UNIVERSITY OF WARSAW | FACULTY OF ECONOMIC SCIENCES

WORKING PAPERS
No. 16/2024 (452)

ENHANCING LITERATURE REVIEW WITH NLP METHODS ALGORITHMIC INVESTMENT STRATEGIES CASE

STANISŁAW ŁANIEWSKI
ROBERT ŚLEPACZUK

WARSAW 2024

ISSN 2957-0506

WORKING PAPERS 16/2024 (452)

Enhancing literature review with NLP methods Algorithmic investment strategies case

Stanisław Łaniewski, Robert Ślepaczuk*

University of Warsaw, Department of Quantitative Finance and Machine Learning, Faculty of Economic Science

** Corresponding author: slaniewski@uw.edu.pl*

Abstract: This study utilizes machine learning algorithms to analyze and organize knowledge in the field of algorithmic trading, based on filtering 136 million research papers to 14,342 articles ranging from 1956 to Q1 2020. We compare previously used practices such as keyword-based algorithms and embedding techniques with state-of-the-art dimension reduction and clustering for topic modeling method (BERTopic) to compare the popularity and evolution of different approaches and themes. We show new possibilities created by the last iteration of Large Language Models (LLM) like ChatGPT. The analysis reveals that the number of research articles on algorithmic trading is increasing faster than the overall number of papers. The stocks and main indices comprise more than half of all assets considered, but the growing trend in some classes is much stronger (e.g. cryptocurrencies). Machine learning models have become the most popular methods nowadays, but they are often flawed compared to seemingly simpler techniques. The study demonstrates the usefulness of Natural Language Processing in asking intricate questions about analyzed articles, like comparing the efficiency of different models. We demonstrate the efficiency of LLMs in refining datasets. Our research shows that by breaking tasks into smaller ones and adding reasoning steps, we can effectively address complex questions supported by case analyses.

Keywords: trading, quantitative finance, neural networks, literature review, knowledge representation, natural language processing (NLP), topic modeling, model comparison, artificial intelligence

JEL codes: C4, C15, C22, C45, C53, C58, C61, G11, G14, G15, G17

1 Introduction

The motivation for this work is to explore the extent to which automatic methods can be utilized for reviewing scientific journals, starting from the largest possible dataset, refining it with rules and machine learning to identify topics of interest, and addressing complex questions.

With the exponentially growing number of scientific journals and papers, it is difficult to keep track of how current methods evolve and change in popularity Fire and Guestrin (2019). To organize knowledge in the field of algorithmic trading we did a thorough analysis enhanced by machine learning algorithms based on 136 million research papers from the S2ORC database, which consists of repositories such as SSRN and arXiv, Microsoft Academic Graph or international journals Lo et al. (2020).

Archives and public repositories enable the sharing of research studies at various stages of advancement, including initial preprints. As science becomes increasingly complex, interdisciplinary research involving multiple experts is on the rise. This has resulted in hundreds or thousands of papers being published each year in a specific field, making it challenging to keep track of the latest developments.

Fortunately, advances in technology give us tools that we leverage in this study. We demonstrate how exploratory and machine-learning-enhanced analysis can be performed on a set of papers obtained by filtering one of the largest databases of research publications. These techniques enable us to uncover insights and patterns that might have otherwise gone unnoticed, and ultimately improve our understanding of the field of algorithmic investment strategies.

2 Research Questions

Our work presents a methodology that can be replicated in other fields, but we focus specifically on the case of algorithmic investment strategies. Based on automatic analysis methods such as keyword extraction and topic modeling we analyze a huge amount of research papers, focusing on the dynamics of selected features and how scientists apply different models.

RQ1 - Is algorithmic trading becoming a more popular topic in scientific research? How do scientific themes evolve in articles about algorithmic trading?

In addition, we identify the most popular methods and assets in this field and examine their dynamics over time. In particular, we analyze how the popularity of asset classes as a scientific topic evolves, showing that significant events are visible in aggregated statistics (e.g. rise of the popularity of cryptocurrencies). We expect that as computational power continues to increase and access to data becomes easier, shorter time horizons are studied more frequently and machine learning methods used more frequently.

RQ2 - How does the popularity of different asset classes, time horizons, and models studied in articles change over time?

Finally, we want to answer which models or strategies seem to outperform the benchmark, and which parameters and hyperparameters are most important and often optimized. These questions are difficult for keyword-based systems, as they require an understanding of concepts such as various comparison methods, the fact that models can be trained on different datasets, etc. To address these challenges, we investigate how Large Language Models (LLMs), such as GPT, can enhance the quantitative literature review process that currently relies on N-grams, keywords, and topic modeling. We compare different versions across time and highlight how much more insight can be distilled from full papers than abstracts.

RQ3 - Which models outperform other models? How to optimize hyperparameters in these models?

- RQ3.1 - How far can we answer this question with SentenceBERT-based topic modeling?
- RQ3.2 - How far can we answer this question with GPT based model like ChatGPT?

RQ4 - How does a version of LLM change the analysis? How much more knowledge can we get from full papers instead of abstracts?

3 Literature review

Examples of literature review in algorithmic trading done in a usual, manual pattern can be seen in Ferreira et al. (2021), or the review of applied machine learning models as in Hewamalage et al. (2023).

One way to apply automatic rules to systematic literature reviews is to reduce the huge database with smart filtering. A filtered set is a good base for expert annotation. In Bao et al. (2019) authors first reduce the dataset using keywords to 3,740 papers. Then they manually annotated them to train two supervised models, SVM and CNN, to classify the paper into one of the two groups.

Pintas et al. (2021) conducted a systematic literature review (SLR) on feature selection methods for text classification, presenting 175 reviewed papers from 2013 to 2020. They did similar analyses and visualizations of the popularity of various features in the scientific literature over time, however, they used keywords for filtering and experts (manual work) for analysis.

Marshall and Wallace (2019) point out that this approach can be either not robust or not sustainable, as maintaining such software can be time-consuming and expensive. They also emphasize the uniqueness of SLR in Medicine, where screening out Non-Randomized Controlled Trials with classification models can be replaced with different use cases for ML in the literature review.

Some like Yu et al. (2022) enhance the manual process with Citespace for automatic citation visualisation. They also study journals and geophysical data, document clusters by topic, word cloud, and keyword burst analysis.

Examples of applications of more sophisticated NLP methods are in Hong et al. (2022), where ScholarBERT, a general-scientific BERT, outperforms even domain-specific embeddings. Another approach is presented in Cachola et al. (2020), where the summarization model for scientific papers is fine-tuned by using training datasets created by experts.

For topic modeling we follow BERTopic Grootendorst (2022), which has been successfully used in research Garcia et al. (2022). Our work is novel to enhance literature review with GPT Tom et al. (2020), comparing the performance of various versions and abstracts against full-text analysis. Previous attempts like Dowling and Lucey (2023) focused on writing new papers, while we use it for knowledge synthesis and answering intricate questions that keywords-based methods cannot. We follow findings of how to cope with GPT issues such as hallucination Li et al. (2023) or change in performance over time Tu et al. (2024).

4 Methodology

The methodology used in the paper can be summarised as follows.

To refine our dataset, we employed a comprehensive search strategy and various filtering techniques based on keywords, topics, expert knowledge, and journals. We then evaluated multiple embedding methods, including word2vec and universal-sentence-encoder, before selecting sentence BERT as the optimal approach for our problem. To perform topic modeling, we utilized state-of-the-art algorithms, such as BERTopic Grootendorst (2022), which involved dimension reduction using UMAP McInnes et al. (2018) and clustering with HDBSCAN McInnes et al. (2017).

We curated the outcomes of topic modeling algorithms to find the major themes and analyze which areas of research are growing most rapidly.

We calculated embeddings for our research questions and identified topics with the closest cosine distance. We subsequently validated our results by scrutinizing the papers using both expert knowledge and a GPT-based model (ChatGPT).

To illustrate our findings, we present an analysis that includes a statistical overview and supporting visualizations that highlight distinctions in papers, including the algorithms employed, the markets, and the main subjects of study. We also examine these differences across various dimensions such as time and popularity.

4.1 Data selection

To ensure we review a broad range of relevant research, we began by identifying the most suitable database. After careful consideration, we selected the S20RC database Lo et al. (2020), which is a huge corpus of over 136M scientific papers enriched with citation data derived from Semantic Scholar, a research tool developed at the Allen Institute for AI. They span over 70 years with the last entries from April 2020.

We designed a schema to extract a relevant database (corpus) of documents. Our approach could be replicated for any research topic, but we focused on algorithmic trading, which required a smart filtering process. We considered that essential research, models, or findings could be outside the scope of regular economic journals or be interdisciplinary.

Table 1: Frequency of keywords

Regular expression	Abstract	Title	Both
Algo(rithmic)* trading	615	390	841
Investment strateg.	9473	2921	11362
Vola(tility)* trading	86	54	129
High.frequency trading	832	719	1248
Investment system.	870	174	963
Benchmark strateg.	170	7	177
Pair.trading	67	35	85
Momentum (trading | strateg.)	1074	461	1315
Contrarian (trading | strateg.)	380	169	477
SUM	**13567**	**4930**	**16597**

Our filtering reduced the corpus to 16 197 documents. We removed from further analysis those for which we could not fetch an abstract, ending up with 14,342 documents. Table 1 presents the statistics for each keyword used in the process. Investment strategies were found to be the most popular, with almost 3k occurrences in titles and 10k in abstracts. The other keywords were mentioned over 2k in the title and 4k in the abstract.

5 Exploratory analysis

To address our research question about the popularity of algo trading strategies and methods over time, we conducted analyses of the dataset, including publication dates, citation data, and keyword-based and topic-modeling methods.

Furthermore, we preprocessed the collected documents by removing English stopwords, lemmatizing the words, and tokenizing the texts. We also calculated descriptive statistics to provide a detailed overview of the documents we collected.

To understand the corpus more thoroughly, we generated word clouds and found N-grams and noun chunks. The generated world cloud confirms we have captured relevant articles from the targeted domain. Analysis of N-grams revealed the most popular concepts such as efficient market hypotheses, limit order book signals, time series momentum, and models such as the Fama French factor model and CAPM. Additionally, we used Named Entity Recognition algorithms to identify the most commonly studied markets and countries in the scientific literature related to our selected topics.

5.1 Time horizon and top asset classes

We want to highlight three findings: first, the increasing popularity of algorithmic investment strategies by showing how it is becoming a greater part of the total database in Figure 1.

Second, the majority of publications deal with daily and monthly data. In Figure 2 we plot the frequency of each time horizon, which we defined by analyzing keywords such as "daily", or "5-minute", or "monthly" in the context of data or training periods. The evaluation based on sampling from results and checking manually gave an 80% positive rate.

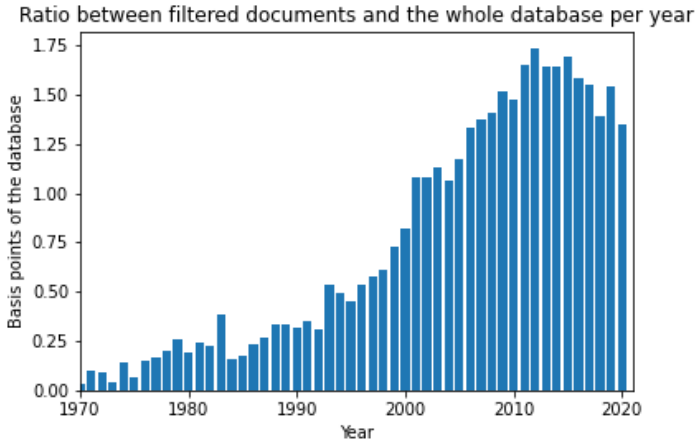

Fig. 1: How many basis points (0.0001) of whole S2ORC is in our dataset

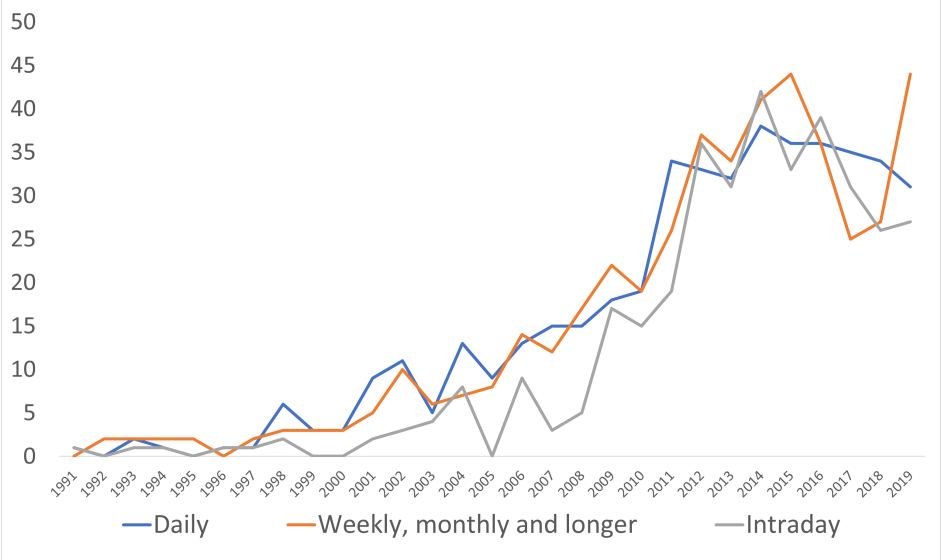

Fig. 2: Time horizon

By applying the same approach (Figure 3), we also found that researchers tend to focus on stocks, their indices, and derivatives, with more than half of the papers covering these topics. Cryptocurrencies have only recently become a topic of scientific interest, we also noticed increased interest in commodities around the time of the 2014-2016 oil crisis, which saw a 70 percent price drop.

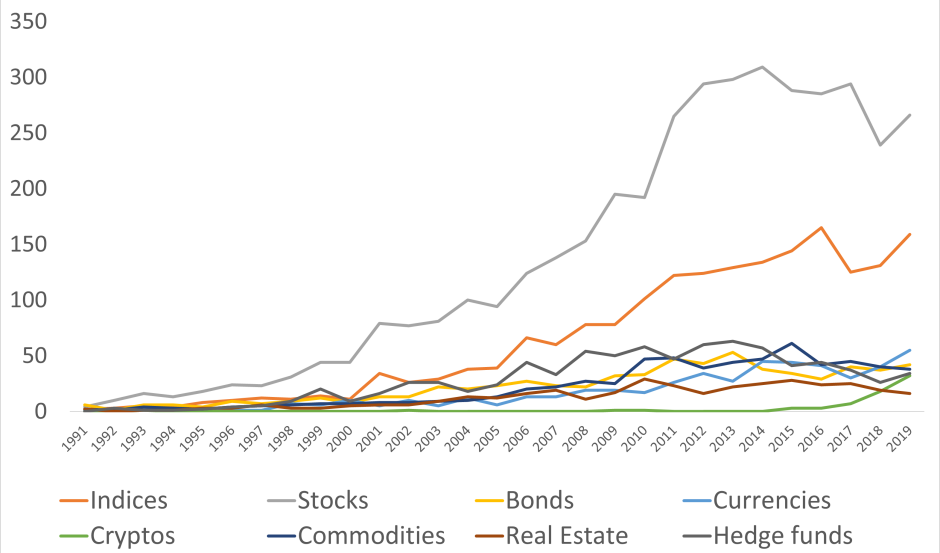

Fig. 3: Popularity of asset classes in time

5.2 Top methods used for modelling

To answer the question of the increasing popularity of machine learning-based methods in recent years, we aggregated them and compared them to linear models and time series. Although linear models account for more than half of all methods considered in the database, we examine the trends of different model families over time. We do this by using regular expressions to search for specific model names and grouping them into three categories: linear, time series, and machine learning (Figure 4). We then plot the number of papers per year that mention each category of models (full regex in Appendix A.1).

Machine learning methods are promising and gaining popularity; even though they have gained popularity recently, the real boom happened during 2016-2019 with ML-based methods taking over linear models for the first time in history. This supports trends found by previous researchers, e.g. Ferreira et al. (2021). Neural network is the most researched system from the machine learning environment. Furthermore, we see that time series modeling is not picking up traction and the ratio of papers using them in algorithmic trading scope is decreasing.

Our analysis shows that machine learning methods are rapidly gaining popularity in algorithmic trading research, especially since 2015. In 2019 machine learning methods surpassed linear models in popularity for the first time in history. The neural network is the most researched system from a machine learning environment. On the other hand, the use of time series modeling appears to be losing traction, with a decreasing ratio of papers incorporating them in the algorithmic trading scope.

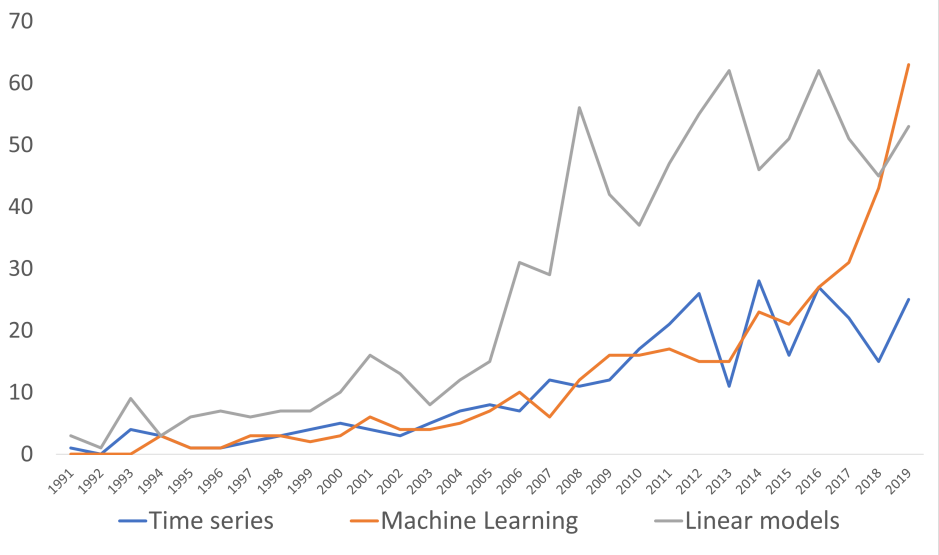

Fig. 4: Popularity of models classes in time

6 Topics

6.1 Procedure

To delve deeper into the underlying topics of the research papers, we recognize the need to augment our analysis with more advanced techniques. The statistical-based methods employed thus far have provided valuable insights, but to gain a more nuanced understanding, we require the ability to comprehend language, identify similarities between words and sentences, and extract meaningful summaries from the texts. By doing so, we can uncover the topics that are considered crucial by the scientific community.

To do so we apply three various embeddings word2vec, BERT, and Universal Sentence Encoder - to better understand the language, find similarities between words and sentences, and summarise the texts. While word2vec has been successfully used in previous research, we found that its lack of sentence interpretation could be a potential flaw in our analysis. After testing various sentence transformer-based methods, we ultimately chose the 384-dimensional all-MiniLM-L6-v2 model for its superior performance while maintaining a small size Wang et al. (2020).

To reduce the dimensionality of our embedded documents, we use Uniform Manifold Approximation and Projection McInnes et al. (2018). UMAP is a non-linear dimension reduction algorithm that combines aspects of principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE). By using UMAP, we aimed to preserve the essential global structures of the documents, making it easier to identify similar topics.

After applying UMAP, we further grouped the documents using the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm McInnes et al. (2017), as it was capable of detecting clusters of different densities and

able to hand outliers. Therefore documents that had similar embeddings, such as those of the same topic or containing significant word overlap, end up grouped together based on their lower-dimensional representations.

Following the implementation of BERTopic Grootendorst (2022) we used the class version of TF-IDF. The documents which fall into the same cluster create a topic.

Finally, we curated outcomes of topic modeling algorithms to identify common themes and determine which areas of research are growing most rapidly. We used a technique of merging smaller topics with the closest thematically larger ones based on the shortest Euclidean distance in the lower dimensional space. This allowed us to reduce the number of topics to 20. To validate our results, we sampled 50 documents and manually checked them. Lastly, we prompt ChatGPT to generate a 3-word title for each topic based on the top 10 words and scores from the TF-IDF table.

6.2 Analysis

We identified three distinct clusters: one major group and two smaller (Figure 5). The middle cluster is centered on strategic investments, such as those in transportation, the military, and electricity, while the top-left cluster focuses on education, agriculture, foreign investment, and development. The main group in the bottom-right is about investment strategies; we notice some sub-clusters about the pension system (upper part of the group), the main part consisting of various strategies, and in the bottom right optimal investments for longer periods (from portfolio manager and insurer perspective).

We can also notice that topic 16 (Neural Network Trading) is interestingly first matched with topic 4 (HFT) and 13 (Volatility) rather than general group 0, which matches with 19 (Figure 6). This suggests that topic 16 has stronger connections with the specialized areas covered by topics 4 and 13.

Topics that are close to each other based on our model will be grouped first. Comparing to 5, we notice that indeed most often topics that were clustered together are grouped first, e.g. 7: Foreign Direct Investment and 11: Social Welfare Policies, or two pairs 2 & 5 and 6 & 8. (2: Renewable Energy Planning, 5: Real Options Analysis, 6: Innovation Technologies Investment, 8: Transportation Planning Strategies).

In Figure 7 we notice that trends change over time, for example, topic 4 (HFT) experienced bursts of popularity in response to events such as the first flash crash. On the other hand, topic 16 (Neural Network Trading) has become one of the fastest-growing areas of research in this field in recent years.

6.3 Neural Network Trading

To analyze the topic that compares different models used in algorithmic trading, we started by identifying relevant keywords and queries, such as "model outperforms", "hyperparameter optimization", "learning rate", "comparing models", and specific method-related terms like "recurrent neural network", "LSTM", and "reinforcement learning". For each query we created an embedding, compared them with the topic embeddings, and identified the most similar topics by calculating the distance based on cosine similarity values (Table 2, the higher Simil. the better).

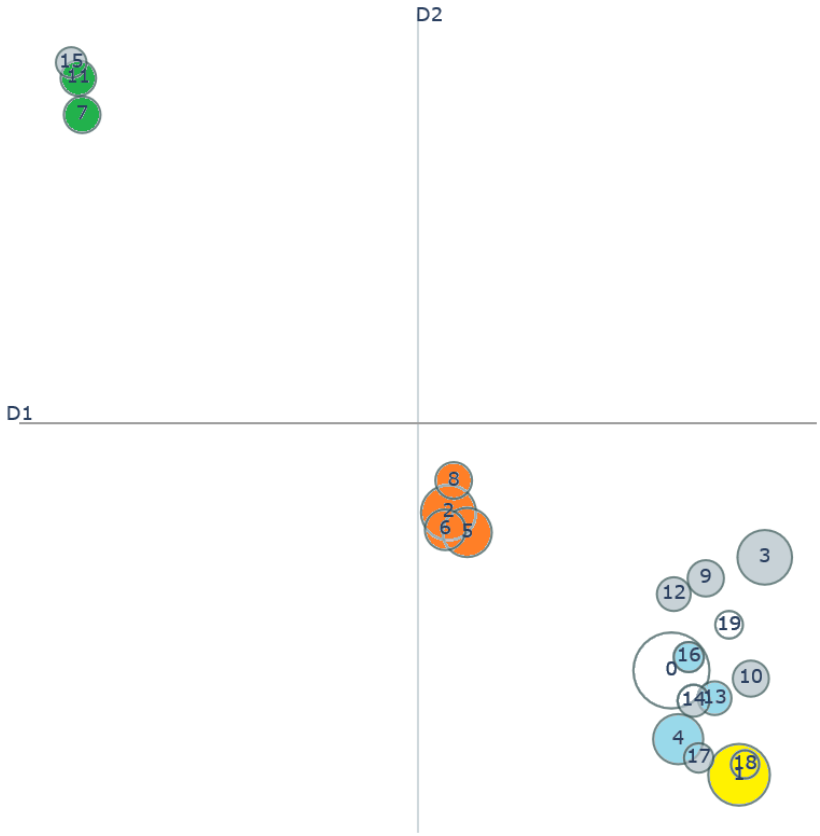

Fig. 5: Topic clusters

For each of our queries, topic 16 (Neural Network Trading) gets the highest cosine similarity. The 10 top words of this topic include trading, prediction/forecasting, neural network, stock, and machine learning.

In the limited domain of 176 research papers, we conducted a detailed analysis to answer our research questions: what assets and venues are most frequently used, how are they tested (models), and which techniques perform better (Table 3 and Tables A1,A2 in Appendix). We use keyword- and LLM-based methods and validate them by asking experts to read through the abstracts as well.

As expected, this topic is dominated by neural networks and reinforcement learning. Nonetheless, other methods such as rough sets for data mining, support vector regression, support vector machines, and various mapping or pattern searching algorithms are also prominent. Several models incorporate machine learning-based feature

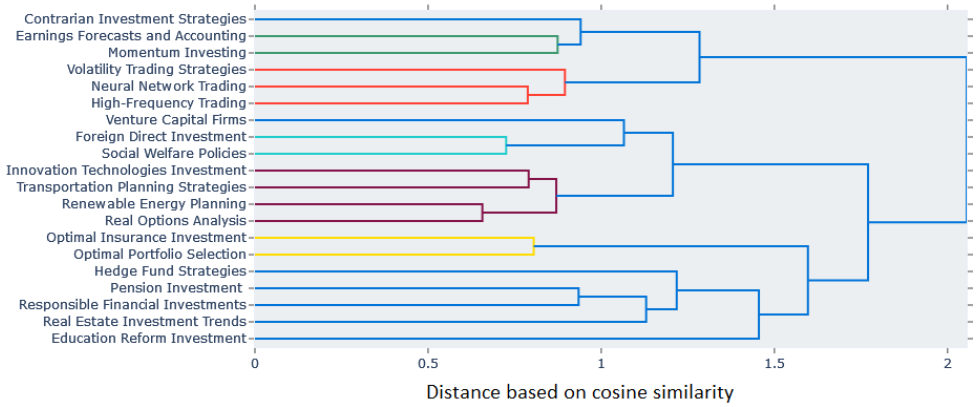

Fig. 6: Hierarchical clustering

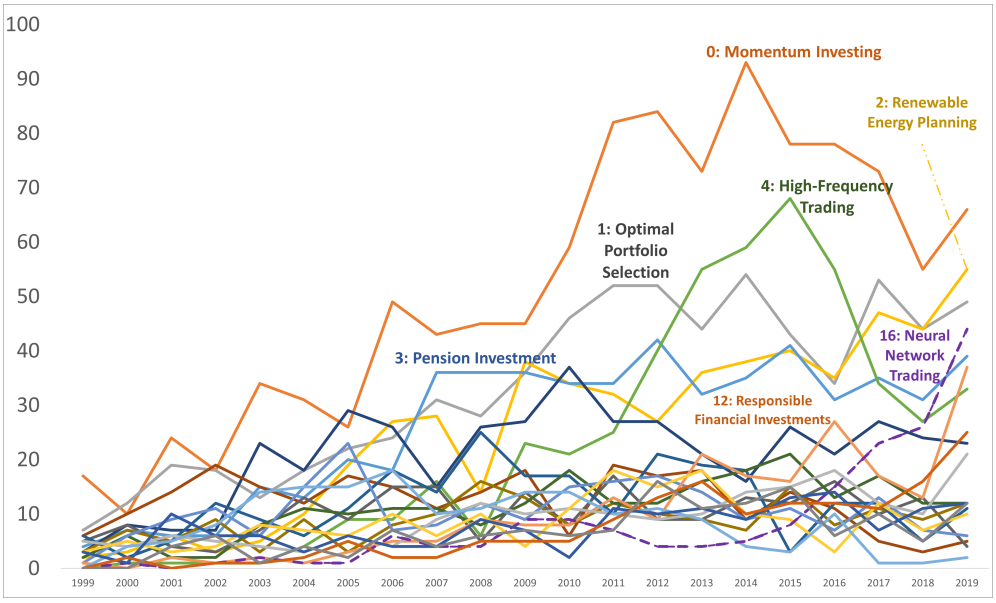

Fig. 7: The trend of 20 main topics in the last 20 years with labels for 7 top topics in 2019

creation, particularly based on technical analysis. Linear models and Buy-and-Hold strategies are often used as benchmarks, although not always since some B&H strategies are based on classification models for market entry and exit.

For model comparison or hyperparameter optimization (HPO), unfortunately, frequently there was minimal detailed information beyond statements such as 'Model X is compared with model Y'. Sometimes they were just contrasted with simple benchmarks or some other strategy with inconclusive results (not mentioned in the abstract,

Table 2: Top topics for selected queries

Method	Value	1st	2nd	3rd	4th	5th
Model Outperforms	Topic	16	5	1	-1	2
	Simil.	0.43	0.30	0.30	0.28	0.28
Learning Rate	Topic	16	15	2	1	5
	Simil.	0.42	0.31	0.28	0.27	0.26
Hyperparameter Optimization	Topic	16	1	2	5	13
	Simil.	0.34	0.23	0.22	0.20	0.19
Comparing Models	Topic	16	13	5	2	6
	Simil.	0.33	0.24	0.23	0.23	0.21
LSTM	Topic	16	8	19	13	18
	Simil.	0.50	0.26	0.26	0.25	0.25
Recurrent Neural Network	Topic	16	1	2	18	-1
	Simil.	0.58	0.30	0.28	0.28	0.27
Reinforcement Learning	Topic	16	2	5	1	6
	Simil.	0.60	0.51	0.46	0.44	0.43

Table 3: Top 5 techniques used in topic 16 (Neural Network Trading)

Model	Count
Neural Network (NN)	80
Imitation Learning, Reinforcement Learning, Q-Learning/Network, Actor-Critique, A3C	62
Machine Learning	48
Data Mining, Rough Set, Fuzzy	37
Technical Analysis (TA), Technical Indicator, MACD, Oscillator	36

nor the comparison method). Additional keyword frequency listed in Table 4 suggests that a more sophisticated model is required for such complex questions.

Table 4: How often were the models compared?

Compare	51
Accuracy	39
Outperform	29
Benchmark	18
Sharpe	5
Precision, Recall, F1	2

Table 5: What does ChatGPT tell us?

	No model comparison	Comparing models
No HPO	64	48
With HPO	35	29

7 Analysis with LLM

To answer RQ3&4 and to evaluate the efficiency of LLM, we test two ChatGPT models, specifically ChatGPT 3.5 (23.03.23) and ChatGPT-4o (01.06.24), on the selected subset of papers, namely the ones labeled in topic modeling as topic 16 - Neural Network Trading.

7.1 Comparing GPT versions on abstracts

We employed both ChatGPT versions 3.5 and 4o to answer RQ3 regarding comparing models and hyperparameter optimization (HPO). We designed a prompt that asks if each abstract contains two aspects: a comparison of different models or methods used and hyperparameter optimization (Table 5). We required each answer to be summarised with a yes/no response. To validate the results, we manually evaluated the abstracts and checked the longer answers provided by the LLM.

The 4o ChatGPT reveals an increase in the number of papers identified as comparing models and performing HPO. The overall number of papers without model comparison decreased by 17, illustrating the effectiveness of the 4o approach in uncovering methodological details.

Table 6: Confusion Matrix Comparing LLM versions (3.5 Turbo to 4o)

HPO	Category	4o Abstracts Count	3.5 Abstracts Count	Difference (4o - Abstracts)
No HPO	No model comparison	47	64	-17
	Comparing models	98	48	50
With HPO	No model comparison	6	35	-29
	Comparing models	25	29	-4
	Total Sum	176	176	**0**

In 3.5, model comparison was not observed in 99 abstracts, whereas in 4o, it was noted in only 53 abstracts, marking a difference of 46 papers. A significant portion of this difference can be attributed to the classification of abstracts as comparing models without HPO: 98 abstracts in 4o compared to 48 abstracts in 3.5, accounting for a difference of 50 papers. Additionally, 4o adopted a more stringent criterion for HPO, identifying only 31 abstracts as employing such methods, compared to 64 abstracts in 3.5 (difference of 33 papers).

7.2 Full text analysis

From the 176 articles on the topic of Neural Network Trading, we accessed 153 full papers and removed 7 biggest files (books), ending up with 146 full texts for analysis.

Furthermore, since we analyze full papers now, we asked more elaborate questions. There are 3 new questions added to model comparison and HPO, namely frequency of data used, loss function used, and what was chosen as the best model. We expect two answers for each question - one with yes/no, the other with the explanation provided for each question.

```
If there is a comparison of different models or methods used.
If there is hyperparameter optimization.
The frequency of data used.
The loss function used.
The best model (chosen in comparison).
```

7.2.1 Comparing to Abstracts

Despite having 30 fewer full texts than abstracts, the LLM was able to find snippets where researchers compared models or performed HPO, leading to a significant increase of 59 affirmative answers to both questions. Additionally, the overall number of papers without model comparison decreased by 42, illustrating the effectiveness of full-text analysis in uncovering methodological details.

The significant increase in the detection of model comparisons and HPO highlights the necessity of full-text analysis for comprehensive research reviews. This study demonstrates that intricate methodological nuances are often embedded deeper in the papers, which can be effectively uncovered using advanced language models.

Table 7: Confusion Matrix Comparing Full Texts and Abstracts

HPO	Category	Full Texts Count	Abstracts (4o) Count	Difference (Full Texts - 4o)
No HPO	No model comparison	6	47	-41
	Comparing models	51	98	-47
With HPO	No model comparison	5	6	-1
	Comparing models	84	25	59
	Total Sum	146	176	**-30**

We notice increased detection of model comparison and HPO (comparing models and performing HPO rose from 123 and 31 in abstracts to 135 and 89 in full texts respectively) and reduction in papers without methodological details (not comparing models, not performing HPO fall from 53 and 145 in abstracts to 11 and 57 in full texts respectively).

7.2.2 Time intervals

The full-text analysis provides us with more accurate information about the frequency of data. We defined the bins by taking a list of unique answers (A.6) and grouping them manually.

Table 8: Frequency of Data Used

	Intraday	Daily	Longer	Not specified
Count	37	73	24	12
Regex on Abstracts	15	16	8	119

7.2.3 Loss functions

The majority of the loss functions fall into the "Other/Unspecified" category, indicating a variety of less commonly named or unique loss functions. The most commonly specified loss function group is MSE-related, followed by Cross-entropy-related.

Table 9: Summary of Loss Function Instances

Loss Function Group	Instances
MSE Related	28
Cross-Entropy Related	13
Other Common Loss Functions	11
Specialized/Custom Loss Functions	11
RMSE Related	8
Sharpe Ratio Related	4
MAPE Related	1
Other/Unspecified	69

To present the results, we grouped the loss functions based on expert knowledge A.7.

7.2.4 Best models

Here is the summary based on NLP and lazy ChatGPT (that is, the one that uses Python to analyze data instead of reading manually, as it's longer than its context).

Table 10: Summary of Best Model Categories based on NLP and ChatGPT

Model Category	Count
Neural Networks	25
Traditional Statistical Models	13
Recurrent Neural Networks	12
Reinforcement Learning	5
Self-Organizing Maps (SOMs)	4
Ensemble Methods	3
Fuzzy Logic Models	3
Other/Unspecified	70

11 is the detailed categorization of models that reflects the thorough analysis performed by prompting LLM with each answer and its corresponding elaboration. This method of batching ensures that the LLM meticulously 'reads' the elaboration, leveraging its capabilities to provide accurate and insightful classifications.

Not only does this result in a more precise classification of models (with the "Other/Unspecified" category dropping from 70 to 20), but it also captures more categories

and nuances, such as creating a distinct topic for deep learning models. This enhanced granularity in classification demonstrates the LLM's capability to discern subtle differences and provide a comprehensive overview of the diverse range of models used in the studies.

Table 11: Detailed Summary of Best Model Categories

Model Category	Count
Deep Learning Models	25
Traditional Statistical Models (including 6 Trees)	21
Neural Networks	19
Recurrent Neural Networks and extensions	13
Reinforcement Learning	11
Ensemble Methods and Hybrid Models	11
Specialised Models	11
Support Vector Machine Models	8
Rough sets	7
Not applicable/Unspecified	20

The full list can be found in A.8.

7.3 Issues

7.3.1 LLM Laziness

First of all, the answering scheme is different than one would imagine AI to use. By default, ChatGPT does not read and understand the papers. Instead, it uses regex and NLP methods to answer each question. When prompted, it even provided us with the Python code it used for analysis A.4. As expected, it is suspect to simple false positives (the word 'compare' is used in different contexts) or to omitting keywords (not provided in the short list of options).

For example, 12 is the initial analysis provided for each question, as it simply treated it all as one batch and ran a regex analysis on it. It performed surprisingly well - the words selected in regex produced quite accurate results for the number of papers with model comparison (138 compared to 135 based on LLM full-text analysis, 7) and for HPO (96 compared to 89). In data frequency, question it had problems capturing intraday horizons, thus predicting 107 papers to state frequency of dataset used, compared to 134 in LLM full paper analysis 8. Again it did well in the loss function, stating that 65 are unspecified, compared to 69 in 9, while the choice of the best model proved to be too difficult question for regex - 127 papers identified as not stating best method compared to 20 based on full-text analysis by LLM 11.

To use LLM capabilities we specifically mentioned we want it to use the context. It would be wiser to find specific keywords for Regex search based on the abstract. Then read the context and decide what is the final answer with elaboration. To confirm it followed our guidance, we asked it to summarise logic afterward A.5.

Table 12: Is there information in full text about:

	Model Comparison	HPO	Data frequency	Loss function	Best model
No	8	50	39	65	127
Yes	138	96	107	81	19

The results prove the efficiency of keyword-based searches, such as regex-based, with domain expert knowledge used to select them, as an efficient way to do filtering and find some simple information. However, more complex questions require attention - a small batch of one paper, which then can be filtered down to elaborations on the most crucial parts. A.8 shows that such a method with LLM can be an insightful and effective way to use in research.

7.3.2 Errors and consistency

The most common error encountered was due to excessively large files, such as loading a book with over 100 pages. In some instances, papers were too mathematical for GPT to parse and understand accurately. Parsing such files did not necessarily produce an error; instead, the model either attempted to list chapters as different papers or began to hallucinate based on its partial understanding, as found in research (e.g. Li et al. (2023)). Following their findings, we provide external domain knowledge and break the task by adding reasoning steps.

Excluding books (7 instances, which were easy to filter out), there were 4 errors out of 150 files. To maintain accuracy, it was crucial to send files in small batches—preferably one by one— as larger batches led to confusion and loss of context in GPT. Following best practices from social experiments, we included questions to check GPT's attention (such as asking it to summarize the task at hand), which it passed.

There are idiosyncratic risks associated with relying on a single LLM. For example, GPT-3.5 Turbo 23.03 frequently identified the need for hyperparameter optimization (HPO) while rarely recognizing model comparison. Consequently, when abstracts did not explicitly mention HPO but the models required proper tuning, ChatGPT might incorrectly affirm that HPO was performed. Conversely, GPT-4.0 demonstrated more strict criteria for HPO (33 fewer papers identified) but recognized more instances of model comparison (46 more papers). In the full-text analysis, GPT-3.5 performed better on HPO-related questions, while GPT-4.0 excelled in identifying model comparisons. This domain-specific comparison of performance over time, which shows irregularities, is in line with findings in Tu et al. (2024).

Another issue observed was the lack of consistency in the LLM's responses. For example, it sometimes did not consider arbitrarily selected benchmarks (e.g., buy and hold) as model comparisons, treating them merely as sanity checks, whereas in other abstracts it did. Occasionally, it treated parameter tuning as HPO. However, analyzing longer answers revealed that the LLM's certainty varied. To address this, we incorporated the elaborations provided by the LLM in our full-text analysis, which enhanced the reliability of the responses.

8 Conclusions

8.1 Case. Algorithmic trading literature review

Recent advancements in computer science and natural language processing have enabled researchers to access vast databases of scientific papers and narrow them down to their areas of interest. In our study of algorithmic investing strategies, we used a keyword-based approach to filter a large dataset of research papers. Our analysis revealed that algorithmic trading has become increasingly popular over time, particularly between 1990 and 2010. In recent years, shorter time horizons have gained popularity, driven by cheaper computational power and easier access to relevant data. While stocks and indices are the most commonly studied assets, other asset classes have experienced spikes in popularity during certain periods, such as the oil crisis of 2014-2016 or the rise of cryptocurrencies after 2018.

Machine learning-based techniques have become the most widely tested statistical models in the field. Our topic modeling analysis revealed major trends in contemporary research and identified the topic of comparing various algorithms and models, particularly those based on ML. While keyword-based approaches are useful for finding popular methods and their intersections, they have limitations in answering questions about which models outperform others in general.

Full-text analysis has confirmed that HPO, while not often being the main focus of the study, is performed in the vast majority of the papers. Recent studies Probst et al. (2018); Li et al. (2020) have shown that many fine-tuned algorithms are sensitive to changes in hyperparameters, so it is important to be cautious about the robustness of some methods.

Deep learning models prove to be the most promising models for algo trading, closely followed by traditional statistical models. However, there are many successful neural networks, especially recurrent ones, and there is plenty of research applying reinforcement learning or ensemble methods.

8.2 LLM for literature review

Both regex-based filtering and LLM proved to be successful and useful in refining a huge corpus of research papers. Abstracts, while giving some insight into the study, often omit parts that are found later in the paper (e.g., details about models, HPO, or data). Furthermore, the results varied from version to version, showing inconsistencies reported in previous studies.

ChatGPT has shown that without reasoning steps, it tends to oversimplify the problem. By breaking the task - reading and understanding research papers - into simpler parts and guiding the process, we were able to extract nuanced knowledge about used models, datasets, or (loss) functions. This proved that full paper analysis with LLM can be a sophisticated method of knowledge extraction.

The study confirmed the added value of a step-by-step approach. By grouping the papers into small batches, we were able to first extract the information in a particular context and save it as elaboration, which was then used in further steps of the analysis. This approach yielded the most accurate and elaborate results.

Table A1: Asset classes and venues in topic 16

Asset	Count
Stocks	104
Indices	61
Commodities	14
Currencies	11
Bonds	3
Cryptos	3

Market Indices/Location	Count
S&P	16
EU	8
Hang Seng & other Chinese	8
KOSPI/Korea	7
NYSE	5
DJIA	5
Bovespa/Brazilian	3
Nikkei/Japan	3
other US	2

9 Declarations

Large Language Models, namely ChatGPT 3.5and 4o, were used in this research for evaluation in 8, as well as for text, code, and table polishing.

Appendix A Appendix

A.1 Classification and Regex for models

Here is the list of regular expressions for each topic:

Linear models: ordinary least square| OLS| linear model.| lasso| ridge

Machine Learning: random forest| decision tree.| regression tree.| xgboost| boosting| extreme gradient| LSTM| Long.short.term.| support vector regressions| SVR| support vector machine| SVM| k.nearest neighbour.| knn| clustering algo.| mapping algo.| neural network| (imitation| reinforcement| unsupervised) learning

Time series: GLM| Generalized linear model| (Poisson(.?point)?| Gaussian| Normal) (proces.| regress.)| (s)?ar(i)?ma(x)?| garch

A.2 Asset classes and venues in topic Neural Network Trading

While we were able to find the traded asset class, the identification of the venue based on keyword search failed to deliver meaningful results based on abstracts. By human validation, we confirmed that in over two-thirds of this reduced dataset, there is no mention of the particular assets.

A.3 Topics per year

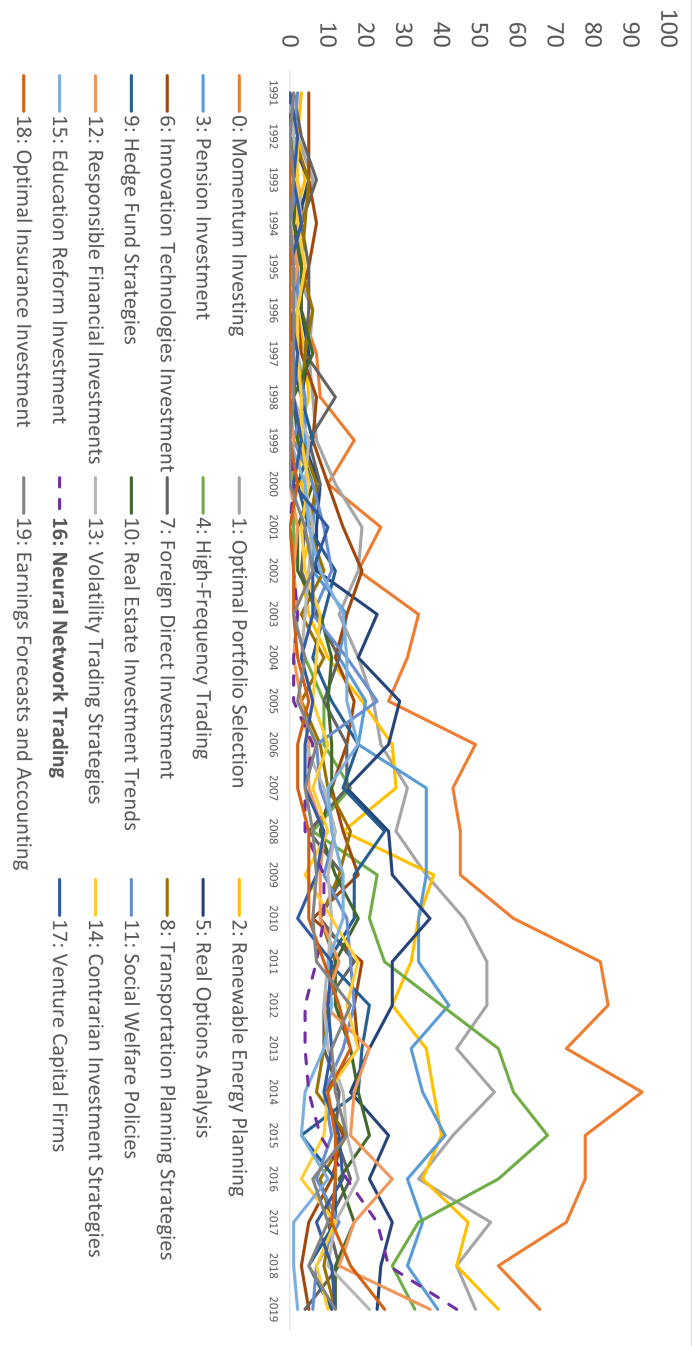

Fig. A1: The trends of 20 main topics

Table A2: Techniques used in topic 16: Neural Network Trading

Model	Count
Neural Network (NN)	80
Imitation Learning, Reinforcement Learning, Q-Learning/Network, Actor-Critique, A3C	62
Machine Learning	48
Data Mining, Rough Set, Fuzzy	37
Technical Analysis (TA), Technical Indicator, MACD, Oscillator	36
K-Nearest Neighbour (KNN), Clustering Algo, Mapping Algo, Pattern	27
Rule-based system	26
Support Vector Regression (SVR), Support Vector Machine (SVM)	25
GLM, Classification, Logistic, Multinomial Regression	22
Buy and Sell, Buy Sell, Buy and Hold (B&H)	18
Random Forest, Decision Tree, Regression Tree, CART, CHAID	16
Long Short-Term Memory (LSTM)	13
Ordinary least square (OLS), linear model	11
Ensemble, voting	9
Convolutional Neural Network (CNN)	8
Fourier and Kernel tricks	8
Recurrent Neural Network (RNN)	7
XGBoost, Boosting, Extreme Gradient	6
Genetic algorithms	6
Correspondence Analysis (CA)	5
Principal Component Analysis (PCA), encoder, autoencoder	5
SARIMA, GARCH	5
Extreme Learning Machine (ELM)	4
Stochastic Gradient Descent (SGD)	2
Hyperparameter, Hyperparameter Optimization (HPO)	2
Lasso, Ridge	2
Natural Language Processing (NLP)	2
eXplainable AI	1
Residual Neural Network (ResNet)	0

A.4 Lazy GPT regex

```
import re
import pdfplumber

def extract_text_from_pdf(pdf_path):
    with pdfplumber.open(pdf_path) as pdf:
        text = ""
        for page in pdf.pages:
            text += page.extract_text()
    return text

def analyze_text(text):
    results = {
        "comparison": {"yes_no": "No", "elaboration": ""},
        "hyperparameter_optimization": {"yes_no": "No", "
            ↪ elaboration": ""},
        "data_frequency": {"yes_no": "No", "elaboration": ""
            ↪ },
        "loss_function": {"yes_no": "No", "elaboration": ""},
        "best_model": {"yes_no": "No", "elaboration": ""}
    }

    # Comparison of Different Models or Methods
    if re.search(r'comparison|compare|benchmark|evaluate|
        ↪ versus|comparison-study|side-by-side|comparative-
        ↪ analysis', text, re.IGNORECASE):
        results["comparison"]["yes_no"] = "Yes"
        results["comparison"]["elaboration"] =
            ↪ extract_comparison_details(text)

    # Hyperparameter Optimization
    if re.search(r'hyperparameter|tuning|optimization|grid-
        ↪ search|random-search|bayesian-optimization|
        ↪ hyperparameter-tuning|parameter-search|hyper-
        ↪ optimization', text, re.IGNORECASE):
        results["hyperparameter_optimization"]["yes_no"] = "
            ↪ Yes"
        results["hyperparameter_optimization"]["elaboration"]
            ↪  = extract_hyperparameter_details(text)

    # Frequency of Data Used
    frequency_match = re.search(r'daily|weekly|monthly|minute
        ↪ -level|hourly|annually|yearly|bi-weekly|quarterly',
        ↪  text, re.IGNORECASE)
```

```
    if frequency_match:
        results["data_frequency"]["yes_no"] = "Yes"
        results["data_frequency"]["elaboration"] =
            ↪ frequency_match.group(0)

    # Loss Function
    loss_function_match = re.search(r'mean-squared-error|mse|
        ↪ mean-absolute-error|mae|cross-entropy|log-loss|
        ↪ hinge-loss|squared-loss|absolute-error|mean-bias',
        ↪ text, re.IGNORECASE)
    if loss_function_match:
        results["loss_function"]["yes_no"] = "Yes"
        results["loss_function"]["elaboration"] =
            ↪ loss_function_match.group(0)

    # Best Model
    best_model_match = re.search(r'best-model|optimal-model|
        ↪ most-accurate|highest-performing|top-model|leading-
        ↪ model|best-performing', text, re.IGNORECASE)
    if best_model_match:
        results["best_model"]["yes_no"] = "Yes"
        results["best_model"]["elaboration"] =
            ↪ extract_best_model(text)

    return results

def extract_comparison_details(text):
    # Implement a detailed extraction logic for comparison
        ↪ details
    comparison_sentences = re.findall(r'comparison|compare|
        ↪ benchmark|evaluate|versus|comparison-study|side-by-
        ↪ side|comparative-analysis.*?\.', text, re.
        ↪ IGNORECASE)
    return "-".join(comparison_sentences)

def extract_hyperparameter_details(text):
    # Implement a detailed extraction logic for
        ↪ hyperparameter details
    hyperparameter_sentences = re.findall(r'hyperparameter|
        ↪ tuning|optimization|grid-search|random-search|
        ↪ bayesian-optimization|hyperparameter-tuning|
        ↪ parameter-search|hyper-optimization.*?\.', text, re
        ↪ .IGNORECASE)
    return "-".join(hyperparameter_sentences)
```

```
def extract_best_model(text):
    # Implement a detailed extraction logic for best model
    best_model_sentences = re.findall(r'best-model|optimal-
        ↪ model|most-accurate|highest-performing|top-model|
        ↪ leading-model|best-performing.*?\.', text, re.
        ↪ IGNORECASE)
    return "-".join(best_model_sentences)

# Example usage
pdf_path = "example_research_paper.pdf"
text = extract_text_from_pdf(pdf_path)
results = analyze_text(text)
print(results)
```

A.5 LLM Logic

Reviewing the Abstract: I examined the abstract of the paper to get an overview of its focus, methods, and key findings.

Identifying Comparisons: I looked for any mention of comparisons between different models or methods within the abstract and any additional text available from the paper. This included looking for keywords such as "compare," "comparison," "evaluate," "versus," and "against."

Hyperparameter Optimization: I searched for information on any tuning or optimization of hyperparameters. This typically involves looking for terms like "optimize," "hyperparameter," "parameter tuning," and "settings."

Frequency of Data: I checked for any mention of the data frequency used in the study. This could include daily, weekly, monthly, or any other specific time intervals mentioned in relation to the data.

Loss Function: I looked for any explicit mention of a loss function used in the paper. If a specific loss function was not mentioned, I inferred the optimization criteria from the context, such as the focus on maximizing returns or minimizing errors.

```
Best Model: I reviewed the findings to identify which model
    or configuration was reported as the best performing one.
     This often involves looking for terms like "best," "
    optimal," "highest performance," and specific model names
     or configurations.
```

A.6 Unique time horizons found in Neural Network Trading topic

- Intraday
  - 'Minute', 'Milliseconds', 'Intra-day', 'High-frequency (minute-level)', 'Hourly', '30-minute', 'High-frequency', 'Intraday', '5-minute intervals', '15-minute', 'High Frequency', '30-minute intervals', '5, 10, and 15-minute intervals', '10-minute intervals', 'Tick-level (every microsecond)', '1-minute intervals', 'Tick-level (microseconds)', 'High-frequency financial data sampled at an interval of one minute', 'High-frequency financial data sampled at one-minute intervals', 'Minute-level', 'High-frequency (5-minute intervals)'
- Daily
  - 'Daily', 'Daily and weekly', 'Daily and minute-level', 'Daily and 15-minute intervals', 'Daily and Monthly', 'Daily, Monthly, Yearly'
- Longer
  - 'Yearly', 'Quarterly', 'Monthly', 'Weekly', 'Various'

A.7 Grouping of loss function found in Neural Network Trading topic

1. **Mean Squared Error (MSE) Related:**
   - Mean Squared Error (MSE)
   - Mean Square Error (MSE)
   - Mean Squared Error with penalizing coefficient
   - Sum of Square Errors
   - Mean Squared Forecast Error (MSFE)
   - Mean Squared Error (MSE) and Cross-Entropy Loss
2. **Root Mean Squared Error (RMSE) Related:**
   - Root Mean Square Error (RMSE)
   - RMSE
   - RMSE and MAPE
   - RMSE, MAE, MAPE, Theil's U (U1, U2)
3. **Cross-Entropy Related:**
   - Cross-entropy loss

- Cross-Entropy
- Binary Cross-Entropy
- Categorical Crossentropy
- Cross-entropy
- Cross-Entropy Loss
- Softmax loss function

4. **Mean Absolute Percentage Error (MAPE) Related:**
   - MAPE
   - Mean Absolute Percentage Error (MAPE), Directional Accuracy (DA), Theil's U, Average Relative Variance (ARV)
5. **Sharpe Ratio Related:**
   - Sharpe Ratio
   - Differential Sharpe Ratio
   - Sharpe Ratio and Mean Squared Drawdown (MSDD)
   - Sharpe Ratio Maximization
6. **Other Common Loss Functions:**
   - Accuracy
   - Classification Error
   - Negative Log-Likelihood
   - Percentage Error
   - ?-insensitive Loss Function (?-ILF)
   - ?-insensitive loss function
7. **Specialized/Custom Loss Functions:**
   - Cost function with a regularization term
   - Arctangent Cost Function
   - Structural loss
   - Reward function and temporal difference error for DDQN, clipped objective function for PPO
   - Combination of loss functions for actor and critic networks
   - Optimization criterion based on annualized rate of return, annualized standard deviation, and maximum drawdown
   - Minimization of the smallest singular vector
   - Profitability metrics (e.g., return on investment)
   - Quadratic criterion
   - Return on Investment (ROI)
   - Wasserstein distance with Gradient Penalty

A.8 The best models with explanation and grouped

1. **Deep Learning Models**

- Integrated CNN and Deep Learning: Integrated CNN with higher prediction accuracy and cumulative yield.
- DeepLearninH2gO Agent: Outperforms MLP and B&H.
- Deep Neural Network (DNN):
  - Best performing with 5 hidden layers and sliding window size of 3 minutes.
  - Uses a neural network ensemble to predict stock returns.
  - Found to be the best model for predicting financial market movement directions.
  - Using Stacked Denoising Autoencoders (SdAE) outperformed other models.
  - Outperforms traditional models with an out-of-sample Sharpe ratio of 2.6.
  - With a small window size showed the highest directional accuracy and profitability.
  - Showed superior performance in predicting the direction of financial market movements, achieving up to 68% accuracy.
- TABL (Temporal Attention-Augmented Bilinear Network): Outperforms CNN and LSTM.
- CNN with GAF Mapping: Achieved the highest accuracy.
- CNN-TA: Outperformed Buy & Hold, RSI, SMA, LSTM, and MLP regression models.
- VLSTM: Outperforms vanilla LSTM, LSTM with attention, multi-scale LSTM, and MLP in terms of mean F1 score.
- DLNN: Outperformed the ZIP trading algorithm in live trading tests.
- Deep Learning Model:
  - Outperforms traditional machine learning methods such as ARIMA and SVM.
  - Particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks outperformed traditional models.
- VAE-LSTM: Demonstrated the highest prediction accuracy and rate of return.
- Autoencoder Network: Performed best in terms of profitability in trading simulations.
- AANN with Bagging Approach: Showed better performance in terms of trading profits and detection accuracy compared to a buy-and-hold strategy.
- Dual Deep Learning Agents: Showed superior performance in option pricing and bid-ask spread estimation.
- UFCNN: Showed superior performance in time-series modeling tasks.
- CNN-LSTM: Showed superior performance in terms of an annualized rate of return and maximum retracement.
- Deep Reinforcement Learning with Sentiment Analysis: Showed superior performance in profitability and risk-adjusted returns.

2. **Neural Networks (NN)**
   - General Neural Network: Superior returns compared to traditional value strategies.
   - Neural Network with Specific Topologies:

  - Three hidden layers (4, 8, 4 nodes) with inverse tangent and sigmoid transfer functions.
  - ANN with topology [6, 10, 1]: 6 and 10 neurons in the first two hidden layers, 1 in output.
  - Neural Network (15 hidden neurons): Inputs include stochastic oscillator %K %D, MACD, RSI index, and backward regressions for 5 and 10 days.
- MACD Crossover Neural Network: Outperformed TIPP and ANN models in various scenarios.
- Feedforward Neural Network (FFNN): Chosen for universal representation capabilities and fast prediction.
- Back-Propagation Neural Network (BPNN):
  - Outperforms the Genetic Programming model.
  - Showed superior performance in terms of forecasting accuracy and profitability for inter-commodity spread trading.
- BPN2 (Backpropagation Neural Network with architecture 5-3-3-1): Best performance compared to other models (BPN1, BPN3, MR).
- Artificial Neural Network (ANN):
  - Based on fundamental analysis (FA) concepts for higher prediction accuracy.
  - Outperformed the ARMA model in some cases.
- Self-evolving Trading Strategy Based on BP Neural Network: Outperforms classical strategies in terms of yield and risk management.
- Neural Network-Based Framework:
  - Demonstrated better performance in predicting profitable trading actions.
  - Outperforms traditional moving averages and other statistical measures.
- FNN with Reduced Complexity Encoding: Showed the best performance in terms of profitability and trading efficiency.
- ANFIS: Showed superior performance in terms of Profit Factor, ROI, Sharpe Ratio, and Sortino Ratio.
- ANFIS with Active Investment Strategy: Shows the best performance in stock price prediction.
- ANFIS-RL: Showed superior performance in terms of predictive accuracy and trading profitability.
- AANN: Showed superior performance in detecting trends and generating profitable trading signals.
- Red Ward Neural Network: Superior performance in predicting weekly profitability.

3. **Reinforcement Learning (RL)**
   - Lipschitz Extension-based RL: Highlighted for performance compared to neural networks.
   - Deep Q-learning Networks (DQN):

  - Outperformed classical time-series momentum strategies.
  - Outperforms other models in terms of profitability and stability in the stock market investment strategy.

- Time-driven Feature-aware Jointly Deep Reinforcement Learning (TFJ-DRL): Outperforms other models in terms of total profits and Sharpe ratio.
- Fitted Q Iteration with Extra-Trees Regressor: Outperforms basic Q-learning algorithm in handling continuous state and action spaces.
- Dynamic Q EKF with ANN: EKF model with dynamically set Q parameter using ANN outperformed constant Q EKF model.
- DRL (DQN and A3C): Found to outperform traditional methods and other machine learning models in terms of risk-adjusted returns.
- PPO: Showed better convergence and robustness compared to DDQN.
- Asynchronous Advantage Actor-Critic (A3C) Method: Demonstrated a stable winning strategy with high profitability and risk management.
- Deep Reinforcement Learning with Sentiment Analysis: Showed superior performance in profitability and risk-adjusted returns.
- AlphaStock: Showed superior performance in terms of risk-adjusted returns, adaptability to diverse market states, and control of extreme losses.

4. **Traditional Machine Learning Models**

- Distance-based Model: Feature-weighted Euclidean distance to the centroid of a training cluster.
- Random Forest:
  - Adaptability to non-stationary time series data.
  - Produced the most accurate forecasts and highest abnormal returns.
  - Showed the best performance in predicting stock price movements.
  - Combined with boosting algorithms, showed superior performance in detecting economic turning points.
- Decision Trees (ID3 Algorithm): Constructed based on financial indicators.
- Logistic Regression: Outperformed buy-and-hold and dual momentum strategies.
- Multi-classifier System: kNN, Logistic Regression, Naive Bayes, Decision Tree, SVM with genetic algorithms.
- CHAID: Best prediction accuracy of 85.64%.
- Linear Regression (LR): Outperforms Support Vector Regression (SVR) in short-term prediction.
- Heuristic Forecasting Model (HFM): Outperformed buy-and-hold strategy and non-heuristic forecasting model.
- Manifold Learning: Found to yield promising results for FX forecasting.
- Cooperative Learning Model: Group knowledge refinement learning model (combination of XCS and neural network).
- Extended Hill Climbing (EHC): Effective with lower computation time compared to Exhaustive Search.
- Principal Component Regression (PCR): Highest predictive performance, best hit ratio, and R^2_OS.

- Kernel Price Pattern Trading (KPPT) System: Utilizes a kernel-based approach to predict price patterns.
- Zero-Truncated Poisson Mixture Model (ZTP): Outperforms Poisson and Negative Binomial mixture models.
- Incremental SVR: Outperforms batch-mode and individual experts.
- Differential Evolution Method (DEM): The best balance between computation time and strategy performance.
- Gradient Boosting Decision Tree (GBDT): Combined with multi-view feature construction showed superior performance.
- Ensemble of SVR Models: Showed superior performance in terms of risk-adjusted returns and profitability.

5. **Support Vector Machine (SVM) Models**

- PCA-WSVM: Outperforms WSVM, PCA-ANN, and BHS.
- PLR–FW-WSVM: Outperformed PLR–WSVM and PLR–ANN.
- Support Vector Machine (SVM):
  - Radial basis function (RBF) kernel outperformed logistic regression.
  - Integrated with GARCH and VPIN, effective in predicting market liquidity and returns.
  - With Polynomial Kernel: Showed the highest accuracy and returns.
  - With Radial Basis Function Kernel: Superior performance in volatility forecasting compared to traditional GARCH models.
  - Based Strategy: Showed competitive performance but did not outperform the equally weighted portfolio strategy (EqW).
- ABC-ANFIS-SVM: The hybrid model showed superior performance in terms of accuracy and quality.

6. **Rough Sets**

- Rough Sets with LEM2 Algorithm: Outperforms other methods in accuracy and fewer attributes.
- Rough Set Analysis: Used to generate trading rules.
- Rough Set-Based Rule Generation: High return rates with trend coordination.
- Rough Set-Based Real-time Rule-Based Trading System (RRTS): Chosen as the best model.
- Rough Set-Based Rule Extraction: Higher return rates compared to traditional technical analysis methods.
- S-Rough Sets: More effective than Z. Pawlak rough sets in dynamic information recognition.
- Rough Sets Classifier: Chosen as the best model due to handling vagueness, uncertainty, and incomplete data.

7. **Recurrent Neural Networks and extensions**

- NARX Network: Outperforms SVM when combined with ICA.
- SFM Network: Outperforms AR and LSTM.

- CR (Candlestick-based RRL): Outperforms basic RRL, ZI, and BH models.
- VG-RAM WNN: Outperformed ARNN predictors in computational efficiency.
- SAF-ARC-MMSGD: Outperforms other models in terms of convergence speed and robustness against impulsive noise.
- DeepMTA: Outperforms Logistic Regression, Hidden Markov Model, and Dual-Attention RNN.
- LSTM with OSTSC: Improved performance metrics compared to the model without oversampling.
- RF-WMGEPSVM: Outperforms other strategies in terms of ROR, MDD, and PP across various market scenarios.
- EMD-ELM-PLUS: Outperforms EMD-ELM-ELM and single models in terms of RMSE, MAPE, and DA.
- PMTS with DTW: Outperformed other methods in terms of trading profitability and stability.
- FA-FFLANN with RLS: Outperforms other models in terms of MAPE, DA, Theil's U, and ARV.
- WGAN-GP: Superior performance in generating realistic financial time series.
- PLR-IRF and DRNN-Based Model: Showed higher prediction accuracy and lower critical error rate.

8. **Ensemble Models**

   - Ensemble Model: Combines predictions of multiple models (SVM, decision trees, neural networks).
   - Hybrid Model: Combining PNN, rough sets, and C4.5 decision tree.
   - Random Forest and Boosting Algorithms: Superior performance in detecting economic turning points.

9. **Hybrid and Composite Models**

   - PLR–FW-WSVM: Outperformed PLR–WSVM and PLR–ANN in accuracy and profit.
   - Multi-classifier System: kNN, LR, NB, DT, SVM with genetic algorithms.
   - MACD crossover neural network: Outperformed TIPP and ANN in various scenarios.
   - DeepMTA: Outperforms Logistic Regression, Hidden Markov Model, and Dual-Attention RNN.
   - GABPN: Outperforms BPN and multiple regression models.
   - Hybrid Fuzzy Inference System (HyFIS): Achieved the highest hit ratio and best cumulative wealth performance.
   - ABC-ANFIS-SVM: The hybrid model showed superior performance in terms of accuracy and quality.
   - MLP: Showed superior performance in prediction accuracy and profitability.

10. **Specialized Models**

    - Models with Negative Coefficients and Small Intercepts: Perform well in terms of profitability and hit ratios.

- Polynomial Solver: Shows promising results in minimizing prediction error.
- eFSM-Based Straddle Trading System: Superior performance in volatility prediction and trading profitability.
- Machine Learning-Based Synthetic Data Generation: More effective in addressing issues of small data and outliers.
- Machine Learning-Based Trading Algorithms: Outperform traditional models and human traders.
- AML-Based Strategy: Superior performance in the probability of correct selection and efficiency.
- ELM-SVR Combined with Kalman Filter: Showed the best performance in terms of annualized returns, Sharpe ratio, and reduced volatility.
- WiSARD: Improved trading performance with higher win ratios and expectancies.
- k-NN with Fuzzy Candlestick Patterns: Showed superior performance in predicting future market behavior.
- SOM-Based Strategy: Showed superior performance in profitability and accuracy of trading signals.

11. **Others**

- Not applicable: The concept of the best model does not apply.

References

Bao, Y., Deng, Z., Wang, Y., Kim, H., Armengol, V.D., Acevedo, F., Ouardaoui, N., Wang, C., Parmigiani, G., Barzilay, R., Braun, D., Hughes, K.S.: Using machine learning and natural language processing to review and classify the medical literature on cancer susceptibility genes. JCO Clin Cancer Inform **3**, 1–9 (2019) https://doi.org/10.1200/cci.19.00042

Cachola, I., Lo, K., Cohan, A., Weld, D.S.: Tldr: Extreme summarization of scientific documents. CoRR **abs/2004.15011** (2020) 2004.15011

Dowling, M.M., Lucey, B.M.: Chatgpt for (finance) research: The bananarama conjecture (2023)

Fire, M., Guestrin, C.: Over-optimization of academic publishing metrics: observing goodhart's law in action. GigaScience **8**(6) (2019) https://doi.org/10.1093/gigascience/giz053

Ferreira, F.G.D.C., Gandomi, A.H., Cardoso, R.T.N.: Artificial intelligence applied to stock market trading: A review. IEEE Access **9**, 30898–30917 (2021) https://doi.org/10.1109/ACCESS.2021.3058133

Grootendorst, M.: Bertopic: Neural topic modeling with a class-based tf-idf procedure. arXiv preprint arXiv:2203.05794 (2022)

Garcia, J., Villavicencio, G., Altimiras, F., Crawford, B., Soto, R., Minatogawa, V., Franco, M., Martínez-Muñoz, D., Yepes, V.: Machine learning techniques applied to construction: A hybrid bibliometric analysis of advances and future directions. Automation in Construction **142**, 104532 (2022) https://doi.org/10.1016/j.autcon.2022.104532

Hewamalage, H., Ackermann, K., Bergmeir, C.: Forecast evaluation for data scientists: common pitfalls and best practices. Data Mining and Knowledge Discovery **37**(2), 788–832 (2023) https://doi.org/10.1007/s10618-022-00894-5

Hong, Z., Ajith, A., Pauloski, G., Duede, E., Malamud, C., Magoulas, R., Chard, K., Foster, I.: Scholarbert: Bigger is not always better. arXiv (2022) https://doi.org/10.48550/arxiv.2205.11342

Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., Soatto, S.: Rethinking the hyperparameters for fine-tuning. arXiv pre-print server (2020) https://doi.org/Nonearxiv:2002.11770 2002.11770

Li, J., Cheng, X., Zhao, W.X., Nie, J.-Y., Wen, J.-R.: Halueval: A large-scale hallucination evaluation benchmark for large language models. arXiv pre-print server (2023) arXiv:2305.11747 [cs.CL]

Lo, K., Wang, L.L., Neumann, M., Kinney, R., Weld, D.: S2ORC: The semantic

scholar open research corpus. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4969–4983. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.447 . https://aclanthology.org/2020.acl-main.447

McInnes, L., Healy, J., Astels, S.: hdbscan: Hierarchical density based clustering. The Journal of Open Source Software **2**(11), 205 (2017) https://doi.org/10.21105/joss.00205

McInnes, L., Healy, J., Saul, N., Großberger, L.: Umap: Uniform manifold approximation and projection. Journal of Open Source Software **3**(29), 861 (2018) https://doi.org/10.21105/joss.00861

Marshall, I.J., Wallace, B.C.: Toward systematic review automation: a practical guide to using machine learning tools in research synthesis. Systematic Reviews **8**(1), 163 (2019) https://doi.org/10.1186/s13643-019-1074-9

Probst, P., Bischl, B., Boulesteix, A.-L.: Tunability: Importance of hyperparameters of machine learning algorithms. arXiv pre-print server (2018) https://doi.org/Nonearxiv:1802.09596

Pintas, J.T., Fernandes, L.A.F., Garcia, A.C.B.: Feature selection methods for text classification: a systematic literature review. Artificial Intelligence Review **54**(8), 6149–6200 (2021) https://doi.org/10.1007/s10462-021-09970-6

Tu, S., Li, C., Yu, J., Wang, X., Hou, L., Li, J.: Chatlog: Carefully evaluating the evolution of chatgpt across time. arXiv pre-print server (2024) arXiv:2304.14106 [cs.CL]

Tom, Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Daniel, Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. arXiv pre-print server (2020) https://doi.org/10.48550/arXiv.2005.14165

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. arXiv pre-print server (2020) https://doi.org/Nonearxiv:2002.10957

Yu, Y.-X., Gong, H.-P., Liu, H.-C., Mou, X.: Knowledge representation and reasoning using fuzzy petri nets: a literature review and bibliometric analysis. Artificial Intelligence Review (2022) https://doi.org/10.1007/s10462-022-10312-3

University of Warsaw
Faculty of Economic Sciences
44/50 Długa St.
00-241 Warsaw
www.wne.uw.edu.pl
ISSN 2957-0506