Decreased AND 0.1 AND points AND 24.8%
3	"Finnish software developer Basware Oyj said on November 30, 2006 its U.S. subsidiary Basware, Inc. won an order to provide software for contract lifecycle management to an unnamed U.S. medical technology company."	Correctly classified positive statement	1.00	Won AND order
4	"Operating profit rose to EUR 27.8 mn from EUR 17.5 mn in 2008."	Correctly classified positive statement	0.98	Profit AND rose AND EUR 27.8 AND EUR 17.5

Table 9. xFiTRNNs' anchors four test cases.

"profit," and "rose," which the model consistently associates with positive sentiment. This consistency not only builds trust in the model's decision-making process for similar inputs but also highlights the features that effectively drive accurate predictions. Conversely, examining misclassified instances reveals where the model struggles, where terms like "decreased" and specific numerical data led to an incorrect positive classification. Reasoning of these misinterpretations allows to pinpoint areas needing improvement, such as better handling of numerical indicators or more nuanced contextual understanding. Furthermore, the inclusion of diverse scenarios facilitates targeted enhancements in feature engineering and model training. By observing which features contribute to both correct and incorrect classifications, we can refine the feature selection process to mitigate the undue influence of certain terms or enhance the representation of contextually significant words. This targeted refinement ensures that the model becomes more adept at discerning sentiment, in complex or nuanced contexts, financial statements where numerical data can be particularly challenging to interpret accurately. Additionally, utilizing explainability tools, LIME and Anchors across these varied scenarios provides transparent insights into the model's decision-making processes. LIME highlights individual feature contributions, while Anchors reveal the sufficient conditions that drive specific predictions. This dual approach not only elucidates how the model arrives at its conclusions but also helps in identifying consistent patterns and potential biases in its interpretations. Such transparency is crucial for building trust among users and stakeholders, as it allows them to understand and validate the model's behavior comprehensively.

Additional tests

This further explainability analysis incorporates fidelity (mean = 0.89) and stability (82%) for LIME, alongside coverage metrics for Anchors. Human evaluation revealed 85% relevance for LIME explanations and 78% intuitiveness for Anchors. Compared to SHAP and IG, LIME and Anchors balance interpretability and computational efficiency but lag in stability and granularity. For instance, SHAP's game-theoretic approach better captures contextual interactions (e.g., "contract lifecycle" in Test 3), while IG's gradient-based method offers finer attribution for numerical phrases like "27.8 mn." The following parts presents the concise details of these further explainability tests with their respective outcomes.

Performance evaluation of explainability methods

We assess the explainability methods using three key metrics:

- **Fidelity**: Measures how accurately the explanation approximates the model's behavior locally. For LIME, fidelity is calculated as the R^2 score between the surrogate model and the original model's predictions. For SHAP and IG, we use the correlation between feature attributions and model outputs.
- **Stability**: Evaluates the consistency of explanations across multiple runs or perturbations. We use the Jaccard similarity index for the top-k features ($k = 5$) across 10 runs.
- **Coverage (for Anchors)**: Reports the precision and coverage of the generated rules, where precision indicates the accuracy of the rule in predicting the correct class, and coverage reflects the proportion of instances to which the rule applies.

The performance of each method is summarized in Table 10, based on the xFiTRNN model evaluated on the Financial Phrasebank dataset.

Method	Fidelity (R^2)	Stability (Jaccard)	Coverage	Precision
LIME	0.89	0.82	–	–
Anchors	–	–	0.15	0.96
SHAP	0.92	0.91	–	–
IG	0.88	0.85	–	–

Table 10. Performance evaluation of explainability methods for xFiTRNN on the Financial Phrasebank dataset. Fidelity and stability are reported for LIME, SHAP, and IG, while coverage and precision are specific to Anchors.

Method	Relevance (%)	Intuitiveness (%)	Trust (1–5)
LIME	85	78	3.8
Anchors	82	85	4.1
SHAP	88	72	3.9
IG	80	65	3.5

Table 11. Human evaluation results for explainability methods. Relevance and intuitiveness are percentages; user trust is on a 1–5 scale.

- **LIME**: Achieves a fidelity of 0.89, indicating a strong local approximation, but its stability is lower (0.82) due to variability in feature importance from perturbation sampling.
- **Anchors**: Offers high-precision rules (0.96) but limited coverage (0.15), applying primarily to instances with clear sentiment indicators.
- **SHAP**: Outperforms others in fidelity (0.92) and stability (0.91), leveraging its game-theoretic approach for consistent feature interaction capture.
- **IG**: Shows comparable fidelity (0.88) but slightly lower stability (0.85), as gradient-based attributions can vary with model architecture and input scaling.

SHAP's superior performance comes at a computational cost, averaging 5.2 s per explanation, compared to LIME's 2.1 s and Anchors' 3.8 s—a critical trade-off for real-time financial applications.

Human evaluation study
To evaluate practical utility, we conducted a study with 10 financial domain experts. Participants reviewed predictions and explanations from LIME, Anchors, SHAP, and IG for 20 randomly selected test cases from the Financial Phrasebank dataset. They rated each explanation on:

1. **Relevance**: Alignment of highlighted features or rules with expert judgment of sentiment drivers.
2. **Intuitiveness**: Ease of understanding and applicability in decision-making.

We also measured user trust on a scale from 1 (low) to 5 (high), reflecting confidence in predictions post-explanation. Results are shown in Table 11.

- **LIME**: High relevance (85%) due to intuitive feature highlights (e.g., "profit"), but lower intuitiveness (78%) from abstract weights.
- **Anchors**: Top intuitiveness (85%) and trust (4.1) thanks to clear rules (e.g., "IF 'won' AND 'order' THEN positive"), though relevance dips (82%) for oversimplified cases.
- **SHAP**: Highest relevance (88%) but lower intuitiveness (72%), as SHAP values are less actionable for non-experts.
- **IG**: Lowest in relevance (80%) and intuitiveness (65%), with gradient attributions less interpretable for financial text.

Inter-rater reliability (Fleiss' κ) was 0.72 for relevance and 0.68 for intuitiveness, indicating moderate expert agreement.

Analysis and trade-offs
The evaluation reveals distinct trade-offs:

- **LIME**: Balances fidelity and efficiency, ideal for real-time use, but lower stability (0.82) risks inconsistent explanations.
- **Anchors**: Excels in intuitiveness and trust, meeting transparency needs (e.g., FINRA guidelines[63]), but low coverage (0.15) limits applicability.
- **SHAP**: Best in fidelity and stability, capturing complex interactions, yet its computational cost (5.2s) hinders real-time use.

- **IG**: Offers granular insights for numerical features, but lower relevance and intuitiveness suit it for debugging over end-user explanations.

The LIME–Anchors framework balances detail and usability, enhancing trust (3.8 and 4.1) in financial contexts where transparency and efficiency are paramount.

Computational cost vs. interpretability trade-offs
The xFiTRNN model, while achieving state-of-the-art performance in financial sentiment analysis, is resource-intensive due to its hybrid transformer-BiGRU architecture and dual explainability framework. This section rigorously examines the trade-offs between the computational costs of xFiTRNN and its interpretability benefits, comparing it to simpler models that may offer practical advantages in resource-constrained environments. We also explore scenarios where lower-complexity models might be preferable, even if they compromise slightly on performance or explainability.

Computational costs of xFiTRNN The xFiTRNN model incurs significant computational overhead due to its multi-component design:

- **Training time**: With 3,545 s (approximately 59 min) per epoch on a single NVIDIA A100 GPU, xFiTRNN's training is time-intensive, especially when fine-tuning on large financial datasets.
- **Inference latency**: The model's inference time averages 154.15 s for a batch of 16 sentences, translating to approximately 9.6 s per sentence. This latency is prohibitive for real-time applications like high-frequency trading (HFT).
- **Memory usage**: The model requires 12.3 GB of GPU memory during training and 4.8 GB during inference, limiting its deployment on edge devices or systems with constrained resources.

In contrast, simpler models such as logistic regression (LR) with TF-IDF features or a single-layer RNN exhibit significantly lower computational demands:

- **Logistic Regression**: Trains in under 10 s, with inference times of milliseconds and negligible memory usage.
- **Single-layer RNN**: Trains in approximately 300 s per epoch, with inference times around 0.5 s per sentence and 2.1 GB memory usage.

Interpretability gains of xFiTRNN Despite its computational demands, xFiTRNN provides substantial interpretability advantages through its LIME and Anchors framework:

- **Fidelity**: LIME achieves a fidelity score of 0.89, indicating strong local approximation of the model's behavior, while Anchors provide high-precision rules (0.96) for key predictions.
- **Stability**: LIME's stability (Jaccard similarity of 0.82) ensures consistent feature attributions, though it lags behind SHAP's 0.91.
- **Trust**: In human evaluations, xFiTRNN's explanations garnered a trust score of 4.1 (Anchors) and 3.8 (LIME), compared to 3.2 for logistic regression's feature coefficients.

Simpler models, while inherently more interpretable, often fall short in explanation quality:

- **Logistic Regression**: Offers global feature importance but lacks local context, resulting in lower fidelity (0.75) and relevance (70% in human evaluations).
- **Single-layer RNN with attention**: Provides basic attention weights but struggles with stability (0.65 Jaccard) and user trust (3.0), as attention mechanisms can be opaque in financial contexts.

Scenarios favoring simpler models Despite xFiTRNN's advantages, simpler models may be preferable in specific scenarios:

- **High-Frequency Trading (HFT)**: In HFT systems, where decisions must be made in milliseconds, the 9.6-second inference latency of xFiTRNN is impractical. A lightweight model like logistic regression, with sub-second inference, is essential, even if it sacrifices some accuracy (e.g., 85% vs. xFiTRNN's 95.86%).
- **Edge deployment**: For applications on resource-constrained devices (e.g., mobile trading apps), the memory and power requirements of xFiTRNN are prohibitive. A single-layer RNN, with 2.1 GB memory usage, offers a feasible alternative, albeit with reduced performance (e.g., 88% accuracy).
- **Regulatory compliance**: In jurisdictions with strict model transparency requirements (e.g., EU's AI Act), simpler models like logistic regression may be favored for their inherent interpretability, despite lower fidelity, as they align with regulatory mandates for auditable logic.

To illustrate these trade-offs, Table 12 compares xFiTRNN with simpler alternatives across computational cost, performance, and interpretability metrics.

Mitigating computational costs To address xFiTRNN's resource demands, we propose several optimization strategies:

- **Model distillation**: Distilling xFiTRNN into a smaller student model (e.g., a lightweight transformer) can reduce inference latency while retaining much of its performance and interpretability.
- **Quantization**: Applying 8-bit quantization reduces memory usage by 50%, enabling deployment on lower-end hardware without significant accuracy loss.

Model	Train time (s)	Infer time (s)	Memory (GB)	A	F1	Fidelity	Trust
xFiTRNN	3,545	9.6	12.3	95.86	95.73	0.89	4.1
Logistic regression	10	0.001	0.1	85.00	84.50	0.75	3.2
Single-Layer RNN	300	0.5	2.1	88.00	87.50	0.80	3.0

Table 12. Performance of xFiTRNN with simpler models across computational cost, performance, and interpretability metrics.

Hardware	Batch size	Precision	Latency (ms)	F1 (%)
NVIDIA A100 (Baseline)	16	FP32	154 ± 12	95.73
NVIDIA T4	16	FP16	218 ± 18	95.12
CPU Cluster (16 cores)	128	INT8	192 ± 21	93.88
Quantized lite model	64	INT8	64 ± 5	91.10

Table 13. Inference latency under hardware constraints.

Technique	Params	Latency (ms)	ΔF1	Use Case
Full Model (Baseline)	214M	154	–	Regulatory reporting
Pruned (40%)	89M	98	– 2.3	Real-time alerts
Distilled Student	67M	48	– 3.0	HFT signal processing

Table 14. Compression performance trade-offs.

- **Efficient attention**: Replacing standard self-attention with sparse or linear attention mechanisms (e.g., Linformer) can lower computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, improving scalability.

Model efficiency and deployment scalability

Latency analysis across hardware configurations

To evaluate xFiTRNN's operational feasibility, we benchmarked inference latency across hardware platforms (Table 13). While the full model achieves 95.86% accuracy on an NVIDIA A100 (154 ms latency), the 8-bit quantized variant reduces latency to 64ms with minimal performance loss (91.1% F1), enabling deployment in high-frequency trading (HFT) pipelines. On CPU clusters, model parallelism across 16 cores achieves sub-200ms latency for batch sizes $\leq$128, meeting real-time requirements for earnings call analysis.

Model compression techniques

We applied iterative magnitude pruning and knowledge distillation to balance accuracy and efficiency:

Structured pruning
Removing 40% of self-attention heads reduced model size by 58% while retaining 93.4% F1.

Distillation
A 4-layer student model trained on xFiTRNN logits achieved 92.8% F1 with 3.2× faster inference (Table 14).

High-Frequency Trading (HFT) simulation

In a simulated HFT environment processing 5000 sentences/sec:

- The quantized lite model sustained 98.7% throughput at peak load, outperforming FinBERT (84.2%) and BERT-base (71.5%).
- Dynamic attention pruning reduced GPU memory usage by 37% during market volatility spikes without accuracy degradation.

Multilingual robustness

While the xFiTRNN model exhibits robust performance on English-language financial texts, such as the Financial Phrasebank and IMBSEntFiN datasets, its ability to generalize across multilingual and cross-regional financial corpora warrants further exploration. Financial narratives differ significantly across languages and regions in terms of structure, sentiment expression, and domain-specific terminology. These variations may

impact both the model's predictive accuracy and its explainability. For example, a phrase like "strong growth" in English might translate directly to "starkes Wachstum" in German, yet culturally specific expressions or financial jargon—such as "bullish" versus "bearish" in Western markets or "dragon" and "phoenix" metaphors in Asian contexts—could introduce additional complexity. Moreover, phenomena like code-switching (e.g., blending English financial terms with local languages) and region-specific regulatory terminology (e.g., EU's MiFID II versus U.S. SEC regulations) further challenge the model's adaptability.

Additional evaluation plan for multilingual datasets

To rigorously assess xFiTRNN's multilingual robustness and generalizability, we propose the following comprehensive evaluation framework:

Multilingual benchmark datasets

- **Selection**: We will curate a diverse set of financial sentiment datasets spanning multiple languages, including the Japanese Financial Phrasebank (translated from English), Chinese Earnings Call Transcripts (e.g., from SSE-listed companies), and EU Regulatory Filings (e.g., in French, German, and Spanish). These datasets will encompass both translated and native-language financial texts to test linguistic and domain-specific adaptability.
- **Evaluation metrics**: In addition to standard metrics (accuracy, F1, AUC), we will measure multilingual transfer performance, focusing on zero-shot and few-shot learning capabilities. This will leverage multilingual embeddings, such as XLM-RoBERTa, fine-tuned on financial corpora.

Domain-specific terminology handling

- **Terminology mapping**: We will construct a multilingual financial lexicon to align sentiment-laden terms across languages (e.g., "dividend" in English, "Dividende" in German). This lexicon will enhance the model's ability to interpret financial concepts consistently.
- **Adversarial testing**: We will introduce adversarial examples by perturbing domain-specific terms (e.g., substituting "profit" with "gain" or its translation) to evaluate robustness. This will also assess whether explainability methods maintain consistent feature attributions across languages.

Explainability across languages

- **Multilingual attribution consistency**: We will examine whether explainability tools like LIME and Anchors yield coherent explanations for equivalent financial concepts across languages (e.g., "growth" in English versus "croissance" in French in parallel contexts).
- **Human evaluation with multilingual experts**: A study involving financial analysts fluent in multiple languages will evaluate the relevance and intuitiveness of model explanations, identifying potential language-specific biases or misattributions.

Impact of domain-specific terminology on sentiment prediction and explainability

Domain-specific terminology presents distinct challenges in multilingual settings:

- **Sentiment prediction**: Financial jargon often carries implicit sentiment that varies by region. For instance, "quantitative easing" might imply positive sentiment in expansionary policy contexts but neutrality or negativity amid inflation concerns. The model will require region-specific fine-tuning or multilingual knowledge distillation to capture these nuances.
- **Explainability**: Methods like LIME may falter with low-frequency, high-impact terms (e.g., "IPO" or "M&A") if their embeddings lack sufficient contextualization in the target language. Similarly, Anchors' rule-based approach may struggle to generalize across languages with divergent syntactic structures (e.g., English's subject-verb-object versus Japanese's subject-object-verb).

To address these challenges, we propose:

- **Multilingual pretraining with financial corpora**: Fine-tuning multilingual transformers (e.g., mBERT) on financial texts from diverse languages will bolster the model's grasp of domain-specific terminology.
- **Syntax-aware explainability**: Incorporating syntax-aware attribution methods (e.g., dependency-based feature importance) will account for linguistic structural differences, ensuring explanations remain contextually faithful.

Practical implementation plan for practitioners

The deployment of xFiTRNN in real-world financial systems requires careful alignment with operational workflows, regulatory frameworks, and ethical standards. Below, we outline actionable strategies for practitioners:

Risk mitigation strategies

Adversarial robustness protocols

- Implement real-time adversarial detection modules (e.g., unicode normalization layers) to counter glyph substitution attacks.

- Conduct quarterly stress tests using frameworks like TextAttack to evaluate model resilience against evolving attack vectors (see Table 5).
- Deploy ensemble-based fallback mechanisms where low-confidence predictions trigger human-in-the-loop validation.

Bias and concept drift monitoring

- Track feature attribution stability via KL divergence between training and production attention distributions.
- Integrate fairness-aware metrics (e.g., demographic parity difference) to detect biases in sector-specific predictions.

Step-by-step integration plan

Pilot phase

- **Domain adaptation**: Fine-tune xFiTRNN on institution-specific lexicons (e.g., earnings call transcripts, internal reports).
- **Workflow mapping**: Identify high-impact use cases (e.g., earnings sentiment alerts, ESG risk scoring) and align model outputs with analyst workflows.

Deployment phase

Hybrid human-AI pipelines:

- Flag predictions with $< 90\%$ confidence for manual review.
- Use Anchors rules to generate FINRA-compliant audit trails (see Table 9).

Latency optimization: For high-frequency trading systems, deploy the 8-bit quantized lite variant (91.1% F1, 64ms latency).

Post-deployment

- Establish weekly retraining cycles using analyst-corrected labels to address concept drift.
- Monitor SHAP interaction values to detect emerging feature correlations (e.g., "inflation" $\leftrightarrow$ "rate hikes").

Ethical and regulatory compliance

Transparency mandates

- Align LIME/Anchors explanations with MiFID II's "sufficiently granular" reporting requirements by mapping key features to predefined financial concepts (e.g., FIBO ontologies).

EU AI act compliance

- Classify xFiTRNN as a "high-risk" system under Article 6(2) due to its influence on investment decisions.
- Implement SR 11-7-guided model risk management, including third-party validation of explanation fidelity (see Table 10).

Bias audits

- Partner with auditing firms to evaluate fairness across demographic axes (e.g., regional market coverage).
- Disclose false positive/negative rates by asset class in quarterly transparency reports.

Discussion and future works

While the xFiTRNN model demonstrates robust performance on benchmark datasets (Financial Phrasebank and IMBSEntFiN), its generalizability to multilingual financial texts, region-specific corpora, or distinct regulatory contexts remains to be validated. The current study focuses on English-language datasets with Western financial terminology, which may limit direct applicability to emerging markets (e.g., Asian or African financial texts) or multilingual environments where code-switching and localized jargon are prevalent. For instance, the model's reliance on FinBERT—trained primarily on English financial documents—could hinder performance on texts incorporating non-English terms (e.g., Chinese stock codes or EU regulatory abbreviations). Future work should evaluate more multilingual transfer capabilities using aligned multilingual datasets (e.g., XLM-Fin) and assess robustness to domain shifts caused by regional reporting standards or niche financial instruments (see section "Multilingual robustness"). Additionally, incorporating adversarial testing with perturbed inputs (e.g., mixed-language earnings calls or region-specific slang) would further stress-test the model's interpretive flexibility. While the architecture's attention mechanisms theoretically enable dynamic adaptation to contextual nuances, explicit training on geographically diverse data and hierarchical domain adaptation layers could enhance its global applicability. Furthermore, our human evaluation study with 10 financial domain experts revealed critical insights about explanation quality. While 85% of LIME-highlighted terms like "profit surge" and "order win" were deemed relevant to sentiment judgments, experts identified two key limitations: (1) 22% of Anchors rules contained ambiguous conjunctions (e.g., "growth AND restructuring" without contextual weighting), and (2) explanation consistency dropped significantly for misclassifications (Fleiss' $\kappa = 0.72$ vs $\kappa = 0.84$ for correct predictions). Additionally, comparative studies with 23 CFA charterholders reveal both alignment and divergence

in explainability. While 87% of model-generated Anchors rules (e.g., "IF 'guidance raise' THEN positive") matched analyst logic, critical disagreements emerged in 62% of cases involving ratio context – human analysts rejected explanations omitting countervailing factors like "costs fell 10%" without revenue change annotations. This gap underscores the need for hybrid human-AI workflows, particularly given the model's 1.2s explanation speed versus the analyst average of 4.7 min per report. These results underscore the necessity of hybrid human-AI workflows rather than autonomous deployment. Practical implementation requires institutional guardrails which can integrate pre-trade checks that flag predictions with $< 90\%$ confidence or Anchors rules conflicting with Morningstar ratings[64], alongside FINRA-compliant audit trails of all model decisions[63]. Financial institutions adopting xFiTRNN should implement weekly retraining cycles using analyst-corrected labels and monitor concept drift via KL divergence between training/production attention distributions – requirements aligning with both SR 11-7 guidelines[65] and the EU AI Act's transparency mandates[66] (see section "Practical implementation plan for practitioners").

Conclusion

This paper evaluates various NLP models in the context of financial sentiment analysis, highlighting the superior performance of our proposed model xFiTRNN. Through a comparative study and detailed analysis, the research reveals the critical importance of selecting appropriate features and methodologies tailored to the specific dynamics of financial markets. Furthermore, findings reveal that, we have done comprehensive testing and adaptability in model deployment, especially in real-time market monitoring. We believe the research marks a significant contribution to the domain, offering a novel approach in sentiment analysis and paving the way for future advancements in the field. We wish that this study's outcomes enhance the understanding of sentiment analysis in finance and provide a valuable resource for decision-making in financial contexts.

Data availability

We have used the datasets in this study are open benchmark datasets which are available for download at: https://huggingface.co/datasets/takala/financial_phrasebank/tree/main and https://www.kaggle.com/code/ankurzing/sentfin/input?select=SEntFiN-v1.1.csv.

Received: 16 December 2024; Accepted: 26 June 2025
Published online: 04 July 2025

References

1. Malo, P., Sinha, A., Korhonen, P., Wallenius, J. & Takala, P. Good debt or bad debt: detecting semantic orientations in economic texts. *J. Am. Soc. Inf. Sci.* **65**, 782–796 (2014).
2. Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L. T. & Trajanov, D. Evaluation of sentiment analysis in finance: from lexicons to transformers. *IEEE Access* **8**, 131662–131682 (2020).
3. Arrieta, A. B. et al. Explainable artificial intelligence (xai): concepts, taxonomies, opportunities and challenges toward responsible ai. *Inf. Fusion* **58**, 82–115 (2020).
4. Huang, A. H., Wang, H. & Yang, Y. Finbert: a large language model for extracting information from financial text. *Contemp. Account. Res.* **40**, 806–841 (2023).
5. Hoover, B., Strobelt, H. & Gehrmann, S. Exbert: a visual analysis tool to explore learned representations in transformers models. arXiv preprint arXiv:1910.05276 (2019).
6. Aken, B. V., Winter, B., Löser, A. & Gers, F. A. Visbert: hidden-state visualizations for transformers. *Companion Proc. Web Conf.* **2020**, 207–211 (2020).
7. Jain, S. & Wallace, B. C. Attention is not explanation. arXiv preprint arXiv:1902.10186 (2019).
8. Pearson, K. Liii on lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **2**, 559–572 (1901).
9. Shao, Y. & Nakashole, N. On linearizing structured data in encoder-decoder language models: insights from text-to-sql. arXiv preprint arXiv:2404.02389 (2024).
10. Ribeiro, M. T., Singh, S. & Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1135–1144 (2016).
11. Ribeiro, M. T., Singh, S. & Guestrin, C. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32* (2018).
12. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **30**, 253 (2017).
13. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning* 3319–3328 (PMLR, 2017).
14. Fatouros, G., Soldatos, J., Kouroumali, K., Makridis, G. & Kyriazis, D. Transforming sentiment analysis in the financial domain with chatgpt. *Mach. Learn. Appl.* **14**, 100508 (2023).
15. Sousa, M. G. et al. Bert for stock market sentiment analysis. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)* 1597–1601 (IEEE, 2019).
16. Naresh, A. & Venkata Krishna, P. An efficient approach for sentiment analysis using machine learning algorithm. *Evol. Intell.***14**, 725–731 (2021).
17. Aslam, N., Rustam, F., Lee, E., Washington, P. B. & Ashraf, I. Sentiment analysis and emotion detection on cryptocurrency related tweets using ensemble lstm-gru model. *Ieee Access* **10**, 39313–39324 (2022).
18. Ahmad, H. O. & Umar, S. U. Sentiment analysis of financial textual data using machine learning and deep learning models. *Informatica* **47**, 142 (2023).
19. Ahmad, Z., Bangyal, W. H., Nisar, K., Haque, M. R. & Khan, M. A. Comparative analysis using machine learning techniques for fine grain sentiments. *J. Artif. Intell.* **4**, 49–60 (2022).
20. Daudert, T. Exploiting textual and relationship information for fine-grained financial sentiment analysis. *Knowl.-Based Syst.* **230**, 107389 (2021).
21. Consoli, S., Barbaglia, L. & Manzan, S. Fine-grained, aspect-based sentiment analysis on economic and financial lexicon. *Knowl.-Based Syst.* **247**, 108781 (2022).
22. Yadav, A., Jha, C., Sharan, A. & Vaish, V. Sentiment analysis of financial news using unsupervised approach. *Procedia Comput. Sci.* **167**, 589–598 (2020).

23. Xing, F., Malandri, L., Zhang, Y. & Cambria, E. Financial sentiment analysis: an investigation into common mistakes and silver bullets. In *Proceedings of the 28th International Conference on Computational Linguistics* 978–987 (2020).
24. Du, K., Xing, F. & Cambria, E. Incorporating multiple knowledge sources for targeted aspect-based financial sentiment analysis. *ACM Trans. Manag. Inf. Syst.* **14**, 1–24 (2023).
25. Zhang, B., Yang, H., Zhou, T., Ali Babar, M. & Liu, X.-Y. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the Fourth ACM International Conference on AI in Finance* 349–356 (2023).
26. Štrimaitis, R., Stefanovič, P., Ramanauskaitė, S. & Slotkienė, A. Financial context news sentiment analysis for the lithuanian language. *Appl. Sci.* **11**, 4443 (2021).
27. Liapis, C. M., Karanikola, A. & Kotsiantis, S. A multi-method survey on the use of sentiment analysis in multivariate financial time series forecasting. *Entropy* **23**, 1603 (2021).
28. Du, K., Xing, F., Mao, R. & Cambria, E. Finsenticnet: a concept-level lexicon for financial sentiment analysis. In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)* 109–114 (IEEE, 2023).
29. Hansen, K. B. & Borch, C. Alternative data and sentiment analysis: prospecting non-standard data in machine learning-driven finance. *Big Data Soc.* **9**, 20539517211070700 (2022).
30. Hartmann, J., Heitmann, M., Siebert, C. & Schamp, C. More than a feeling: accuracy and application of sentiment analysis. *Int. J. Res. Mark.* **40**, 75–87 (2023).
31. Chang, Y.-C., Ku, C.-H. & Le Nguyen, D.-D. Predicting aspect-based sentiment using deep learning and information visualization: the impact of covid-19 on the airline industry. *Inf. Manage.* **59**, 103587 (2022).
32. Zhang, T., Yang, K., Ji, S. & Ananiadou, S. Emotion fusion for mental illness detection from social media: a survey. *Inf. Fusion* **92**, 231–246 (2023).
33. Singh, V. et al. How are reinforcement learning and deep learning algorithms used for big data based decision making in financial industries-a review and research agenda. *Int. J. Inf. Manage. Data Insights* **2**, 100094 (2022).
34. Leelawat, N. et al. Twitter data sentiment analysis of tourism in thailand during the covid-19 pandemic using machine learning. *Heliyon* **8**, 412 (2022).
35. Kumar, S., Sharma, D., Rao, S., Lim, W. M. & Mangla, S. K. Past, present, and future of sustainable finance: insights from big data analytics through machine learning of scholarly research. *Ann. Oper. Res.* **2022**, 1–44 (2022).
36. Alsayat, A. Improving sentiment analysis for social media applications using an ensemble deep learning language model. *Arab. J. Sci. Eng.* **47**, 2499–2511 (2022).
37. Passalis, N. et al. Multisource financial sentiment analysis for detecting bitcoin price change indications using deep learning. *Neural Comput. Appl.* **34**, 19441–19452 (2022).
38. Valle-Cruz, D., Fernandez-Cortez, V., López-Chau, A. & Sandoval-Almazán, R. Does twitter affect stock market decisions? financial sentiment analysis during pandemics: a comparative study of the h1n1 and the covid-19 periods. *Cogn. Comput.* **14**, 372–387 (2022).
39. Rizinski, M., Peshov, H., Mishev, K., Jovanovik, M. & Trajanov, D. Sentiment analysis in finance: From transformers back to explainable lexicons (xlex). *IEEE Access* (2024).
40. Makridis, G. et al. Xai for time-series classification leveraging image highlight methods. In *International Conference on Management of Digital* 382–396 (Springer, 2023).
41. Ardizzone, E., Chella, A., Frixione, M. & Gaglio, S. Integrating subsymbolic and symbolic processing in artificial vision. *J. Intell. Syst.* **1**, 273–308 (1992).
42. Thompson, N. C., Greenewald, K., Lee, K. & Manso, G. F. The computational limits of deep learning. arXiv preprint arXiv:2007.05558**10** (2020).
43. Jordan, M. I. & Mitchell, T. M. Machine learning: trends, perspectives, and prospects. *Science* **349**, 255–260 (2015).
44. Xu, J., Zhu, B., Jiang, W., Cheng, Q. & Zheng, H. Ai-based risk prediction and monitoring in financial futures and securities markets. In *The 13th International Scientific and Practical Conference "Information and innovative technologies in the development of society" (April 02–05, 2024) Athens, Greece* 321 (International Science Group, 2024).
45. Go, E. J., Moon, J. & Kim, J. Analysis of the current and future of the artificial intelligence in financial industry with big data techniques. *Glob. Business Financ. Rev. (GBFR)* **25**, 102–117 (2020).
46. Ying, X. An overview of overfitting and its solutions. *J. Phys. Conf. Ser.* **1168**, 022022 (2019).
47. Leippold, M. Sentiment spin: attacking financial sentiment with gpt-3. *Financ. Res. Lett.* **55**, 103957 (2023).
48. Malo, P., Sinha, A., Korhonen, P., Wallenius, J. & Takala, P. Good debt or bad debt: setecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.* **65**, 523 (2014).
49. Sinha, A., Kedas, S., Kumar, R. & Malo, P. Sentfin 1.0: entity-aware sentiment analysis for financial news. *J. Am. Soc. Inf. Sci.* **73**, 1314–1335 (2022).
50. Peng, B. et al. Rwkv: Reinventing rnns for the transformer era. arXiv preprint arXiv:2305.13048 (2023).
51. Kabir, M. R., Bhadra, D., Ridoy, M. & Milanova, M. Lstm-transformer-based robust hybrid deep learning model for financial time series forecasting. *Sci* **7**, 7 (2025).
52. Cao, K., Zhang, T. & Huang, J. Advanced hybrid lstm-transformer architecture for real-time multi-task prediction in engineering systems. *Sci. Rep.* **14**, 4890 (2024).
53. Mathanker, S., Weckler, P., Bowser, T., Wang, N. & Maness, N. Adaboost classifiers for pecan defect classification. *Comput. Electron. Agric.* **77**, 60–68 (2011).
54. Baby, D. et al. Leukocyte classification based on feature selection using extra trees classifier: atransfer learning approach. *Turk. J. Electr. Eng. Comput. Sci.* **29**, 2742–2757 (2021).
55. Balakrishnama, S. & Ganapathiraju, A. Linear discriminant analysis-a brief tutorial. *Inst. Signal Inf. Process.* **18**, 1–8 (1998).
56. Bose, S., Pal, A., SahaRay, R. & Nayak, J. Generalized quadratic discriminant analysis. *Pattern Recogn.* **48**, 2676–2684 (2015).
57. Chang, C.-C., Li, Y.-Z., Wu, H.-C. & Tseng, M.-H. Melanoma detection using xgb classifier combined with feature extraction and k-means smote techniques. *Diagnostics* **12**, 1747 (2022).
58. Khan, M. S. I., Islam, N., Uddin, J., Islam, S. & Nasir, M. K. Water quality prediction and classification based on principal component regression and gradient boosting classifier approach. *J. King Saud Univ.-Comput. Inf. Sci.* **34**, 4773–4781 (2022).
59. Morris, J. X., Lifland, E., Yoo, J. Y. & Qi, Y. Textattack: a framework for adversarial attacks in natural language processing. arXiv preprint arXiv:2005.05909 (2020).
60. Lutz, B., Pröllochs, N. & Neumann, D. Sentence-level sentiment analysis of financial news using distributed text representations and multi-instance learning. arXiv preprint arXiv:1901.00400 (2018).
61. Sohangir, S., Wang, D., Pomeranets, A. & Khoshgoftaar, T. M. Big data: deep learning for financial sentiment analysis. *J. Big Data* **5**, 1–25 (2018).
62. Dettmers, T., Lewis, M., Belkada, Y. & Zettlemoyer, L. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Adv. Neural Inf. Process. Syst.* **35**, 30318–30332 (2022).
63. Schulp, J. J. Gamestop and the rise of retail trading. *Cato J.* **41**, 511 (2021).
64. Sorrosal-Forradellas, M.-T., Barberà-Mariné, M.-G., Fabregat-Aibar, L. & Li, X. A new rating of sustainability based on the morningstar sustainability rating. *Eur. Res. Manag. Bus. Econ.* **29**, 100208 (2023).
65. Kiritz, N. & Sarfati, P. Supervisory guidance on model risk management (sr 11-7) versus enterprise-wide model risk management for deposit-taking institutions (e-23): a detailed comparative analysis. *Available at SSRN 3332484* (2018).

66. Sloane, M. & Wüllhorst, E. A systematic review of regulatory strategies and transparency mandates in ai regulation in europe, the united states, and canada. *Data Policy* **7**, e11 (2025).

Author contributions

Conceptualization M.T.H.; software, M.T.H.; validation, M.T.H. and M.K.M.; formal analysis, M.T.H.; investigation, M.T.H. and M.J.H.; writing-original draft preparation, M.T.H.; writing-review and editing, M.T.H. and M.K.M.; project administration, M.T.H and M.K.M.; funding acquisition, M.J.H.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.J.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.