УДК 004.85:316.472.4]:070.16
DOI: 10.31673/2412-9070.2025.027701

V. DANYLCHENKO[1], Ph.D., associate professor;
ORCID: 0009-0004-6839-2132
A. VOITKO[2], lecturer;
ORCID: 0009-0001-9521-0507
V. KUZMINYKH[2], Ph.D., associate professor;
ORCID: 0000-0002-8258-0816
V. KOLUMBET[2], senior lecturer,
ORCID: 0000-0002-0871-9402

[1] State University of Information and Communication Technologies, Kyiv
[2] National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»

MODERN CHALLENGES OF DISINFORMATION IN MEDIA: ANOMALY DETECTION IN SOCIAL NETWORK METRICS USING MACHINE LEARNING MODELS

The article considers the problem of detecting anomalies in the time series of media content metrics obtained from Telegram channels. The task of detecting anomalies is relevant in the context of combating disinformation and analyzing the dynamics of content distribution in social networks. Anomalies in metrics, such as the number of views, shares, comments, and reactions, may indicate manipulative actions, including the use of bots, falsification of reach, or the spread of disinformation.

A dataset obtained through the official Telegram API was used for the analysis. A feature of the data is the lack of retrospective values of metrics, which complicates the analysis of their dynamics. This problem was partially solved by collecting data at fixed time intervals after the publication of each post. The collected data was grouped by the channels of origin of the posts and time intervals after publication to ensure the accuracy of the analysis. Since the data were unlabeled, manual processing was used to remove outliers and ensure the reliability of the modeling.

The article analyzes the functionality of five Python libraries for detecting anomalies in time series: PyOD, TODS, PySAD, Darts, and Prophet. Their compliance with the requirements was assessed, in particular, with respect to working with time series, processing incomplete data, real-time support, seasonality, and computational efficiency. A comparison was made based on tables and graphs that demonstrate the results of using each library. In particular, the PyOD library is a well-known tool for detecting anomalies, but does not support direct work with time series. TODS has the potential to detect anomalies in streaming data, but its development has been discontinued. PySAD specializes in streaming data analysis, but requires a fixed frequency of input data, which limits its application. The Darts library offers a wide set of algorithms for time series analysis, but requires pre-filling of missing values, which creates an additional load on the model. The best results were achieved using the Prophet model, which is able to work with irregular time series without the need for additional data augmentation.

The experiments conducted showed that Prophet provides the best balance between forecast accuracy and computational efficiency. The choice of the amount of historical data for modeling is crucial, since excessive data increases processing time, and insufficient data reduces forecast accuracy.

The results of the study indicate that none of the considered libraries is a universal solution for all tasks. However, Prophet showed the greatest potential for detecting anomalies in the time series of Telegram metrics, which makes it the best candidate for further development and adaptation in media content monitoring tasks.

Keywords: predictive model, optimization, social networks, machine learning, data analytics, anomaly detection, engagement metrics.

Introduction

The realities of the modern information society demand new data monitoring and analysis approaches, especially given the increasing volume of content and its rapid dissemination. Social networks have become critical platforms for information exchange, yet their scalability and accessibility also create fertile grounds for the spread of disinformation. The need for rapid responses to these challenges is securing the information space, ensuring societal stability, and protecting democratic processes.

Analyzing anomalies in engagement metrics helps uncover mechanisms behind coordinated disinformation campaigns and develop tools to neutralize them, emphasizing the relevance of this research. Such anomalies can indicate orchestrated disinformation efforts to manipulate public opinion, deepen social divides, or even interfere in political processes.

By examining engagement metrics, researchers can identify isolated cases of disinformation and more complex networks involving fake accounts, bots, and coordinated groups. These networks are often employed to amplify harmful content and promote it to a broader audience. Detecting and analyzing these anomalies not only traces the sources of disinformation but also reveals the mechanisms of its dissemination.

Problem Statement

The primary goal of this research is to conduct a comprehensive analysis of the problem of anomaly detection in media data and evaluate the suitability of existing approaches to the defined requirements. This problem includes collecting and analyzing test data from relevant sources, defining criteria for anomaly detection methods and models for successful application in media analysis, and validating existing approaches for conformity with these criteria and their applicability to test data.

Anomaly detection in social network metrics can employ specialized methods for identifying outliers in data and prediction-based approaches. In the latter case, anomalies are identified as significant deviations between actual values and those predicted by a model.

However, anomaly detection in social network metrics has unique challenges that distinguish it from classical time-series analysis. These include irregular time series and uneven data distribution, which complicate the application of standard approaches. To address these challenges, specialized algorithms designed for such data or data augmentation methods can be used to adapt generic models to the specific characteristics of the analyzed metrics.

Current Approaches

Research on disinformation detection in social networks primarily focuses on clustering and network relationship analysis [1, 2]. While such methods effectively identify key disseminators of harmful content and trace back to its sources [3], they exhibit several limitations, including:

- Inability to process streaming data.
- Higher data volume requirements compared to alternative approaches.
- Complexity in analyzing data from messaging platforms as opposed to traditional social networks.
- Lack of insight into the dynamics of information spread.

Conversely, anomaly detection in time series offers distinct advantages [4] over clustering and network analysis:

- Near real-time responsiveness for timely detection and mitigation of disinformation.
- Reduced data volume requirements for practical analysis.
- Applicability to diverse data sources with public metrics such as views, shares, and reactions.

Critical Requirements for Effective Tools. The ideal tool for this task must satisfy the following criteria:

- Time Series Support: Handling temporal data is crucial for processing information with a time dimension.
- Incomplete Data Handling: Media data often lacks regular frequency, necessitating algorithms that can accommodate irregularities.

- Real-Time Processing: Immediate anomaly detection enables rapid responses to changes in metrics.
- Up-to-date Support: For sustainability, active development, regular updates, and a robust developer community are essential.
- Integration of Artificial Intelligence (AI): AI enhances predictive accuracy for chaotic data and uncovers hidden trends not detectable by conventional methods.
- Seasonality Awareness: Accounting for recurring patterns, such as daily or weekly cycles, reduces false positives.
- High Processing Speed: Efficient handling of large datasets ensures scalability.
- Multi-Metric Analysis: Accounting for correlations across multiple metrics improves detection accuracy.

The emphasis on time series and incomplete data handling arises from the inherent nature of media data. Real-time processing shortens detection and reaction times while maintaining tool relevance and ensures continued effectiveness.

Tool Comparison: Python Libraries

1. PyOD

PyOD is a widely used anomaly detection library offering a range of algorithms from basic to advanced [5]. However, it lacks native support for time series and real-time processing. While adaptable to specific needs, these limitations constrain its applicability to the problem. Nonetheless, its active community and extensive algorithm base make it a foundational resource for other libraries such as TODS and PySAD.

Evaluation. PyOD supports incomplete data with preprocessing but requires significant adaptation for real-time functionality. Although AI elements are integrated into some algorithms, the library lacks tools for seasonality handling and multi-metric analysis. Its performance is generally efficient, depending on the chosen algorithm.

2. TODS

TODS specializes in anomaly detection in multivariate time series and offers a comprehensive framework for creating detection pipelines, including preprocessing, feature extraction, and anomaly detection algorithms [6]. It supports common detection scenarios, such as point anomalies and pattern deviations. However, it does not provide real-time processing, and its development has ceased, leaving it without a stable release version.

Evaluation. Despite its excellent support for time series and incomplete data, TODS lacks AI integration and tools for real-time operation. Its support for seasonal adjustments and multi-metric analysis, combined with optimized performance for time-series tasks, makes it a strong contender for specific use cases. However, its long-term utility is limited by its development status.

3. PySAD

PySAD is a high-performance library focused on real-time anomaly detection for streaming data [7]. It updates models dynamically with new incoming data, ensuring immediate responsiveness. This focus makes it particularly suitable for real-time applications, although its feature set is less extensive than other libraries.

Evaluation. The library supports time series and incomplete data but lacks advanced features such as AI integration, seasonality adjustments, and multi-metric analysis. While its performance in real-time scenarios is excellent, the absence of recent updates and broader functionality limits its applicability to more complex requirements.

4. Darts

Darts is a Python library designed for user-friendly forecasting and anomaly detection in time series [8]. It includes a diverse range of models, from classical ones like ARIMA to deep neural networks. All forecasting models can be utilized in a unified manner through the fit() and predict() functions, resembling the interface of scikit-learn. The library also simplifies the backtesting of models, combining forecasts from multiple models and incorporating external data. Darts supports both univariate and multivariate time series and models. Machine learning-based models can be trained on po-

tentially large datasets containing multiple time series, and some models offer extensive probabilistic forecasting capabilities.

Evaluation. The library specializes in time series analysis, providing tools for handling missing data and accounting for seasonality. However, it lacks built-in artificial intelligence elements and cross-metric analysis capabilities. While not inherently designed for real-time applications, it can be adapted for such use, with performance dependent on the selected forecasting method. The library is actively updated and maintained by the community. Among its drawbacks, the library requires complete data for operation. Its built-in TimeSeries class only accepts complete datasets (i.e., data with a defined periodicity, such as minute-by-minute values) or datasets with an apparent periodicity, which is unsuitable for irregularly spaced data. This issue can be addressed through interpolation, but it imposes an additional computational load. For instance, 600 posts over a week with minute-by-minute interpolation would result in 10,080 points, two orders of magnitude more than the original count. Considering the need to train thousands of models, this approach can become resource-intensive. However, this limitation arises from the unified interface used to interact with models that employ varying methods and approaches. Specific models implemented in the Darts library can still handle incomplete data.

5. Prophet

The Prophet algorithm was developed by Facebook (now Meta) for time series forecasting, particularly in scenarios with pronounced seasonality and multiple seasonal cycles in historical data [9].

Prophet is based on an additive approach that combines time series with nonlinear trends. The trend component is modeled using a saturating growth model and piecewise linear functions. The seasonal component is modeled using the Fourier series. All of this is implemented using the Stan programming language, which is specialized in creating statistical mathematical models.

One of Prophet's key characteristics is its resilience to missing data, trend changes, and outliers, enabling the model to produce reliable forecasts regardless of input data quality. Additionally, Prophet offers intuitive parameters for customization, allowing domain experts to fine-tune predictions based on their expertise. Regarding efficiency, Prophet provides reasonable computation speeds and moderate computational resource requirements.

Evaluation. The library specializes in time series analysis, offering optimized performance for forecasting, built-in support for seasonality, and the ability to handle incomplete data. However, it is not designed for real-time operation, lacks artificial intelligence components, and does not support cross-metric analysis. It also requires additional adaptation for real-time use. Nevertheless, given its forecasting accuracy and computational speed, the algorithm is a solid choice for anomaly detection in article metrics.

Investigation of promising approaches

Dataset. This study utilized a dataset obtained from Telegram channels via the official API. The analysis was based on numerical metrics, including the number of views, shares, comments, and reactions to posts.

Historical data, however, was deemed irrelevant to the study as Telegram retains only the most recent values of metrics without storing their change history. This limitation complicates the analysis of dynamic trends in coverage metrics. Also, coverage metrics are unavailable when collecting data in real-time due to insufficient time for their accumulation. To address this issue, each post was collected at equal intervals after its publication (e.g., a post published at 12:11 was collected at 12:41, and one published at 15:43 was collected at 16:13) to ensure methodological accuracy. For forecasting purposes, the data was grouped by the originating channels and time intervals post-publication. Each dataset comprised posts from a single channel collected at identical intervals after their publication.

As the dataset was unannotated and the studied methods relied solely on unsupervised learning, manual processing was employed for metric evaluation. Outliers were identified and excluded manually, ensuring accurate metric calculations and enabling the evaluation of the proposed approaches' effectiveness.

In the task of detecting anomalies in media coverage metrics, the depth of data used for model training is not constrained by strict requirements. This constraints allows flexibility in selecting the volume of historical data optimal for forecasting and anomaly detection. A balance must be maintained between avoiding model overfitting and making efficient use of computational resources. Profound historical data may lead to unnecessary processing overhead, while insufficient data may compromise prediction accuracy. Thus, determining the data depth should be guided by experimental evaluation of the optimal volume for a given application.

The compliance of anomaly detection libraries with the defined criteria is summarized in table 1.

Comparative Analysis of Anomaly Detection Libraries

Criteria	Libraries				
	PyOD	TODS	PySAD	Darts	Prophet
Time series support	-	-	+	+	+
Handling incomplete data	-	+/-	-	+/-	+
Real-time operation	-	+	+	-	-
Is maintained	+	-	-	+	+/-
Artificial intelligence	+/-	-	-	-	-
Seasonality	+/-	+	+	+	+
High performance	+	+/-	-	+/-	+
Cross-metric analysis	-	+	-	-	-

The PyOD library, while recognized as a general-purpose tool for anomaly detection, does not natively support time series. It can only serve as a foundation for evaluation algorithms, similar to implementations in libraries like TODS or Darts. However, the development of TODS is currently on hold, and the library lacks a stable release version. Due to its limited functionality, including the lack of real-time support, its current form is impractical for use. PySAD, specifically designed for anomaly detection in streaming data, generates deviation probabilities for each data point but requires a fixed frequency of input data, necessitating additional preprocessing. To address this, the fill_missing_values function from the Darts library was employed to augment input data.

Fig. 1 illustrates anomaly values (red points) for each message (black points) obtained through sequential model training. The graph shows how the model gradually learns patterns in the data during the first day and stabilizes its predictions on the second day. However, the model exhibits overfitting tendencies, particularly noticeable between the third and seventh days, limiting its ability to effectively analyze complex seasonal patterns.

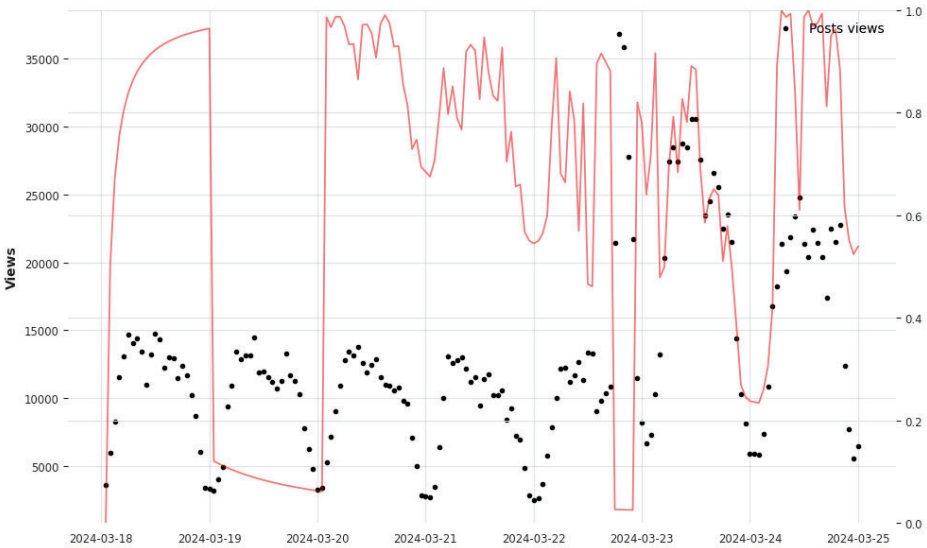

Fig. 1. Example of PySAD Results on Incomplete Data

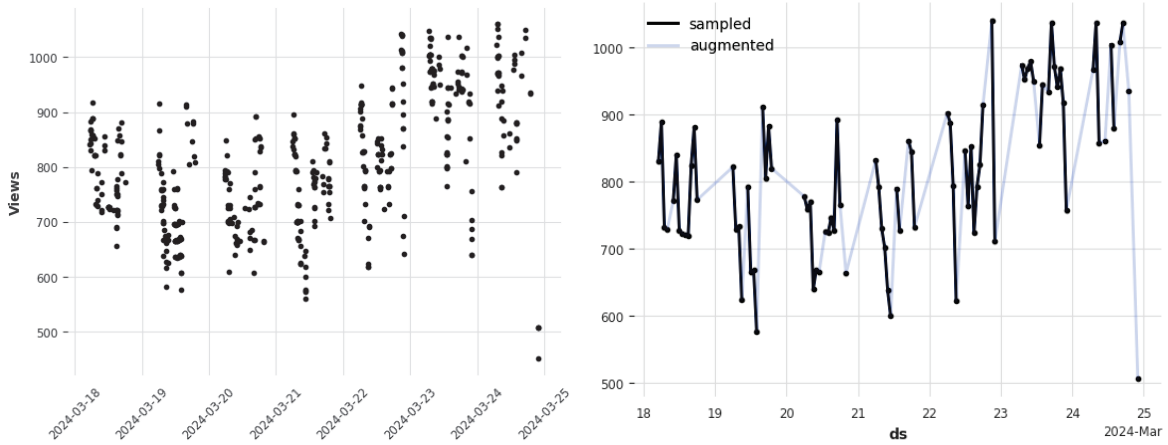

Fig. 2. Original (left) and Augmented (right) Data Example

The Darts library offers a wide range of time series analysis and forecasting algorithms, including TBATS and Prophet models, which delivered the best results. Data from a one-week period was used for forecasting, with the first six days designated as the training set and the seventh day for validation. Accuracy was evaluated using the Mean Absolute Percentage Error (MAPE) calculated on outlier-cleaned data. The data frequency was set to hourly, and missing values were filled using the Darts library, as shown in fig. 2. Simpler models, such as AutoARIMA and Theta, demonstrated comparable error rates but were unable to capture complex periodic trends, limiting their applicability in tasks requiring intricate seasonality analysis.

At the same time, the Prophet model, implemented independently, showed significant advantages due to its ability to handle irregular time series without prior sampling or data augmentation (fig. 3). Using "clean" data significantly reduced model execution time and improved prediction accuracy, as confirmed in table 2. This features makes Prophet a promising tool for solving forecasting and anomaly detection tasks in media metrics time series.

Comparative Table of Forecasting Models

Models	Input Data Frequency	Prediction Frequency	Mean Absolute Percentage Error (MAPE) (Lower is better)	Execution Time (s)
TBATS (Darts)	Hourly	Hourly	11.37%	22.260
Prophet (Darts)	Hourly	Hourly	13.11%	0.326
Prophet (Standalone)	Non-periodic	5 minutes step	11.00%	0.151

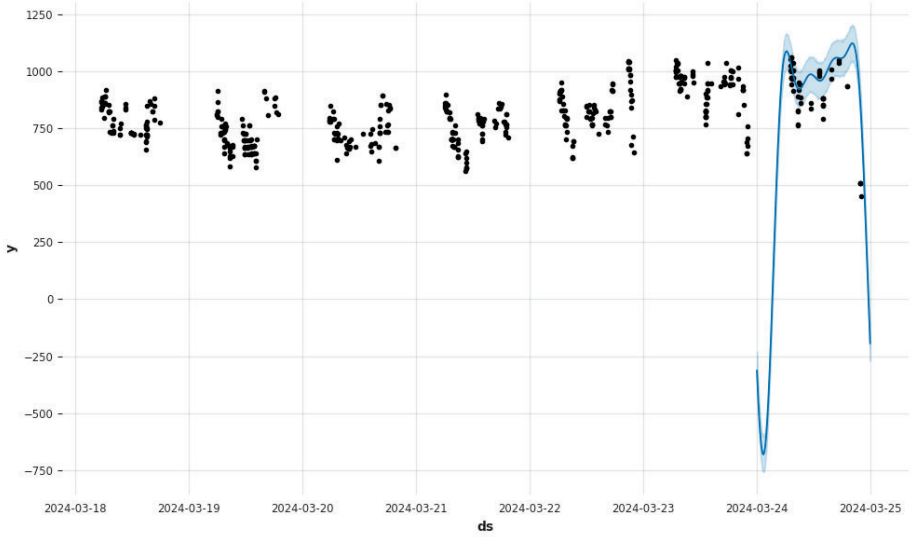

Fig. 3. Prophet Library Forecasting Example

Conclusions

This study addressed the problem of detecting anomalies in media data, specifically in Telegram channels, a critical tool for combating disinformation. Identifying anomalies in coverage metrics enables timely detection of individual instances of disinformation and complex networks, including fake accounts and bots. The primary objective was to determine effective methods and models for real-time anomaly analysis and detection, particularly relevant to social networks and messaging platforms.

The results indicate that no library simultaneously meets all the maximum and minimum requirements (e.g., time series handling, incomplete data support, and real-time functionality). While each library has its strengths, none offers a universal solution for all aspects. Libraries like PyOD and TODS excel in certain areas, such as performance or seasonality support, but are unsuitable for time series analysis. Meanwhile, libraries like PySAD and Darts cannot work with irregular data frequency. Among the studied tools, only Prophet handles irregular time series effectively, making its development and adaptation the most suitable option for outlier detection in coverage metrics.

References

1. Camacho, D., та ін. The four dimensions of social network analysis: An overview of research methods, applications, and software tools. // Information Fusion. – 2020. – Т. 63. – С. 88–120.

2. Yan, L., Rose, Y., Huida, Q., та ін. A Survey on Social Media Anomaly Detection // SIGKDD Explorations. – 2016. – Т. 18. – С. 1–45. – Режим доступу: https://kdd.org/exploration_files/Vol18-Issue1.pdf. – DOI: 10.48550/arXiv.1601.01102.

3. Koulouri, T., Hoy, N. A Systematic Review on the Detection of Fake News Articles // arXiv. – 2021. – № 2110.11240. – С. 1–22. – Режим доступу: https://arxiv.org/abs/2110.11240. – DOI: 10.48550/arXiv.2110.11240.

4. Kamran, S., Talha, M.A. A Review of Time-Series Anomaly Detection Techniques: A Step to Future Perspectives // Advances in Intelligent Systems and Computing. – 2021. – Т. 1363. – С. 1–13. – Режим доступу: https://www.researchgate.net/publication/344704514_A_Review_of_Time-Series_Anomaly_Detection_Techniques_A_Step_to_Future_Perspectives. – DOI: 10.1007/978-3-030-73100-7_60.

5. Zhao, Y., Nasrullah, Z., Li, Z. PyOD: A Python Toolbox for Scalable Outlier Detection // Journal of Machine Learning Research. – 2019. – Т. 20, № 96. – С. 1–7. – Режим доступу: http://jmlr.org/papers/v20/19-011.html.

6. Lai, K.-H., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., Chen, Y., Zumkhawaka, P., Wan, M., Martinez, D., Hu, X. TODS: An Automated Time Series Outlier Detection System // Proceedings of the AAAI Conference on Artificial Intelligence. – 2021. – Т. 35, № 18. – С. 16060–16062. – DOI: 10.1609/aaai.v35i18.18012.

7. Yilmaz, S.F., Kozat, S.S. PySAD: A Streaming Anomaly Detection Framework in Python // arXiv preprint arXiv:2009.02572. – 2020.

8. Herzen, J., Lässig, F., Piazzetta, S.G., Neuer, T., Tafti, L., Raille, G., Van Pottelbergh, T., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., Grosch, G. Darts: User-Friendly Modern Machine Learning for Time Series // Journal of Machine Learning Research. – 2022. – Т. 23, № 124. – С. 1–6. – Режим доступу: http://jmlr.org/papers/v23/21-1177.html.

9. Taylor, S.J., Letham, B. Forecasting at scale // PeerJ Preprints. – 2017. – Т. 5. – № e3190v2. – С. 1–25. – Режим доступу: https://peerj.com/preprints/3190/. – DOI: 10.7287/peerj.preprints.3190v2.