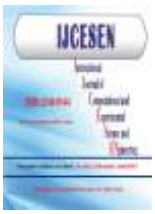
International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 9068-9078
http://www.ijcesen.com



ISSN: 2149-9144

Research Article

Real-Time AI-Driven Anomaly Detection in TV Ad Impression Logs Using Streaming Big Data Pipelines

Viharika Bhimanapati[1]*, Naveen K Chandu[2]

[1]Southern University and A&M College
* **Corresponding Author Email:** viharikaeb1@gmail.com, - **ORCID:** 0000-0002-5247-7990

[2]JNTU Hyderabad
Email: nchanduinbox@gmail.com- **ORCID:** 0000-0002-5247-7899

Article Info:

DOI: 10.22399/ijcesen.4367
Received : 08 February 2025
Accepted : 29 March 2025

Keywords

Real-Time Anomaly Detection;
Streaming Big Data;
TV Ad Impressions;
Machine Learning;
Apache Kafka

Abstract:

The growth of the television advertising ecosystems in complexity and size has gradually increased the importance of real-time monitoring and analysis of ad impression logs. The existing traditional (batch-based) and fixed-point anomaly detection systems are not suitable in the context of the high-velocity, high-volume streaming television data. In this paper, the author introduces an in-depth discussion of the AI-powered anomaly detection algorithms that can be implemented in streams of big data pipelines to provide the accuracy, integrity, and adherence of TV ad impressions. Through the use of more sophisticated machine learning and deep learning algorithms, including autoencoders, LSTMs, and ensemble algorithms that are integrated into scalable frameworks, e.g., Apache Kafka and Flink, real-time insights may be obtained at the lowest possible latency. The paper also examines the nature of anomalies that are often witnessed in the ad impression log, deploying model strategies, and performance of the system based on measures like precision, recall, latency, and throughput. The paper is concluded with a discussion of the existing issues, such as data heterogeneity, concept drift, and explainability, and the forecast of future advances in federated learning, edge AI, and hybrid detection models. This convergent strategy illustrates how AI and streaming data systems have a transformative potential in improving the performance (both operational and financial) of the TV advertising sector.

1. Introduction

Advertisement industry has experienced a paradigm change in the last ten years due to the expansion of digital platforms, programmatic advertising, and real-time bidding. Nonetheless, television is a formidable advertising tool, which is highly demanding in terms of revenue and has a wide reach to the audience. As the delivery and monitoring of television ads continues to grow more digitised, there has been a critical need to have real-time analytics that are capable of processing, interpreting, and responding to streaming data logs produced once television ads are impressioned. These real-time ad exposures as logs are not only large but also time sensitive in that they require an effective system to help identify anomalies like missed impressions, fraudulent insertions, or signal distortions at the time they take place.

The use of Artificial Intelligence (AI), specifically, machine learning (ML) and deep learning (DL) models, has enabled automatic detection of anomalies in mass data settings. Together with streaming big data systems like Apache Kafka, Apache Flink, and Spark Streaming, AI makes it possible to create pipelines that allow the detection of anomalies with a high degree of scalability and efficiency in real-time. The combination of AI and streaming big data is redesigning the capabilities of detecting anomalies in TV ad impression logs, which are highly accurate, scalable, and with actionable insights [1][2][3].

Anomaly detection in this case is critical. Abnormalities in logs of ad impressions may signify technical errors, fraudulence, or inefficiency. An example of this is that the presence

of differences between planned and actually aired impressions can result in the loss of money or failure to respect the agreement of the advertisers. Therefore, it is essential to implement a system that would be able to identify such anomalies in real time to keep TV advertising campaigns intact and profitable [4][5].There are special challenges associated with real-time anomaly detection. Conventional anomaly detectors tend to make use of fixed data sets and batch computation, which are not adapted to the pace and quantity of streaming information in actual TV broadcasting setups. Live television broadcasts are dynamic, so issues like delays on broadcasts, signal drops, or even regional differences in commercial placements require a low-latency solution based on context. AI-based models, which were trained on past and simulated data, are able to learn patterns on the fly and detect deviations with limited interference on the human side [6][7]. Moreover, the emergence of stream processing models has allowed ingestion, processing, and monitoring of moving logs. These systems are designed to deal with high throughput, fault-tolerant tolerant and scaling processes, which are needed to support real-time applications such as TV ad monitoring. Anomaly detection in the field of streaming architecture and AI models is severely demanding; therefore, both integrations are necessary [8][9].The analysis of the architecture of big data pipeline streaming, the use of AI in real-time analytics, and the purpose of the particular approach to the detection of anomalies in TV ad impressions records follow. A review of the technological background will be followed by a discussion of the AI-based models to detect anomalies, a discussion of the deployment strategies, and a conclusion of the analysis with a critical view of the performance, challenges, and future perspectives.

2. Streaming Big Data Architecture for TV Ad Impression Logs

Before analyzing the processes involved in achieving real-time anomaly detection, it is important to look at the infrastructure on which the logs of the TV ad impressions are processed. These are set-top boxes, satellite feeds, and ad insertion equipment generated logs that are continuously sent to centralised systems to be analysed, as illustrated by Figure 1. Data volume, variety, and velocity imply the need to have a streaming big data architecture that will cope with these attributes.

The basic streaming big data workflow of monitoring TV ad impressions includes ingestion layers, stream processing engines, long-term stores, and integration points of machine learning models. Apache Kafka or Pulsar, DMs that are often used as ingestion layers, serve as the point of entry for log data. These frameworks are durable, offer high-throughput and partitioned message delivery, and can process and scale simultaneously [10][11]. After being ingested, it is processed on stream processing engines such as Apache Flink or Apache Spark streaming. The frameworks provide windowed aggregations, support complex event processing, and external data sources/sink integration. They can conduct stateful computations in real time that are essential in identifying anomalies like deviations in the numbers of impressions, inconsistencies in timestamps, and missing records [12][13].Both raw and processed data are stored in persistent storage, which can be any of the following: significant time series, distributed file system, or time-series database, such as Apache HBase, Cassandra, or InfluxDB persistence, and both raw and processed data are later used to perform further analytics, historical comparisons, and model training. The models themselves are deployed either as a microservice or integrated into the stream processors so that they can score in real-time and raise alerts based on the incoming data [14][15]. The process of interfacing between these components is supported by orchestration tools and metadata management systems, which assure compatibility between the schemas, monitoring of latency, and recovery of faults. Another visualization tool with the pipeline is Grafana or Kibana, which can make real-time operational insights, dashboards, and anomaly reports.Streaming architecture is important because it can handle ad impression logs in real-time, with a minimum time response, resulting in the timely identification of anomalies and their remedy. The architecture is particularly important in situations in which the advertisers are charged on a per-impression basis, and when any difference to the campaign metrics is observed, it should be immediately reported so that reputational or financial losses can be unavoidable [16][17]. In addition, in this architecture, load-based scaling is achieved, enabling it to scale to peak broadcasting periods or special live events, which normally experience heavier ad loads and hence create more logs. Consequently, a streaming pipeline is not only the cornerstone of real-time analytics but a condition to operational excellence in contemporary TV advertising systems.

3. AI and Machine Learning for Real-Time Anomaly Detection

Since the basic architecture of streaming data has been established, there is an urgent need to

investigate the ways of embedding AI techniques into the pipeline to facilitate anomaly detection. This is not only difficult in terms of identifying patterns that are not as per expectations but also with low latency, high accuracy, and with the lowest false positives. The use of AI models introduces these opportunities through the learning of previous data, following the new trends, and constantly enhancing their accuracy in the detection. The commonly used machine learning models in this case are supervised, semi-supervised, and unsupervised learning models. Trained supervised models like the Random Forests and Gradient Boosting are used on labeled datasets comprising normal and anomalous ad impression activities. These models are trained in order to differentiate the two categories based on factors like time lapses, geographical locations, device numbers, and recurrence of impressions [18][19].

Nevertheless, in most real-life contexts, labeled anomalies are either rare or not available, and thus, it is more feasible to use unsupervised and semi-supervised methods. Isolation Forests, One-Class SVMs, and autoencoders are the most commonly used techniques. Autoencoders, especially deep learning ones, are effective in learning dense representations of normal behaviour and have demonstrated the ability to identify subtle behavioural anomalies in real-time streaming logs [20][21]. Furthermore, sequential dependencies in impression logs can be modeled with the help of the integration of temporal models, including Long Short-Term Memory (LSTM) networks and Temporal Convolutional Networks (TCNs). These models are skillful in identifying time-dependent anomalies, including an unexpected increase or decrease in the number of impressions within certain broadcasting periods. They can be particularly helpful in detecting the anomalies with the broadcasting schedules, blackouts, or the mismatch of regional feeds [22][23].

Feature engineering is important in improving the performance of models. Some of the important characteristics that are derived from streaming logs are frequency of impressions per time unit, time interval between advertisements, cumulative impressions, and entropy measurements of signal variation. Such characteristics are normalised and inputted into AI models to keep on learning and scoring. Predictions made by the model are sent back into the processing pipeline, and anomalies are raised or automated mitigation measures are taken [24][25]. The detection of concept drift and online learning are used to ensure the model's robustness. Since television programming and user behaviour continue to change, models do not need to be retrained every time to suit the emerging trends. Algorithms based on online learning that update their parameters as new information comes are also being used in the field.

Additionally, the ensemble learning methods, where many models are integrated in order to enhance the ability of prediction, are implemented to minimize the variance and bias. These ensembles can be (a) of different types of models, or (b) of several instances, which have been trained on different windows of data, which improves the generalization capabilities of the system to changing broadcasting conditions.

AI is incorporated into streaming pipelines by serving platforms such as TensorFlow Serving, MLflow, or a custom RESTful API. These platforms enable modeling to be scaled, and they incorporate the support of versioning, logging, and A/B testing. Performance of the models is continually monitored to ensure that the model is retrained and adjusted accordingly to avoid the degradation of the models with time. Therefore, the anomaly detection in TV ad impression logs through AI is powered by a synergistic approach to machine learning, temporal modeling, and real-time scoring to identify anomalies fast and accurately. In the following section, we explore the way these models are implemented in the production setting and how they are used to provide resilience, accuracy, and low latency.

In addition to the theoretical focus on the architecture of AI models and the algorithms applied when performing anomaly detection with reference to TV ad impression logs, one should also focus on the properties of different machine learning methods considered in the context of streaming data. The 4 tables below are a summary and comparison of some popular models used in real-time anomaly detection systems, their advantages, drawbacks, and areas of application. These comparative insights assist engineers and data scientists in selecting the appropriate algorithm depending on the anomaly characteristics, resource constraints, and the real-time processing requirements of the ad impression environment.

4. Deployment Strategies and Operational Considerations

It is not just the creation of AI models in real-time to detect anomalies. How well the models become a part of the wider ecosystem of data streaming and operational intelligence, represented in Figure 2, is equally important with reference to the deployment strategy. The deployment here should be made in such a way that it sets high standards regarding the latency, throughput, scalability, and fault tolerance. Models that perform well during offline or in test-

time may not be able to perform with the load and variability of real-world streaming data. Therefore, precise coordination is needed to make sure that the deployment is smooth, stable, and runs efficiently. Model deployment is usually implemented in two major models: edge deployment and centralized deployment. In edge deployment, the logic of anomaly detection is brought nearer to the source of data, like broadcast centers or set-top boxes, in which the latency of the anomaly detection can be reduced in the shortest possible time. It is especially useful when it is required to take immediate action concerning localized faults or anomalies (e.g., regional feed problems). Conversely, centrally deployed systems are found in cloud or data center architectures, which make use of increased computation and aggregation of a unified data set to increase model accuracy across geographies and channels [26][27].

A successful deployment plan usually uses the technologies of containerization, like Docker, and orchestration tools like Kubernetes. Such tools support the elastic scaling of anomaly detection services according to the data rates. They are also used to isolate model versions and control deployment pipelines in a continuous integration/continuous deployment (CI/CD) system so that the updates to models can be tested and verified, and deployed without interruption [28][29]. Serving platforms are used to provide real-time inference because deployed models can interact with streaming engines. As an example, an AI model deployed inside Apache Flink or via a gRPC API constantly consumes processed impression features, processes them, and provides a prediction score. When the score exceeds some predetermined anomaly threshold, the system sends out an alert or initiates automated recovery processes, including rerouting feeds or asking for retransmission of unsuccessful ad impressions [30].

Monitoring of models is also an operational consideration, which involves performance monitoring, anomaly verification, and feedback loops. Models can affect precision, recall, false positives, and latency values in real time because it is possible to monitor them using tracking systems, since they can become more inaccurate with time because of the concept drift or change of the broadcasting schedules. Visualisations of these metrics in the form of dashboards enable the operations teams to identify model failures at an early stage and initiate retraining workflows accordingly [31][32]. Explainable AI (XAI) methods are incorporated to ensure the models are coherent and explainable to the stakeholders in the business and compliance teams. They are SHAP (Shapley Additive Explanations) values or LIME (Local Interpretable Model-agnostic Explanations), which attempt to explain why a particular impression log was considered anomalous. These insights prove to be invaluable both in debugging as well as regulatory reporting and contract dispute resolution in cases where the anomalies of impression can have an impact on billing [33].

Another critical factor of deployment is security. Since the ad impressions logs can include user, broadcaster, and even advertiser metadata, the systems should meet the data protection laws. This requires encryption protocols, role-based access controls, and a secure API gateway. Furthermore, making sure that the models of anomaly detection are not susceptible to adversarial manipulation, including poisoned training data or inference-time attack, is a new field of study and a key component of sound deployment policy [34]. Finally, hybrid deployment models are becoming increasingly popular where components of the anomaly detection pipeline are deployed on-premise to comply and ensure lower latency, whereas other elements are deployed on a cloud infrastructure to scale and achieve centralised intelligence. This will offer a moderate balance between control, performance, and cost-effectiveness. With the continuing development of real-time TV ad impression monitoring, these deployment strategies will be maturing with more of an automation orientation, resilience, and collaboration with other systems. The following section also covers the nature of anomalies that are present in impression logs and what particular challenges they pose to the detection systems.

5. Types of Anomalies in TV Ad Impression Logs

The phenomenon of anomalies in the logs of impressions in the field of television advertising is not universal; it takes different forms, with other root causes and consequences. The problem of such anomalies should be understood to create specific detection tools and develop the corresponding mitigation measures. The artificial intelligence systems in real-time should be designed to suit the patterns, context, and nature of individual anomalies. The simplest type of anomaly is the volume-based anomalies, which are the inconsistencies in the quantity of impressions that occurred during a specified time period against the historical or anticipated records. Such anomalies usually indicate a problem like blackouts in the broadcast, failure of ad insertions, or loss of signal in the headend. As an illustration, when an advert to be run in prime time is booked on a high-rated program, and is actually recording very low

impressions when compared to other similar advertisements, then the system should issue an alert [35][36]. Temporal anomalies are those in which impressions are recorded out of the ordinary. This may be attributed to non-alignment of ad playouts, early or late transmissions, or imprecision of clocks between logging systems. An example of an advertisement that would cause a temporal anomaly alarm is an ad, at 30 seconds, at 8:00 PM, which records impressions at 8:03 PM. These need to be detected through careful synchronization of timestamps and expected timing distribution learning models [37].

Geospatial anomalies are the abnormalities of patterns of impressions in the various regions. These may be because of problems in the feeds within the region, transmission problems, or local network breakdowns. Indicatively, when the impressions of a traditionally high-activity urban area plummet to zero, and other areas stand at the same level, an anomaly at a local scale will have to be notified. Geo-clustering and regional baselines are AI models that identify such anomalies [38][39]. A pattern-based anomaly is another significant category in which the form or the rate of impression logs does not conform to anticipated behavioural patterns. These are frequent impressions, uncharacteristically brief or prolonged pauses between advertisements, and unusual sequence shifts. As an example, having two identical impression logs in quick succession may be a sign of a logging error in the system or intentional manipulation. LSTM/Markov chain-based sequence modeling is useful in the detection of such fine deviations [40][41].

Content-based anomalies are anomalies of metadata of ad impressions. These may be discrepancies between scheduled and aired creative IDs, a lack of advertiser information, or campaign identifier discrepancies. The cause of these anomalies may be inaccuracies in campaign configuration by humans, database syncing, or failure to integrate systems of various ad tech. The integrity of such logs is checked in real time with models that are trained on ad metadata structure [42]. Semantic anomalies also exist, and this is related to the mismatch of impressions in context. As an illustration, an alcohol advertisement that is aired during a children's show, though technically correct in terms of scheduling and metadata, would be reported as a semantic anomaly because of regulatory or brand safety issues. Such context-sensitive anomalies are increasingly being detected using semantic analysis models, which are usually founded on Natural Language Processing (NLP) and program metadata analysis [43].Lastly, the artificial anomalies are the ones that are added to train or test the model. These are essential in training supervised models, particularly when the real-world labeled anomalies are few. Imperative care should be taken in the sense that these synthetic instances correctly model actual anomalies, and not to overfit or unrealistic model behaviour. The process of identifying such different anomalies involves a complex AI solution that incorporates statistical baselining, time series modeling, automatic grouping, and real-time context analysis. The following section discusses the method of measuring system performance, the measures involved, and the way performance of the system is measured in real-life applications.

6. Evaluation Metrics, Performance Analysis, and Real-World Effectiveness

As soon as AI-driven anomaly detection models are implemented into the streaming big data pipelines to track TV ad impressions, their effectiveness, accuracy, and reliability of operations need to be evaluated continuously. An evaluation strategy that is properly calibrated is not only one that confirms that the model is effective, but one that is determined by actual implementation of the system, that is, variable loads, latency constraints, and a wide variety of types of anomalies. Precision, recall, and F1-score are the initial and most popular metrics of evaluation. Precision is the evaluation of the ratio of the correctly recognized anomalies to the total anomalies recognized, whereas recall is the evaluation of the recognized real anomalies in comparison to all the existing anomalies. The F1-score gives a harmonic average between the precision and recall as an independent measure when there is a need to trade off between false positives and false negatives. High precision means there will be a few false alarms, and this is essential in preventing unnecessary escalations. High recall is used to ensure that true threats or anomalies are not missed [44][45].

The other important measure is latency, which is especially important in streaming pipelines. As anomalies should be identified and responded to in real time, models are required to be able to deliver inference results on very tight time constraints, typically a few milliseconds to a couple of seconds, based on the needs of the system. Streaming systems such as Flink and Spark streaming are frequently supplied with latency dashboards and logs as a part of controlling how long it takes to process data in and then generate anomaly results [46]. One of the important performance measures is also throughput, which is the number of records processed per second. In TV networks with a massive scale, the quantity of ad impressions logs can be millions of impressions per hour. AI models

and streaming infrastructure should be able to process such data without back pressure and latency. The load testing and performance benchmarking are consistently conducted via synthetic traffic or historical replay to check peak conditions as well as gauge the robustness of the system [47]. Other measures that are critical in the operational reliability are the false positive rate (FPR) and false negative rate (FNR). High FPRs give rise to alert fatigue, in which operators start ignoring or procrastinating in responding to flagged anomalies, and may end up as they have missed genuine problems. On the other hand, high FNRs are instances of anomalies that go undetected, and this might incur some financial losses, conflicts with the advertisers, or even breaking of rules. An effective anomaly detector system reduces both, usually by retraining and tuning the threshold continuously [48]. Model discriminative ability at varying thresholds is also determined by Receiver Operating Characteristic (ROC) curves and Area Under Curve (AUC). These measures are of great use in the initial stages of model selection and comparison. Nevertheless, threshold values are optimized in the production environment with regard to business-specific risk tolerance instead of statistical optimality. In practice, a feedback loop used to check anomalies by operators and feedback the labeled feedback into the model training pipeline is often used. This adaptability and accuracy of models are enhanced over time through this iterative learning. Active learning is also used, in which the uncertain predictions are labeled as such to enable human analysis in order to enhance the efficiency of the data [49]. Real-time anomaly detection in TV ad logs has been shown to improve in various deployments in industries. Missing impressions were found in high-profile events such as sports broadcasting, which resulted in the instant retransmission of the content, ensuring no contractual fines. Equally, the detection of the problem of a time drift in the satellite uplinks also assisted in avoiding repetitive scheduling problems between several regional feeds [50][51].

In this regard, high-availability cluster deployments in container orchestration with Kubernetes have provided auto-scaling and self-healing systems with 99.99% uptime in the face of constant monitoring. There are hot-replication of stream processors and redundant model serving endpoints as strategies of failover to prevent single points of failure. Asynchronous communication patterns and batched scoring of time-windowed data are also optimized to use the interaction between the AI models and stream processors. Such strategies balance between latency and computational efficiency, especially with varying load conditions. As an example, in the case of live boxes, like political debates or award shows, the system changes the batch sizes and model scoring intervals on the fly to avoid overload without having impaired sensitivity to anomalies. Finally, not only the accuracy of the algorithms but also the design, the level of integration, and the feedback of the end-to-end pipeline can influence system performance. The following and final analytical discussion will focus on the existing problems of real-time AI-driven anomaly detection in TV advertising and what is to come in the future, and new technology that can be developed to build the new generation of monitoring systems.

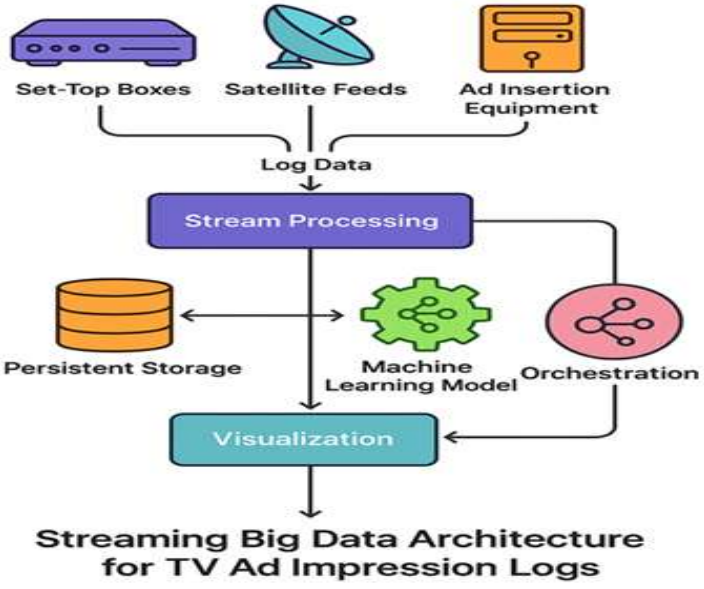


Figure 1: *Streaming Big Data Architecture for TV Ad Impression Logs, illustrating the flow from data sources through stream processing, storage, machine learning, and real-time visualization, with orchestration ensuring system integrity and scalability.*

Table 1: *Comparison of Machine Learning Models for Real-Time Anomaly Detection in TV Ad Logs*

Model Type	Learning Style	Strengths	Limitations	Best Use Cases
Random Forest	Supervised	High accuracy, handles categorical features well	Requires labelled data; less effective for time-series	Detecting ad count anomalies with labels
Isolation Forest	Unsupervised	Efficient, interpretable, low memory footprint	Less effective for high-dimensional sequences	Volume and outlier detection in ad logs
Autoencoder (DL)	Unsupervised	Learns complex patterns, scalable	Black-box, requires tuning, potential overfitting	Rare anomaly patterns in ad metadata
One-Class SVM	Semi-supervised	Effective for boundary-based anomaly detection	Poor scalability with large data	Flagging unusual time gaps or sequences
LSTM Neural Network	Supervised / Seq.	Excellent at temporal sequence modeling	High latency, needs extensive training data	Predicting temporal anomalies in impression timing

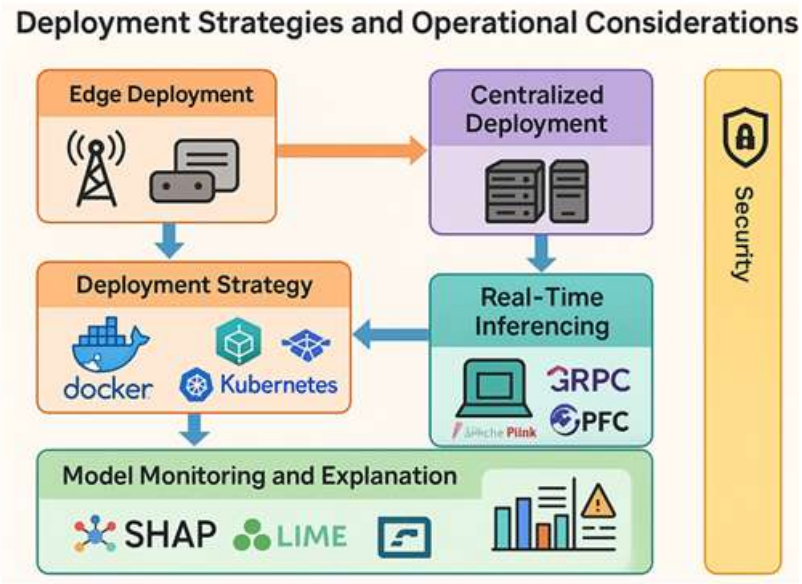


Figure 2: *Diagram illustrating deployment strategies and operational considerations for real-time AI anomaly detection, including edge and centralized deployments, containerization, inferencing platforms, model monitoring, explainability tools, and security safeguards.*

Table 2: *Current Challenges and Emerging Solutions in AI-Driven Anomaly Detection Systems*

Challenge	Description	Impact on System	Emerging Solution
Data Heterogeneity	Varying log formats and inconsistent metadata sources	Increases false positives; data integration issues	Schema-on-read frameworks; data normalization tools
Lack of Labelled Anomalies	Scarcity of real-world anomaly data for training	Reduces model accuracy	Synthetic anomaly generation; semi-supervised learning
Concept Drift	Changes in viewing patterns and broadcasting behaviour	Model degradation over time	Online learning; adaptive thresholding
High Inference Latency	Slow response times for deep models in streaming environments	Missed real-time anomaly detection windows	Model pruning; inference optimization
Limited Explainability	Difficulty interpreting complex model decisions	Hinders stakeholder trust and adoption	Explainable AI frameworks (e.g., SHAP, LIME)

7. Challenges and Future Directions

Although the introduction of AI-based anomaly detection systems in the streaming big data pipelines has revolutionized the operation of TV ad impressions monitoring, there are various issues that are still impeding the best functionality and extensive use. The problem with these limitations should be addressed to make the most out of these technologies in an industry where money and thoughts count. One of the major issues is that of data quality and heterogeneity. The TV ad impression logs come in a variety of devices, vendors, and broadcasting regions, and have

differing data schema, logging format, and reliability levels. It is non-trivial to normalize and harmonize such data in real time, and it may take, however, considerable pre-processing and metadata reconciliation. Unstable data raises the risk of creating a false positive and complicates the process of training AI models to adopt consistent patterns [52]. The other problem that exists is the unlabeled anomaly data. Anomalies in the real world are not labeled, and annotating by hand is very expensive and time-intensive. This impacts the supervised learning methods, requiring the implementation of unsupervised models that are not necessarily accurate or interpretable. This can be solved by synthetic data generation, although synthetic anomalies do not necessarily always reflect the complexity of real-world problems, particularly in edge cases [53]. Another major challenge is concept drift, or the shifting nature of what is considered normal behavior in ad impression logs as time goes on. The distribution shifts in the data can be caused by seasonal programming changes, by regional preferences in the ad, or by fluctuations in the regulatory requirements. In the absence of systems to monitor and adjust to these changes, like online learning or an adaptive thresholds model, accuracy may deteriorate quickly [54]. Scalability is also of concern. Although the streaming systems are theoretically very scalable, bottlenecks can be created when resource-consuming deep learning models are implemented in the high-throughput systems. Inference latency, memory footprint, and model compression methods are critically important to optimize, particularly when it is necessary to deploy at a resource-constrained deployment target like an edge device or a regional broadcast centre [55]. Model explainability is yet another area of concern. Although more complicated models, such as LSTMs or autoencoders, are useful when it comes to temporal and latent anomalies, they tend to be black boxes. This restricts their use in situations where accountability and interpretability are important, like in court proceedings on a contract involving impression counts or an audit by government bodies. Methods such as SHAP or integrated gradient are under development and research, but in the streaming anomaly detection scenario [56].

In the near future, the sphere will develop in a number of promising ways. The federated learning models can enable several broadcasters or networks to concurrently train the anomaly detection systems without having to share the raw data, maintaining privacy and compliance, and enhancing the robustness of the models. Equally, transfer learning methods have the potential to allow pre-trained models on one broadcast network to be adapted to another with minimal labeled data [57]. The involvement of emerging graph-based anomaly detection mechanisms is also likely to be used, especially in detecting the relational anomalies across channels, advertisers, and regional groupings. Such methods have been used to model complicated associations between various objects in ad ecosystems and identify anomalies that are not apparent in a flat, tabular log data. In addition, hybrid AI systems, which are a combination of rule-based logic and machine learning, are becoming popular. These systems are more interpretable and at the same time flexible and adaptable. As a case in point, business regulations can be used to regulate base levels, whereas AI algorithms can be used to highlight less obvious and more context-dependent anomalies. Serverless computing and edge AI are facilitating more distributed applications in terms of infrastructure and at lower costs. These technologies enable the ad monitoring systems to scale elastically during peak times and scale down to minimal operations overheads during low-peak periods. This will enable these edge systems to become feasible with the emergence of 5G and ultra-low-latency networks, where it is possible to employ real-time anomaly detection directly at the sources of data. Lastly, ethical and regulatory compliance are being given more consideration. With more and more financial transactions and advertiser relations being based on AI decisions, it will be essential to make sure that detection models are not prejudiced, biased, or hidden. It may only take regulatory frameworks a short time to make AI-driven impression tracking systems auditable, which will further demand strict documentation, logging, and validation pipelines. To sum up, the real-time anomaly detection of TV ad impressions logs is a growing field with enormous potential. The future generation of systems will provide unprecedented accuracy, speed, and transparency by solving the present-day issues and utilizing the developing technologies to transform the way the television advertising business operates.

This is essential to plan the future improvement and address the current drawbacks of the real-time anomaly detection system, and outline the main issues of deployments today and the potential technological improvements. A systematic description of these challenges, their implications, as well as future strategies that are currently being explored, is provided in the table 2.

Managing these issues with a mixture of architectural and algorithm development is crucial to the continued development and reliability of

anomaly detection systems in television advertisement analytics.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

1. Fragkoulis, M., Carbone, P., Kalavri, V., & Katsifodimos, A. (2024). A survey on the evolution of stream processing systems. *The VLDB Journal*, *33*(2), 507-541.
2. Chan, J. O. (2013). An architecture for big data analytics. *Communications of the IIMA*, *13*(2), 1.
3. Thayyib, P. V., Mamilla, R., Khan, M., Fatima, H., Asim, M., Anwar, I., ... & Khan, M. A. (2023). State-of-the-art of artificial intelligence and big data analytics reviews in five different domains: a bibliometric summary. *Sustainability*, *15*(5), 4026.
4. Ezzat, R. (2024). Enhance the advertising effectiveness by using artificial intelligence (AI). *Journal of Art, Design and Music*, *3*(1), 1.
5. Lu, T., Wang, L., & Zhao, X. (2023). Review of anomaly detection algorithms for data streams. *Applied Sciences*, *13*(10), 6353.
6. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: a review. *Data mining and knowledge discovery*, *33*(4), 917-963.
7. Singh, P. (2021). Deploy machine learning models to production. *Cham, Switzerland: Springer*.
8. Cherniack, M., Balakrishnan, H., Balazinska, M., Carney, D., Cetintemel, U., Xing, Y., & Zdonik, S. B. (2003, January). Scalable Distributed Stream Processing. In *CIDR* (Vol. 3, pp. 257-268).
9. Nazari, E., Shahriari, M. H., & Tabesh, H. (2019). Big Data analysis in healthcare: Apache Hadoop, Apache Spark, and Apache Flink. *Frontiers in Health Informatics*, *8*(1), 14.
10. Pal, G., Li, G., & Atkinson, K. (2018, August). Big data real-time ingestion and machine learning. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 25-31). IEEE.
11. Bhandari, R. (2024). Data storage handler & technology: evaluation of real-time data handling technology.
12. Daksa, R. P., & Kemala, A. P. (2025). A Comparative Study On Real-Time Data Streaming For Fraud Detection Using Kafka With Apache Flink And Apache Spark. *Procedia Computer Science*, *269*, 192-199.
13. Zahra, F. T., Bostanci, Y. S., Tokgozlu, O., Turkoglu, M., & Soyturk, M. (2024). Big Data Streaming and Data Analytics Infrastructure for Efficient AI-Based Processing. In *Recent Advances in Microelectronics Reliability: Contributions from the European ECSEL JU project iRel40* (pp. 213-249). Cham: Springer International Publishing.
14. Zhang, J., Wu, G., Hu, X., & Wu, X. (2012, September). A distributed cache for Hadoop Distributed File System in real-time cloud services. In *2012 ACM/IEEE 13th International Conference on Grid Computing* (pp. 12-21). IEEE.
15. Thota, S., Chitta, S., Alluri, V., Vangoor, V., & Ravi, C. S. (2022). MLOps: Streamlining machine learning model deployment in production. *African J. of Artificial Int. and Sust. Dev*, *2*(2), 186-206.
16. Kumar, P. (2024). AI-Powered Fraud Prevention in Digital Payment Ecosystems: Leveraging Machine Learning for Real-Time Anomaly Detection and Risk Mitigation. Journal of Information Systems Engineering and Management 2024, 9(4) e-ISSN: 2468-4376
17. Chandramouli, B., Goldstein, J., & Duan, S. (2012, April). Temporal analytics on big data for web advertising. In *2012, IEEE 28th International Conference on Data Engineering* (pp. 90-101). IEEE.
18. Živanović, M., Štrbac-Savić, S., & Minchev, Z. (2023). An application of machine learning methods for anomaly detection in internet advertising. *Journal of Computer and Forensic Sciences*, *2*(1), 53-61.
19. Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, *58*, 308-324.
20. Finke, T., Krämer, M., Morandini, A., Mück, A., & Oleksiyuk, I. (2021). Autoencoders for unsupervised anomaly detection in high-energy physics. *Journal of High Energy Physics*, *2021*(6), 1-32.
21. Ribeiro, D., Matos, L. M., Moreira, G., Pilastri, A., & Cortez, P. (2022). Isolation forests and deep autoencoders for industrial screw tightening anomaly detection. *Computers*, *11*(4), 54.
22. Lin, S., Clark, R., Birke, R., Schönborn, S., Trigoni, N., & Roberts, S. (2020, May). Anomaly detection for time series using vae-lstm hybrid model. In *ICASSP 2020-2020 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4322-4326). IEEE.

23. Wikle, C. K. (2019). Comparison of deep neural networks and deep hierarchical models for spatio-temporal data. *Journal of Agricultural, Biological and Environmental Statistics*, *24*(2), 175-203.
24. Helskyaho, H., Yu, J., & Yu, K. (2021). Building Reproducible ML Pipelines Using Oracle Machine Learning. In *Machine Learning for Oracle Database Professionals: Deploying Model-Driven Applications and Automation Pipelines* (pp. 249-282). Berkeley, CA: Apress.
25. Ziffer, G., Bernardo, A., Della Valle, E., Cerqueira, V., & Bifet, A. (2023). Towards time-evolving analytics: Online learning for time-dependent evolving data streams. *Data Science*, *6*(1-2), 1-16.
26. Maltezos, E., Lioupis, P., Dadoukis, A., Karagiannidis, L., Ouzounoglou, E., Krommyda, M., & Amditis, A. (2022). A video analytics system for person detection combined with edge computing. *Computation*, *10*(3), 35.
27. Davis, J. (2008). Beyond the false dichotomy of centralized and decentralized IT deployment. *The Tower and The Cloud*, 118.
28. Kumar, A., Cuccuru, G., Grüning, B., & Backofen, R. (2023). An accessible infrastructure for artificial intelligence using a Docker-based JupyterLab in Galaxy. *GigaScience*, *12*, giad028.
29. Immaneni, J. (2021). Scaling Machine Learning in Fintech with Kubernetes. *International Journal of Digital Innovation*, *2*(1).
30. Horchidan, S., Kritharakis, E., Kalavri, V., & Carbone, P. (2022, June). Evaluating model serving strategies over streaming data. In *Proceedings of the Sixth Workshop on Data Management for End-To-End Machine Learning* (pp. 1-5).
31. Sørbø, S., & Ruocco, M. (2024). Navigating the metric maze: A taxonomy of evaluation metrics for anomaly detection in time series. *Data Mining and Knowledge Discovery*, *38*(3), 1027-1068.
32. Aissani, N., Beldjilali, B., & Trentesaux, D. (2008). Use of machine learning for continuous improvement of the real-time heterarchical manufacturing control system performances. *International Journal of Industrial and Systems Engineering*, *3*(4), 474-497.
33. Nazat, S., Arreche, O., & Abdallah, M. (2024). On evaluating black-box explainable AI methods for enhancing anomaly detection in autonomous driving systems. *Sensors*, *24*(11), 3515.
34. Khan, K. (2023). Adaptive video streaming: navigating challenges, embracing personalization, and charting future frontiers. *International Transactions on Electrical Engineering and Computer Science*, *2*(4), 172-182.
35. Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y., ... & Zhang, D. (2019, August). Robust log-based anomaly detection on unstable log data. In *Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering* (pp. 807-817).
36. Evensen, P., & Meling, H. (2012, July). AdScorer: an event-based system for near real-time impact analysis of television advertisements (industry article). In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems* (pp. 85-94).
37. Lam, K. Y., Chan, E., & Yuen, J. C. H. (2000). Approaches for broadcasting temporal data in mobile computing systems. *Journal of Systems and Software*, *51*(3), 175-189.
38. Moosmann, P. (2018). A geo-clustering approach for the detection of areas of interest.
39. Lee, J. G., & Kang, M. (2015). Geospatial big data: challenges and opportunities. *Big Data Research*, *2*(2), 74-81.
40. Akbilgic, O., & Howe, J. A. (2017). Symbolic pattern recognition for sequential data. *Sequential Analysis*, *36*(4), 528-540.
41. Abner, E. L., Charnigo, R. J., & Kryscio, R. J. (2013). Markov chains and semi-Markov models in time-to-event analysis. *Journal of biometrics & biostatistics*, (e001), 19522.
42. Visengeriyeva, L., & Abedjan, Z. (2018, July). Metadata-driven error detection. In *Proceedings of the 30th International Conference on Scientific and Statistical Database Management* (pp. 1-12).
43. Häglund, E., & Björklund, J. (2024). AI-driven contextual advertising: Toward relevant messaging without personal data. *Journal of Current Issues & Research in Advertising*, *45*(3), 301-319.
44. Nasir, W., & Jack, H. (2025). Real-Time Machine Learning Pipelines: Optimizing Stream Processing for Scalable AI Applications. *ResearchGate AI & Data Science Journal*.
45. Thirimanne, S. P., Jayawardana, L., Yasakethu, L., Liyanaarachchi, P., & Hewage, C. (2022). Deep neural network-based real-time intrusion detection system. *SN Computer Science*, *3*(2), 145.
46. da Silva Veith, A., de Assuncao, M. D., & Lefevre, L. (2018, November). Latency-aware placement of data stream analytics on edge computing. In *International Conference on Service-Oriented Computing* (pp. 215-229). Cham: Springer International Publishing.
47. Javed, M. H., Lu, X., & Panda, D. K. (2017, December). Characterization of big data stream processing pipeline: A case study using Flink and Kafka. In *Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies* (pp. 1-10).
48. Grill, M., Pevný, T., & Rehak, M. (2017). Reducing false positives of network anomaly detection by local adaptive multivariate smoothing. *Journal of Computer and System Sciences*, *83*(1), 43-57.
49. Ghassemi, M., Sarwate, A. D., & Wright, R. N. (2016, October). Differentially private online active learning with applications to anomaly detection. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security* (pp. 117-128).
50. Li, H. (2019). Special section introduction: Artificial intelligence and advertising. *Journal of Advertising*, *48*(4), 333-337.

51. Pradhan, C., & Trehan, A. (2024). Data engineering for scalable machine learning, designing robust pipelines. *International Journal of Computer Engineering and Technology (IJCET)*, *15*(6), 1840-1852.
52. Gupta, V., & Hewett, R. (2019, November). Adaptive normalization in streaming data. In *Proceedings of the 3rd International Conference on Big Data Research* (pp. 12-17).
53. Sistrunk, A., Cedeno, V., & Biswas, S. (2020). On synthetic data generation for anomaly detection in complex social networks. *arXiv preprint arXiv:2010.13026*.
54. Madireddy, S., Balaprakash, P., Carns, P., Latham, R., Lockwood, G. K., Ross, R., ... & Wild, S. M. (2019, August). Adaptive learning for concept drift in application performance modeling. In *Proceedings of the 48th International Conference on Parallel Processing* (pp. 1-11).
55. Park, J., Aryal, P., Mandumula, S. R., & Asolkar, R. P. (2023). An optimized dnn model for real-time inferencing on an embedded device. *Sensors*, *23*(8), 3992.
56. Li, Z., Zhu, Y., & Van Leeuwen, M. (2023). A survey on explainable anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, *18*(1), 1-54.
57. Su, C., Wei, J., Lei, Y., & Li, J. (2023). A federated learning framework based on transfer learning and knowledge distillation for targeted advertising. *PeerJ Computer Science*, *9*, e1496.