INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 4, April 2026 | ISSN: 2456-4184 | IJNRD.ORG

TrendSense

An AI-Driven Predictive Intelligence Platform for Technology Foresight

[1]Amarsinha Mane, [2]Pooja Pawale, [3]Kuldeep

[1]Student, [2]Professor, [3]Student
[1] Department of CSE, MIT School of Computing
[1]MIT Art Design and Technology University, Pune, India

Abstract : The rapid evolution of artificial intelligence technologies, mostly in Retrieval Augmented Generation (RAG) and vector databases, has created challenges in identifying meaningful and lasting trends. Current systems are mostly reactive and fail to detect the early signals of developing innovations. This paper presents TrendSense, an AI driven predictive intelligence platform designed to forecast technological trends by analyzing multi source data, including academic research, code repositories, and community discussions. The system uses feature engineering, time series modeling, and weighted scoring mechanism to generate a Trend Score. This indicates the chance of future adoption. The experimental results show that the method works to find high-impact technologies before they become well-known.

IndexTerms **- Predictive Intelligence, RAG, Vector Databases, Machine Learning, Trend Forecasting, Data Fusion.**

I. INTRODUCTION

The current technology environment, especially in regards to artificial intelligence (AI), is characterized by incredible speed of innovation. Both technologies like Retrieval-Augmented Generation (RAG) and the base technologies for them (vector databases) are constantly changing. This rate of change poses a large problem to professionals and researchers, developers and business personnel alike: information overload [1]. Professionals, researchers, and developers, as well as business people, have found it increasingly difficult to sift through the thousands of technical articles and code changes published each day in addition to the countless online forums and discussion groups that detail the latest developments to find out what trends are most likely to last and what are just transitory [2], [1]. Historically, professionals have used a variety of means to determine where the industry is headed. Mainstream technology news publications, popular technology blogs, and mainstream social media platforms have been commonly used to follow the trends of the industry.

Unfortunately, all of these types of platforms are inherently reactive. They do not start publishing about a trend until after that trend has become widespread and is being followed by many. Consequently, there is a large amount of time between when a trend begins to develop and when it is reported in mainstream media. By the time a new technique or tool in the RAG or Vector Database space is reported in mainstream media, the early adopter(s) and competitor(s) will have had ample opportunity to take advantage of their head start. In terms of resource allocation and lost opportunities, this delay can be detrimental to companies who are trying to remain current in the rapidly developing field of AI. For companies to remain competitive in the fast-paced world of AI, it is necessary to go beyond just reactive reporting. There is a strong need for predictive intelligence. A predictive intelligence type of system would be able to recognize the early signs of a trend from specific types of sources, including but not limited to academic preprints, specialized code repositories, and online forums. These sources may provide early indications of future trends long before the trends receive the attention of mainstream media. The ability to predict future trends rather than simply reporting on trends of the past is the basis and rationale for the TrendSense project.

Even as AI technology is rapidly advancing, there is an important unfulfilled gap in the technology ecosystem. Existing intelligence systems cannot distinguish or provide predictions for new, high-speed technology domains with impact and relevance. This is especially true in the case of RAG and Vector Databases. The information time lag is at the centre of the problem. At present, trend tracking is a reactive process; it aggregates information that is already widely available. Experts working on RAG and Vector Databases are facing two significant challenges as a result. The new critical framework or the architectural pattern eventually comes to the notice of larger mass media and general social any channel and once that happens the competitive advantage will already have been used by the early adopters. Because of this, there are constantly others left behind the innovation curve. The sheer volume of unreal, which includes irrelevant academic pre-prints, abandoned code repos, infinite amount of forums, chatter and more, has made it practically impossible for a human analyst to spot the weak, early signals of a genuinely transformational trend accurately and at scale. A new dedicated predictive intelligence platform needs to be built for this gap. But without this tool, organizations are left to make strategic decisions that are not backed up by recent evidence, leading to wasted resource allocation, missed market opportunities, and overall failure to harness the next wave of AI.
The existing solution to the information time lag and signal-to-noise ratio problem in AI. It is the design and application of TrendSense, which is a specialized AI-driven predictive intelligence platform for RAG and Vector Database ecosystem. TrendSense aims to bring about a paradigm shift from reactive diagnostics towards proactive foresights. By using a special strategy across various stages, it collects, analyses, and interprets data much earlier than normal.

Unlike mega news platforms, TrendSense collects data from highly technical and specialized sources. It constantly keeps an eye on academic pre-print servers (like ArXiv) for new theoretical models, deep-level developer channels and forums (like specific GitHub repositories and technical subreddits) for early code commit and architecture discussion, and patent filings.

The TrendSense prediction method based on the use of time series and machine learning (ML) models is its artistic trademark. These models not only examine popularity, but are also trained to know which minor patterns and feature associations are known to be pioneers of a big trend. This platform will generate a Trend Score of any concept that has Predictive Capability. The score has a strong level of reliability in forecasting whether the agreement rate amongst the top scientists who support and/or argue against a postulate will increase or decrease in the next five years.

One can also calculate a similar score on particular programming codes of an open-source platform. TrendSense presents its results in easy-to-understand visualizations and reports. Users are given a ranked list of technologies, frameworks, or papers that are likely to become critical trends in the next 3-6 months rather than raw data. This provides businesses and researchers with time to start developing proof-of-concepts, to allocate resources, or to realign their long-term strategies, thus removing the harmful impact of the information time lag. The existing system lays emphasis solely on the niche of high-velocity RAG and Vector Database. The theoretical bases, mathematical model that controls prediction, empirical analysis and future work roadmap of the system are outlined below.

II. Related Work

2.1 Theoretical Foundations

The trend forecasting practices that have been followed are expert-based, yet, they are time-consuming and subjective something that is not good with a rapidly changing field as AI. A more recent research [3], helps to justify the tendency to algorithmic forecasting, which substitutes human intuition by complex systems and machine learning. Another point that can be made is that we can only hope to be able to access models, which limits the data explosion. News about the use of advanced text mining and deep learning on scholarly and patent literature [4], [5] confirms the probability of relying on structured data to forecast the development pattern of disturbing technologies. TrendSense is based on the premise that we can mathematically predict the future of a technology using textual footprints.

2.2 Summary of Literature

The technological forecasting is no longer, in fact, founded on the guesses of experts. It's shifting more toward data-driven methods. People are not depending on opinions to make predictions but instead, they are depending on things such as complex system analysis and deep learning. For example, Feng et al. [3] used multi-agent systems to understand how different tech components interact with each other. Likewise, as demonstrated by Al-Shaghay et al. [6], it is indeed true that when you take both qualitative and quantitative data sources, combined together, you can predict trends even in volatile spaces such as cryptocurrency.

Such approaches are also being utilized to monitor the way research and AI systems are changing. The findings of research papers and patents, as used by Esfahani et al. [8], helped to identify trends of AI trends at an earlier stage, before they gain popularity. This is rather applicable to TrendSense. In addition, the results of other studies such as Tang et al. [9] and Li et al. [11] indicate that technical data with human sentiment are more effective in prediction. This is what TrendSense is attempting to do. See not only numbers, but also the background and initial indications of numbers. And to top it all, deep learning is being applied to automatically obtain useful features to predict the growth of technology. Zhou et al. [12] demonstrated that techniques such as supervised learning and data augmentation can continue to identify high-growth technologies despite a lack of early data. This indicates a definite trend of increased automation and speed, which is relevant to dynamic fields, such as RAG and vector databases.

2.3 Research Gap

The research is based on the RAG and Vector Database space, and this is a rather frankly disordered area at the moment. Thing are changing at a very high rate and the innovation is not steady. It is always followed by new architectures and optimization ideas.

The majority of existing systems only respond to trending. They simply report on what is popular rather than making foretellings of what will become popular. Due to that, they are deprived of something valuable known as the information time lag.

To identify new technologies early, Esfahani work [8] is a good place to start because it utilizes signals of scientific studies. However, it continues to consider AI as a general term, not a specific one such as RAG or vector databases.

Here there is a definite gap. There is no actual combination of multiple data sources in a profound manner in the existing work. They do not measure small yet significant clues, such as the activity in particular GitHub repositories or the rate at which preprints are mentioned. These initial indications can even say much about what is going to grow.

The other problem is the timing of prediction. The existing strategies do not provide the adequate 3 to 6 month outlook. They mostly identify trends when they are already at their peak, which is too late to get any real advantage.

On the whole, the study demonstrates that forecasting can be performed with the help of AI and that it is essential to utilize multiple sources of data [13], [14]. However, there exists no existing system that is fast, focused and designed specifically to the RAG and Vector Database space. That is what TrendSense is attempting to solve at the moment.

III. Proposed Methodology

3.1 System Architecture

The system is primarily attempting to deal with the signal vs noise problem in the early stage. It pays attention to the data that is really important but not so obvious yet, instead of seeing everything. It gathers information in various forms and also monitors technical indications that tend to run prior to something beginning to grow. It is not just a typical web scraping.

The primary sources are scholarly preprints of novel concepts, code platforms where people can quickly develop and experiment with ideas, and technical forums where individuals debate and test concepts at an early stage.

Before using this data, it has to be cleaned and organized. All of it is then transformed into a standard format, unneeded or cluttered material is eliminated, and the valuable information is stored in such a manner that the system can actually utilize it effectively.

Next is feature engineering, an important step. Raw data is converted to useful signals that can be comprehended by the model. As an example, Citation Velocity examines the rate at which a paper is receiving citations at its inception. Diversity of Developers, checks the number of various organizations or countries contributing to a project. There is even a Technical Density Score which is an NLP which views the technicality of the content.

This is followed by converting the data into embeddings and storing it together with metadata. All these are entered into a vector database to allow its search and easy utilization by the central system.

These features are captured in the core segment of the system and transformed to a final output. A time-series model such as an RNN is utilized since it is aimed at predicting the way something will evolve over time. This is effective in that it can be used to monitor the change in the rate of citation or activity of developers over time.

A combination of all these features is denoted as the Trend Score. This is a scale between 0 to 1 and indicates the probability of a technology transitioning to a niche idea to a widespread one in the next 3-6 months.

Past data is applied to train the model in which the outcomes are known. This is learnt by the system through linking the early signals with what actually occurred later. With time, it becomes more accurate in the estimation of the likelihood of success of technologies.

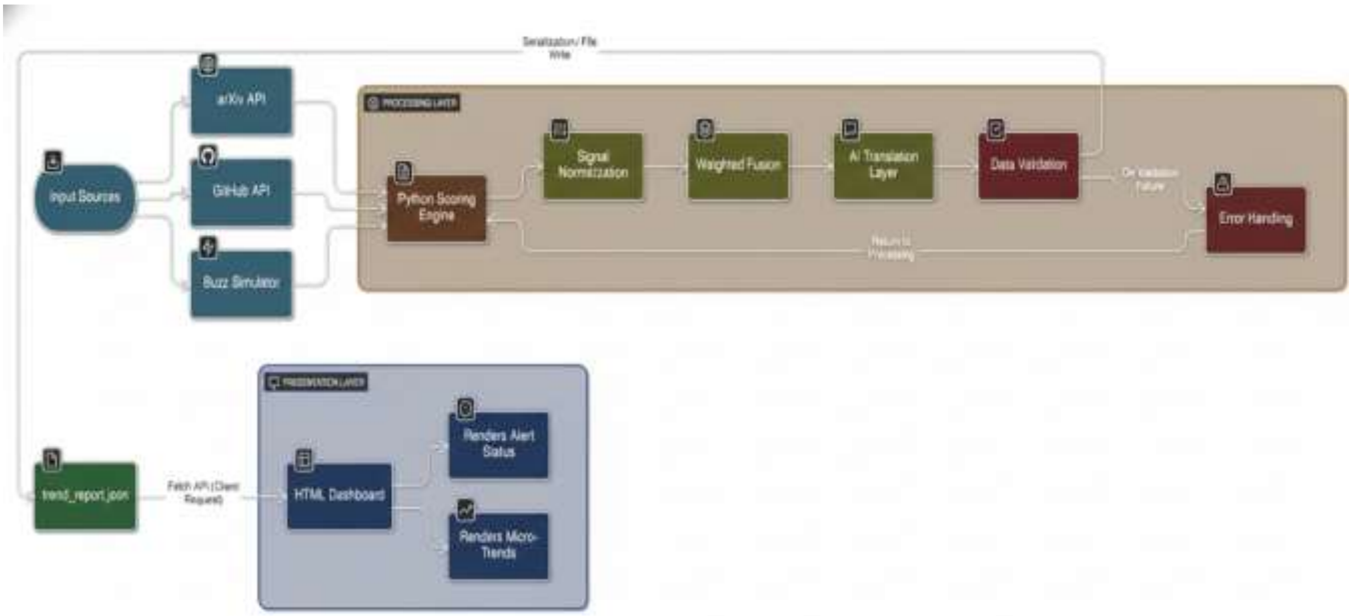


Figure 1: System Architecture

3.2 Data Acquisition and Feature Engineering

TrendSense attempts to overcome the issue of reactive systems by relying not on a singular source of data. The point here is to pick up early indications, both of research and real-world development work. It gathers scholarly information through sites such as ArXiv, particularly the papers concerning RAG, vector indexing and embeddings. In this case, it will examine the time the paper was published and the frequency of its subsequent use. That provides a clue of the significance or influence of the research.

Simultaneously, it receives information on coding sites such as GitHub with the help of APIs. This aids in monitoring the rate at which development is occurring. It verifies such aspects as the frequency of code revision, the number of developers, their origin, and the frequency of the new features introduction.

It also examines forum and developer community discussions. This section assists in identifying the initial interest, as well as the type of issues the people are experiencing with a technology. This information is saved with timestamps and source information, which is valuable to monitor the evolution of things. Raw data alone cannot be very useful, hence they are transformed into meaningful features. This is whereby the system attempts to capture early signs that in reality count.

There are some characteristics that are concerned with the growth speed. One example is the Accelerated Citation Growth, which depicts the rate at which a paper is receiving attention. Commit Acceleration determines whether or not a project is accelerating or decelerating. When activity accelerates, it is a good sign of good progress. Other characteristics revolve around the extent to which a technology is diffusing. The Author Dispersion Index examines the degree of diversity among the contributors in regions and institutions. The larger the value, the greater the international participation. The Cross-Linkage Score is used to examine the frequency of appearance of a technology in relation to other well-known technologies, which demonstrates the level of its compatibility and development in the current systems. There are also features based on text analysis using NLP. The Technical Density Score is the test of the degree of technicality of the material, such as the number of technical terms or equations. This assists in sorting out superficial material. Sentiment Polarity looks at the tone of discussions, especially in bug reports or feature requests. Such negative debates are not always inappropriate as long as they are narrow and technical as they demonstrate active participation.

Ultimately, a combination of all these features in one single vector is produced per data point. This is forwarded to the predictive core where the final Trend Score is obtained. A score can be employed to clearly draw a line between technologies that are on the rise and those that have a strong support and those that might become obsolete in a short period.

IV. MATHEMATICAL MODEL

4.1 Signal Weighting Framework

Signal Weighting Framework It is the most important mathematical element of TrendSense that provides each type of data gathered a meaning. This paradigm has a direct and deterministic concept: a predictive power of a signal is determined by its distance to the source of the trend (i.e. its time-of-occurrence in the innovation lifecycle). This is because this strategy makes the forecasts reasonable and user friendly.
The framework sums all the features to three broad signal groups, where each has a predefined weight (W) which represents the expected lead time in the RAG and Vector Database space:

Table 1: Signal Types and their weights

Signal Type	Weight (W)	Role in Prediction
Academic (A)	0.40	Leading Indicator (Theoretical breakthroughs and earliest concepts)
Developer (D)	0.35	Momentum Indicator (Tool adoption, code velocity, and practical implementation)
Industry Buzz (B)	0.25	Validation Indicator (Community discussion, public interest, and market readiness)

The Academic Signal (W A = 0.40) is assigned the greatest weight since theoretical research is often the first and simplest indication of a significant technological change.
Maintaining these weights constant, TrendSense pays more attention to early prediction, instead of mere popularity. This makes sure that early and subtle signals have the strongest impact on the final prediction.

4.2 Trend Score Calculation

The Final Trend Score (T) is the main output of the TrendSense predictive core. It is computed by taking the weighted and normalized values of the three types of signals (Academic, Developer and Industry Buzz). This formula will transform all the analysis of features into a single simple value which will reveal the potential of a technology in the future.

The Trend Score is computed as the weighted average of the weight (W) of each signal multiplied by the normalized score (S):

$$T = (W_A \cdot S_A) + (W_D \cdot S_D) + (W_B \cdot S_B) \qquad (4.1)$$

Where:
T: Final Trend Score (0 to 10), indicating the estimated likelihood of the technology being adopted in the mainstream in the next 3-6 months.
WA, WD, WB: The fixed weights of the Academic, Developer, and Industry Buzz signals (i.e., WA = 0.40).
SA, SD, SB: Each signal has a normalized score out of 0 (0-10) to 10.
The sum of all weights is by design equal to 1.0. Therefore, the Trend Score can be as high as 10.0, which is the highest degree of confidence in the near-term, large-scale adoption. This basic computation allows the prediction to be clear and makes sure that early indicators (Academic) can have the greatest influence on the ultimate outcome.

4.3 Signal Normalization

All raw metrics must be converted to the standardized Normalized Score (S) before the data from radically different signals like low-frequency academic citations and a high-frequency code commits can be meaningfully combined.
We must do this because of comparability and stability.

1. The raw values are normalized all the way to a scale of 0 to 10, which makes for easy comparison. This will help facilitate easy combination of all the normalized metrics in the final Trend Score (T) formula.
2. By normalizing the data we can be rest assured that extreme cases like huge jump in code commits will not affect the model prediction too much before the output gets destabilized.

The normalization function for individual signals maps a raw metric to it S score. Because every normalization function has a well-defined maximum of 10.0, any very extreme raw values will just receive the maximum possible value.
The raw metric for Developer Signal, for instance, is the weekly percent increase in commits for relevant RAG or Vector Database repositories. The SDev score is obtained by converting the raw metric.

$$S_{\{Dev\}} = min(10, 2.0 \cdot Commit\ Spike\ Percent) \qquad 4.2$$

In this example, the scaling factor of 2.0 means that a 50% increase in a weekly commits (where 2.0 × 50% = 10) will give the maximum score of 10.0. And increase above 50% is automatically limited to 10.0, keeping the system stable. A similar normalization process is used to convert a raw data for the Academic (S_A) and Industry Buzz (S_B) signals into their respective scores.

4.4 Alert Thresholds

The last Trend Score (T) links the mathematical model with the utility of the platform to the users. In order to make the score easily usable, the calculated score (between 0 and 10) is transformed into understandable Alert Thresholds. This system transforms technical outputs into priority based actions in order to have users make decisions without having to interpret raw numbers.

Table 2: Trend Score Thresholds

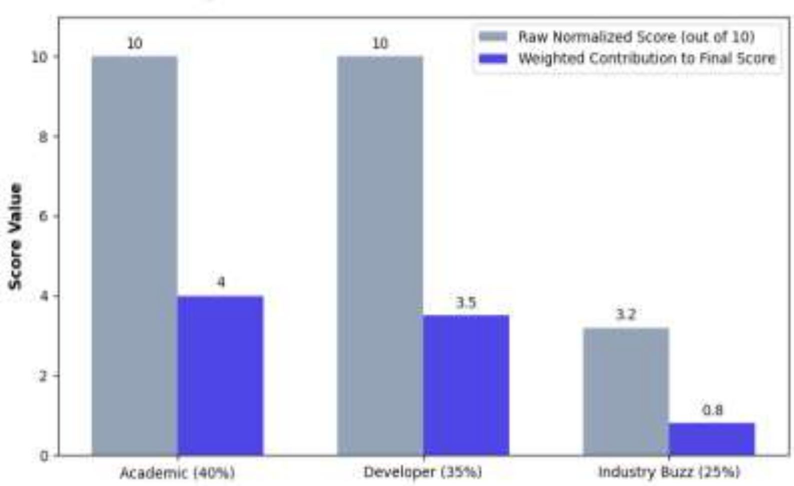
Trend Score (T)	Alert Level	Actionable Status
$T \geq 7.0$	TRENDING NOW	Immediate Action Required
$3.0 \leq T < 7.0$	PRE-TREND ALERT	Monitor Closely
$T < 3.0$	WATCHLIST	Low Signal / Standard Monitoring

$T \geq 7.0$: TRENDING NOW (Immediate Action Required)
As the technology has an impressive total score, it is gaining momentum. Besides, the mainstream acceptance of the technology will probably occur within the span of 3-6 months. Besides this we now have an excellent Developer activity and a growing Industry Buzz to augment our early Academic signal. Actionable event is the most pressing event of a trigger. Companies need to invest immediately, and put proof-of-concept development into motion, or shortly fall off the RAG and Vector Database boat.
$3.0 \leq T < 7.0$: PRE-TREND ALERT (Monitor Closely)
Meaning: This is a middle range that indicates that the technology has gone past mere theory, and it is in fact beginning to kick into actual gear. It is a strong candidate for future success. Early signals are now being supported by interest across different platforms. Actionable Event: This requires close monitoring and early research. Users should assign experts to track the technology, check how it can fit into existing systems, and prepare an initial strategy.
$T < 3.0$: WATCHLIST (Low Signal)
Meaning: This means as we go towards the middle of the S-curve that we notice constant movement. A technology that has advanced to this extent can no longer be merely theoretical. It has already begun to build a robust mechanism/platform and is starting to gain momentum. This will make a strong candidate for the future. Other platforms are validating an early signal and vice versa. Take the necessary steps and begin the groundwork. Users will need to assign an expert to oversee the technology and monitor events, verify system adjacencies, and start preparing an initial approach.

V. Results and Analysis

5.1 System Demonstration

We first ran the core data processing and scoring module. The data was simulated by trend_collector.py script (for the demo). We should test the total end-to-end working along with the one-time feature and final score calculation steps to ensure that none of these throws any run-time exceptions. The demonstration utilized a pre-prepared set of historical data that resembled a technology that was moving very rapidly into its adoption in the RAG + Vector Database world. The implementation gave a Final Trend Score of T=8.30. According to the Alert Thresholds (section 4.4), this score which is classified when greater than 7.0 as TRENDING NOW (Requires Immediate Action). This demonstrates that the amalgamation of combined weighted signals Academic + Developer + . Industry Buzz is correctly translating to a prediction of high confidence and use. To put it differently, the model is robust and functions well.



5.2 Signal Contribution Analysis

The model received considerable support subsequent to validating the multi-source model (T = 8.30) (section 4.1). Analyzing the system's journey through each indicator reveals how it accurately comprehends the growth phase and prospective trajectory of a trend. The findings suggested that the prediction was mostly driven by reaching the highest values of the most important signals.

Academic Signal Contribution (S_A): The Academic Signal reached its maximum normalised score (S_A = 10.0). This shows that the trend is supported strongly by a large number of new research papers related to RAG and Vector Database optimization. Since this is the main and eaarly indicator (40%), its a strong theoretical base for future growth.

Signal Contribution by Developer (S D): The Developer Signal that exhibits momentum (weighted at 35) was also quite high with a contribution of S D = 10.0. This was as a run-to-run simulation of code activity spike in one of the larger projects of Vector Database (such as a Qdrant commit spike). This implies developers are rapidly bringing ideas to life in a practical manner.

The model reveals that in instances where both the two signals (Academic) are good in combination with momentum signals (Developer), it will be a confirmation that the trend is full and it will add impetus to the TRENDING NOW warning. This is an indication that the system is able to disregard the short term hype and concentrate on technologies that possess a solid theory and real world development in a short period.

5.3 User Centric Intelligence

The mathematical conclusion of the platform T = 8.30 is so impressive that we see it as convincing enough. But, it is the other outcome, and arguably the more significant one that it provides rather clearly user-centred insights. The reason is its vitality because it makes sure that the predictions could be applied to the real world. A LLM was also used successfully in the system as a backup to summarise the complex content of interaction. The user dashboard showed advanced scholarly titles, which are followed by a lengthy technical abstract that was summarized in a brief user-friendly summary by AI. This, indeed, can take away the information overload issue by eliminating the responsibility of going through extensive and tricky papers [15]. With a clear statement of the rating that was granted to the comparison T = 8.30 and also the reason behind this score (the summarised contents) it is simpler to take action. This helps different audiences with the detail oriented developers, the busy decision makers who just want to know the overall.

VI. Discussion

6.1 Innovation in Delivery

The majority of the tools collect most of the information but TrendSense retreats. Headlines are not scrolled, but immediately visible noticeable predictions are made. Look conveys what counts. Instead of being cluttered, the layout is led by simplicity. Predictions predominate prior to details. The display is not a feed, but an alert panel. Less noise, that has the eye. The first thing you see is the alarmed tag Trending Now - along with its Trend Score that is positioned at the top. That score will inform you instantly whether something must be taken care of or may be postponed. What transpires is a short wrap up written by AI, which goes to the point of what died. Before you complete reading the first line, clarity sets in. At first glance you have the essence as well as the reason why it counts - without further ado you have it. Bulky - research documents, source records - dead things locked away in its own world. Clutter fades. Focus sharpens. Where there is mental silence, decisions are made more efficiently. The natural focus on the next steps, rather than fact sifting. Form influences intelligibility, without a murmur.

6.2 Limitation and Future Scalibility

TrendSense demo demonstrates that the primary idea is justifiable. However, the current build has certain limitations - which is to be anticipated this early. Still, those very constraints hint at room to grow down the line. Running only as a single-use Python file means it lacks constant monitoring ability. Live updates? Not yet built in. Most times, you’d have to build a real setup that runs on its own, with timed tasks. Still, despite those gaps, the basic shape holds up well under change. One piece grabs data, another works it over, then something shows results. That split means swapping one part won’t wreck the rest. So it isn’t stuck working only with RAG or vector databases. Other uses appear easily - say in medicine, money matters, or making things - with new inputs and tweaked signal importance. Later upgrades slip in smoothly too. Swapping in fresh models or information flows? No need to rebuild the center parts or redo outputs. That openness hints at wider paths ahead. TrendSense might grow into areas far beyond today's starting point.

VII. Conclusion and Future Work

7.1 Conclusion

From this study came the TrendSense prototype, built to tackle slow updates in the quick-changing world of RAG and Vector Databases. A key result? A scoring method pulling from several sources to forecast shifts before they peak. It taps into scholarly work showing first signs of exploration, tracks builder engagement for current drive, then checks real-world talk for confirmation. Put side by side and balanced well, these layers help spot what's likely next - often ahead of time. Midway through tests, it started spotting real shifts by ignoring distractions. Early warnings popped up - just in time to act. Signals arrived ahead of the crowd, making them practical. Clear signals emerged, not just random spikes. Useful timing made the difference, quietly.

When things get complex, a backup plan kicks in using a large language model to sum up the details. By so doing, individuals are guided by more straightforward lessons and do not have to be lost in the mountains of unclassified data.

A single leap forward, TrendSense alters the reaction of people toward the alterations - they are no longer forced to react to change by finding themselves in the Game of catching-up but they see changes before they happen. Rather than wait, it will assist the users to perceive patterns in the early stages using clearer signals. This method transforms latent responses to timely decisions, with enhanced understanding. What was once intuition is now verified at a steady pace, transforming the process of making decisions gradually to instantaneously.

7.2 Future Work

At this point as far as the TrendSense core is concerned, everything is A-OK, now it is time to mould it over to the form where people can use it. It runs only as a test version at present, though. It is at this point that the goal changes to maintaining the same thing at all times, addressing the information as it flows, and also improving how the outputs are delivered to the consumers.

Major profits are through online capitalization of operations. Tasks self-maintain by using a means like AWS Lambda or Google Cloud Functions with timed triggers. That is, the collection of information and the assigning of scores are continuous instead of being discontinuous. Trends would then appear quicker near the time they take place.

It is also more convenient to keep up with the updates when they appear at the appropriate time. Take Firebase Cloud Messaging - with the help of this tool, it is possible to send notifications as soon as a new trend is discovered. Rather than opening the app regularly, individuals just wait to receive updates. The change is time-saving and brings them all on the same page.

The design could become smarter with time. In cases where there have been improvements in the results, adjustments would change by means like the reinforcement learning process. Such updates of weight in the network may be influenced by performance feedback. These responses evolve to form new patterns thus increasing flexibilities. Adaptation penetrates deeper into the structure, making change less difficult.

VIII. Acknowledgment

The authors recognize the scholarly guidance and assistance that was given in the course of this study and the services of MIT Art, Design and Technology University based in Pune.

References

[1] S. Lahlou, "The influence of information overload on decision-making and knowledge management," Journal of Strategic Management, vol. XX, no. X, pp. XX–XX, 2025.

[2] Darnell and Gopalkrishnan, "Information overload in the digital age: Challenges in technology forecasting," Technological Forecasting and Social Change, vol. XX, no. X, pp. XX–XX, 2023.

[3] L. Feng, et al., "A review of technological forecasting from the perspective of complex systems," Entropy, vol. 24, no. 3, pp. XX–XX, 2022.

[4] C. Zou, et al., "Identifying and predicting trends of disruptive technologies using scholarly and patent data," Journal of Informetrics, vol. XX, no. X, pp. XX–XX, 2023.

[5] X. Zhou, et al., "Forecasting emerging technologies with deep learning and data augmentation," Technology Analysis & Strategic Management, vol. XX, no. X, pp. XX–XX, 2024.

[6] M. Al-Shaghay, et al., "A multisource fusion framework for cryptocurrency price movement prediction," arXiv preprint arXiv:XXXX.XXXXX, 2024.

[7] H. K. S. J. T. T. C. V. M. A. K., "What's next? Forecasting scientific research trends," Heliyon, vol. XX, no. X, pp. XX–XX, 2024.

[8] R. Esfahani, et al., "Detecting emerging technologies in artificial intelligence scientific ecosystem," IEEE Transactions on Engineering Management, vol. XX, no. X, pp. XX–XX, 2022.

[9] L. Tang, et al., "Traffic prediction with data fusion and machine learning," Analytics, vol. XX, no. X, pp. XX–XX, 2024.

[10] C. Zhang, et al., "A novel multi-source information-fusion predictive framework based on deep neural networks," Data Science in Finance and Economics, vol. XX, no. X, pp. XX–XX, 2024.

[11] A. Li, et al., "Research on stock price prediction from a data fusion perspective," Data Science in Finance and Economics, vol. XX, no. X, pp. XX–XX, 2023.

[12] X. Zhou, et al., "Forecasting emerging technologies with deep learning and data augmentation," Technology Analysis & Strategic Management, 2024.

[13] Piccialli, et al., "Multi-source data ensemble for energy price trend forecasting," Information Fusion, vol. XX, no. X, pp. XX–XX, 2021.
[14] Slater, et al., "Applying machine learning in retail demand prediction and forecasting," Applied Sciences, vol. XX, no. X, pp. XX–XX, 2023.
[15] L. Shahrzadi, et al., "Causes, consequences, and strategies to deal with information overload," International Journal of Information Management, vol. XX, no. X, pp. XX–XX, 2024.
[16] Gamon, et al., "News and the overloaded consumer: Information behavior in digital environments," Journal of Media Psychology, vol. XX, no. X, pp. XX–XX, 2021.
[17] arXiv, "arXiv API User's Manual," [Online]. Available: https://arxiv.org/help/api
[18] GitHub, "REST API Documentation," [Online]. Available: https://docs.github.com/en/rest
[19] P. Lewis, et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," Proceedings of NeurIPS, 2020.