IRJEAS
NURTURING RESEARCH

INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING & APPLIED SCIENCES

ISSN: 2322-0821(O)
ISSN: 2394-9910(P)

VOLUME 13 ISSUE 1
Jan 2025 - Mar 2025

www.irjeas.org

Review Article

Machine Learning Based Frameworks for Real-Time Anomaly Detection in Big Data Environments

Anurag Shrivashtav[1], Amit Kushwaha[2]

[1]Associate Professor and Head, Computer Science and Engineering, NRI Institute of Information Science and Technology Bhopal
anurag.shri08@gmail.com
[2]Independent Researcher, Bhopal
amitksaga04@gmail.com

Corresponding Author: *anurag.shri08@gmail.com*

DOI –10.55083/irjeas.2025.v13i01004

Abstract: The rapid growth of big data in various domains has led to an increasing need for real-time anomaly detection systems capable of identifying unusual patterns or behaviors as they occur. Traditional methods for anomaly detection often struggle to scale effectively in big data environments due to the high volume, velocity, and variety of data. Machine learning (ML) techniques have emerged as powerful tools for developing real-time anomaly detection frameworks, leveraging their ability to learn from data and detect complex patterns. This paper explores machine learning-based frameworks for real-time anomaly detection, focusing on key approaches such as supervised, unsupervised, and semi-supervised learning. It also highlights their applications in sectors such as cybersecurity, finance, healthcare, and IoT. Furthermore, the paper discusses the challenges faced in real-time anomaly detection, including scalability, latency, and model adaptation. Finally, it examines future research directions, such as explainability, federated learning, and hybrid approaches, aimed at enhancing the effectiveness and reliability of these frameworks in dynamic, real-world environments.

Keywords: Machine learning, anomaly detection, real-time processing, big data, supervised learning, unsupervised learning, semi-supervised learning, stream processing, cybersecurity, fraud detection, IoT, healthcare, scalability, latency, model adaptation.

1. INTRODUCTION

The advent of big data has brought about significant transformations across industries, leading to the generation of vast amounts of information at unprecedented speeds. While this data holds valuable insights, it also presents unique challenges, particularly in the realm of anomaly detection. Anomalies, or deviations from expected patterns, can signal critical events such as fraud, system failures, security breaches, or operational inefficiencies. The timely detection of such anomalies is essential for ensuring the integrity and performance of systems in real-time.

Traditional methods of anomaly detection often rely on predefined thresholds or rule-based systems, which may not scale effectively in big data environments. With the sheer volume, velocity, and variety of data being produced, these methods often fall short in identifying complex, subtle, or evolving anomalies. In contrast, machine learning (ML) techniques have emerged as a powerful solution to address these challenges, offering the ability to learn from large datasets and detect anomalies without explicit programming of rules.

Machine learning-based frameworks for real-time anomaly detection leverage the predictive capabilities of algorithms to identify unusual patterns as data is ingested. These systems can continuously monitor and analyze data streams, making it possible to detect anomalies in real time, even in environments where data is constantly changing and growing. Machine learning approaches, such as supervised, unsupervised, and semi-supervised learning, each offer unique advantages depending on the availability of labeled data, the nature of the anomaly, and the application domain.

The integration of machine learning with real-time data processing frameworks, such as Apache Kafka, Apache Flink, and Apache Spark, has further enhanced the ability to scale anomaly detection in big data environments. However, despite the progress made in this field, several challenges remain, including issues of scalability, latency, data quality, and the need for continuous model adaptation. This paper explores the role of machine learning-based frameworks in real-time anomaly detection within big data environments, discussing key algorithms, their applications across industries, and the challenges that must be overcome to ensure effective and reliable detection.

By providing an in-depth understanding of current advancements, challenges, and future directions, this paper aims to offer a comprehensive overview of the potential and limitations of machine learning-driven anomaly detection systems in the era of big data.

2. BACKGROUND AND LITERATURE REVIEW

The rapid advancement of data generation in modern systems has led to the emergence of big data environments, characterized by large-scale, high-velocity, and high-variety datasets. In these environments, anomaly detection has become a critical task due to its ability to identify irregular patterns that could indicate security breaches, fraud, system failures, or operational inefficiencies. Anomalies can be defined as data points or patterns that significantly deviate from the norm, and their detection is essential for maintaining the integrity, security, and performance of complex systems. Traditional statistical methods often fail to meet the demands of big data environments due to issues with scalability, adaptability, and real-time processing. This has prompted the adoption of machine learning (ML) techniques, which offer more scalable and adaptive solutions for detecting anomalies in real-time big data streams.

2.1 Anomaly Detection in Big Data Environments

Anomaly detection has been widely studied in various domains, including network security, fraud detection, system monitoring, and healthcare. In the context of big data, however, traditional methods, such as threshold-based techniques and simple statistical models, often fall short due to the following challenges:

1. **Volume**: Big data environments involve the continuous generation of large amounts of data, making it difficult for conventional methods to handle such high volumes efficiently.
2. **Velocity**: Real-time anomaly detection requires processing and analyzing data as it streams in, making latency a critical issue. Traditional methods are often too slow to respond in time-sensitive scenarios.
3. **Variety**: Big data is often unstructured or semi-structured (e.g., text, images, sensor data), and detecting anomalies across diverse data types requires more sophisticated models.

In light of these challenges, machine learning has become an ideal approach for anomaly detection. ML algorithms can automatically learn patterns from the data, adapt to changes over time, and scale to handle large datasets efficiently.

2.2 Machine Learning Approaches for Anomaly Detection

Machine learning algorithms can be broadly categorized into three types for anomaly detection: **supervised learning**, **unsupervised learning**, and **semi-supervised learning**. Each of these approaches has its advantages and limitations in big data environments.

2.2.1 Supervised Learning

Supervised learning techniques require labeled datasets where instances are pre-classified as normal or anomalous. These algorithms build models based on this labeled data to classify new, unseen data. Common supervised learning algorithms used in anomaly detection include:

a. **Support Vector Machines (SVM)**: SVMs are particularly effective for binary classification problems and can be adapted for anomaly detection using one-class SVM, where the model learns to identify the boundaries of normal data and flags any deviations as anomalies.
b. **Random Forests**: This ensemble learning method constructs a multitude of decision trees and aggregates their outputs to classify data points. Random forests have been successfully used in detecting anomalies in high-dimensional data.
c. **Neural Networks**: Deep learning models, such as auto encoders, have been employed for detecting subtle anomalies by learning the underlying structure of data.

While supervised methods often achieve high accuracy, their dependence on large labeled datasets poses a significant challenge in real-time anomaly detection, especially in scenarios where obtaining labeled data is difficult or time-consuming.

2.2.2 Unsupervised Learning

Unsupervised learning methods, unlike supervised techniques, do not require labeled data. These methods identify anomalies by learning the underlying distribution of the data and flagging instances that deviate from this learned pattern. Common unsupervised learning techniques used for anomaly detection include:

a. **Clustering**: Techniques such as k-means and DBSCAN group similar data points together, and outliers are identified as points that do not belong to any cluster.
b. **Dimensionality Reduction**: Methods such as Principal Component Analysis (PCA) reduce the dimensionality of data while preserving its variance. Anomalies are detected by analyzing the data points that lie far from the principal components.
c. **Isolation Forests**: This method isolates anomalies through a process of random partitioning. Points that are more easily isolated are considered anomalous.

Unsupervised learning is particularly valuable in big data environments because it does not require labeled data, making it more suitable for real-time anomaly detection where labeled data is scarce or unavailable. However, the challenge lies in maintaining high accuracy, as unsupervised methods can produce false positives if the data is noisy or the model is poorly trained.

2.2.3 Semi-Supervised Learning

Semi-supervised learning combines both labeled and unlabeled data to train models. This approach is particularly useful when labeled data is scarce but a large volume of unlabeled data is available. Semi-supervised techniques offer a middle ground between supervised and unsupervised learning and have been successfully applied in real-time anomaly detection. Notable algorithms include:

a. **One-Class SVM**: This approach involves training the model on normal data and then identifying any data points that deviate significantly from the learned boundary as anomalies.
b. **Semi-Supervised Auto encoders**: Auto encoders can be trained on normal data to learn a compressed representation, and anomalies are detected by measuring reconstruction errors.

Semi-supervised learning provides flexibility and can improve accuracy when labeled data is limited but still available. However, the challenge remains in determining the optimal proportion of labeled to unlabeled data for training the model effectively.

2.3 Real-Time Anomaly Detection Frameworks

The integration of machine learning models with real-time data processing frameworks is essential for implementing anomaly detection in big data environments. These frameworks must be capable of processing continuous streams of data with low latency while integrating machine learning models to detect anomalies in real time. Stream processing technologies such as Apache Kafka, Apache Flink, and Apache Spark Streaming are commonly used in combination with ML models to handle large-scale real-time data streams.

a. **Apache Kafka** is widely used for distributed message queuing and data streaming. It can handle high throughput and provides fault tolerance, making it suitable for integrating machine learning models for real-time anomaly detection.
b. **Apache Flink** is a stream processing engine that can handle stateful computations, allowing complex event processing and anomaly detection in real time. It is well-suited for applications requiring low-latency processing.
c. **Apache Spark Streaming** offers micro-batch processing for real-time data streams, providing a robust platform for integrating ML models like decision trees, clustering algorithms, and deep learning models.

By combining these stream processing systems with ML-based anomaly detection models, organizations can detect anomalies as they occur and take immediate corrective actions.

2.4 Applications of Real-Time Anomaly Detection

Machine learning-based anomaly detection frameworks have been successfully applied in various domains, each requiring real-time anomaly detection due to the critical nature of the data being processed:

1. **Cybersecurity**: Detecting intrusions, malware, and abnormal network behavior in real time to prevent data breaches and attacks.
2. **Finance**: Identifying fraudulent transactions and market manipulation by continuously analyzing transaction data in real time.
3. **Healthcare**: Monitoring patient vitals and detecting abnormal health patterns that may indicate medical emergencies.
4. **IoT and Smart Cities**: Analyzing sensor data from connected devices and infrastructure to detect abnormal behavior, such as equipment failures or unauthorized access.

These applications highlight the growing need for real-time anomaly detection systems that can operate at scale and with low latency, ensuring that any irregularities are detected promptly and mitigating potential risks.

3. MACHINE LEARNING APPROACHES FOR REAL-TIME ANOMALY DETECTION

Machine learning (ML) has become an indispensable tool for real-time anomaly detection in big data environments, enabling systems to learn from data and automatically detect patterns and outliers. Different machine learning approaches can be applied based on the nature of the data and the problem at hand. These approaches can be broadly categorized into three types: **supervised learning**, **unsupervised learning**, and **semi-supervised learning**. Each of these methods has specific use cases, strengths, and limitations, particularly when it comes to real-time anomaly detection in big data environments.

In this section, we will discuss each machine learning approach in detail, including their application in real-time systems, and provide a comparative overview of their characteristics in the table below.

3.1 Supervised Learning for Anomaly Detection

Supervised learning requires labeled data, where the anomalies are pre-identified in the training dataset. The algorithm learns the patterns of normal and anomalous data and can classify new data points as either normal or anomalous. While supervised learning methods tend to achieve high accuracy in anomaly detection, their reliance on labeled data can be a significant limitation, particularly in environments where labeled data is scarce or difficult to obtain.

Common Supervised Learning Algorithms for Anomaly Detection:

a. **Support Vector Machines (SVM)**: Particularly effective for binary classification problems. One-class SVM is widely used for anomaly detection, where the model learns the decision boundary around normal data and flags any points outside this boundary as anomalies.
b. **Decision Trees and Random Forests**: These ensemble methods create multiple decision trees and aggregate their outputs. Random forests are useful in handling large datasets and capturing complex patterns.
c. **Neural Networks**: Deep learning models, such as auto encoders, can reconstruct data from a compressed representation and detect anomalies by measuring reconstruction errors.

Advantages:

a. High accuracy when labeled data is available.
b. Well-suited for structured datasets.

Challenges:

a. Requires a large amount of labeled data.
b. Not scalable in dynamic real-time systems without continuous retraining.

3.2 Unsupervised Learning for Anomaly Detection

Unsupervised learning techniques do not require labeled data. Instead, these methods work by identifying patterns and structures in the data, detecting instances that deviate significantly from the norm. Unsupervised learning methods are particularly well-suited for big data environments where labeled data is scarce or unavailable.

Common Unsupervised Learning Algorithms for Anomaly Detection:

a. **Clustering (e.g., K-Means, DBSCAN)**: These algorithms group similar data points together. Data points that do not belong to any cluster are considered anomalies.
b. **Principal Component Analysis (PCA)**: PCA reduces the dimensionality of the data while retaining the variance, and anomalies are identified as data points that deviate significantly from the principal components.
c. **Isolation Forest**: An ensemble technique that isolates anomalies by randomly partitioning the data and measuring how easily a point can be separated from the rest of the data.

Advantages:

a. No need for labeled data.
b. Effective for high-dimensional and unstructured data.

Challenges:

a. High false positive rate due to noisy data.
b. May struggle with detecting anomalies in highly dynamic environments.

3.3 Comparative Overview of Machine Learning Approaches

The table below provides a summary of the strengths and limitations of each machine learning approach for real-time anomaly detection in big data environments.

Approach	Data Requirements	Common Algorithms	Advantages	Challenges
Supervised Learning	Labeled data required	SVM, Random Forests, Neural Networks	High accuracy, well-suited for structured data	Requires large labeled datasets, limited scalability
Unsupervised Learning	No labeled data needed	K-Means, DBSCAN, PCA, Isolation Forest	No need for labeled data, effective for high-dimensional data	High false positive rates, struggles with dynamic data
Semi-Supervised Learning	Small labeled data, large unlabeled data	One-Class SVM, Autoencoders	Effective with limited labeled data, balances between supervised and unsupervised	Sensitive to small amounts of labeled data, noise

3.4 Integration of Machine Learning with Real-Time Processing Frameworks

To enable real-time anomaly detection, machine learning models need to be integrated with stream processing technologies. These technologies allow continuous data processing with low latency, enabling the detection of anomalies as data is generated. Several stream processing frameworks, such as **Apache Kafka**, **Apache Flink**, and **Apache Spark Streaming**, are commonly used to support real-time anomaly detection systems by handling the volume, velocity, and variety of big data.

Apache Kafka enables the efficient streaming of data, while Apache Flink and Apache Spark Streaming provide real-time data processing and integration with machine learning models for anomaly detection. These frameworks ensure that anomaly detection can occur in real time, with minimal delay, while maintaining scalability.

4. REAL-TIME ANOMALY DETECTION FRAMEWORKS

In a real-time big data environment, anomaly detection systems need to process vast amounts of data as it flows continuously, often from multiple sources. This requires a framework that can support low-latency detection, handle large volumes of streaming data, and integrate seamlessly with machine learning models for anomaly detection. Real-time anomaly detection frameworks are designed to address these needs, ensuring that abnormal patterns are identified and acted upon as soon as they occur. This section discusses several key real-time anomaly detection frameworks, highlighting their features, capabilities, and typical use cases.

4.1 Apache Kafka

Apache Kafka is a distributed event streaming platform widely used in real-time data processing. It is designed to handle high throughput, fault tolerance, and scalability, making it a suitable choice for anomaly detection in big data environments.

Key Features:

a. **Scalability**: Kafka supports massive scalability, able to handle millions of messages per second, making it well-suited for real-time applications.
b. **Fault Tolerance**: Kafka's distributed architecture ensures that data streams are fault-tolerant, with the ability to recover from failures without data loss.
c. **Stream Processing Integration**: Kafka can integrate with stream processing engines such as Apache Flink, Apache Storm, and Apache Spark, which are commonly used for real-time anomaly detection.

Real-Time Anomaly Detection Use Case:

a. **Fraud Detection in Financial Transactions**: Kafka can stream financial transaction data in real-time to detect anomalies, such as fraudulent transactions, by integrating with machine learning models that continuously analyze incoming data for irregular patterns.
b. **Network Intrusion Detection**: Kafka can process real-time network traffic and feed data into anomaly detection models that identify abnormal patterns of activity, potentially indicating a cyberattack.

4.2 Apache Flink

Apache Flink is a powerful, distributed stream processing framework that supports stateful computations and low-latency processing. It is designed for real-time data analytics and can be used for real-time anomaly detection in big data environments.

Key Features:

a. **Stateful Stream Processing**: Flink supports maintaining state across event streams, which is critical for tracking data trends over time and detecting anomalies based on historical patterns.
b. **Low Latency**: Flink is optimized for low-latency processing, ensuring that anomalies

can be detected and acted upon with minimal delay.

c. **Integration with Machine Learning**: Flink can integrate with machine learning models using libraries like FlinkML or external libraries, enabling advanced anomaly detection techniques.

Real-Time Anomaly Detection Use Case:

a. **IoT Sensor Monitoring**: In IoT networks, Flink can process real-time sensor data streams (e.g., temperature, pressure) and use machine learning models to identify anomalies that may indicate equipment failure, thereby triggering proactive maintenance.

b. **Predictive Maintenance in Industrial Systems**: Flink's low-latency capabilities make it ideal for monitoring industrial equipment, detecting abnormal behavior early, and preventing costly breakdowns.

4.3 Apache Spark Streaming

Apache Spark Streaming is a micro-batch stream processing system that extends Apache Spark’s capabilities to real-time data processing. It processes data in small time intervals or "micro-batches," allowing it to handle high-volume data streams efficiently.

Key Features:

a. **Micro-Batch Processing**: Spark Streaming processes incoming data in small, fixed-size batches, making it ideal for applications where data arrives in bursts, such as log processing or user activity tracking.

b. **Integration with MLlib**: Apache Spark includes MLlib, a scalable machine learning library, which allows users to apply anomaly detection algorithms directly to streaming data.

c. **Scalability**: Spark can scale horizontally by adding more nodes to the cluster, ensuring that it can handle very large data volumes without sacrificing performance.

Real-Time Anomaly Detection Use Case:

a. **Real-Time Log Analysis**: Spark Streaming can be used to analyze server logs in real time to detect anomalies, such as sudden spikes in error messages or unexpected system behavior.

b. **Online Recommender Systems**: Spark Streaming can process user activity data from online platforms and detect anomalies that could indicate fraud, such as sudden changes in user behavior or abnormal traffic patterns.

4.4 Apache Storm

Apache Storm is a real-time stream processing system designed for low-latency and high-throughput data processing. It is highly scalable and fault-tolerant, making it suitable for complex, real-time data processing tasks like anomaly detection.

Key Features:

a. **Real-Time Processing**: Storm processes data in real time, providing low-latency anomaly detection and enabling quick reactions to abnormal events.

b. **Distributed Topology**: Storm enables distributed processing through its topology of spouts and bolts, making it highly scalable and fault-tolerant.

c. **Integration with Machine Learning**: Storm can integrate with machine learning libraries or external systems to analyze real-time data streams for anomalies.

Real-Time Anomaly Detection Use Case:

a. **Social Media Sentiment Analysis**: Storm can process real-time social media data to detect anomalies in public sentiment, such as sudden spikes in negative sentiment that may indicate a public relations crisis or a potential threat.

b. **Real-Time Customer Behavior Tracking**: Storm can track customer interactions on e-commerce sites and detect anomalies, such as unusually high cart abandonment rates, which could signal a potential issue with the website or payment system.

4.5 Summary of Real-Time Anomaly Detection Frameworks

The table below summarizes the features of the discussed real-time anomaly detection frameworks:

Framework	Key Features	Use Case Examples
Apache Kafka	Scalable, fault-tolerant, integrates with stream processing tools	Fraud detection, network intrusion detection
Apache Flink	Stateful processing, low-latency, integration with ML libraries	IoT sensor monitoring, predictive maintenance
Apache Spark Streaming	Micro-batch processing, scalable, MLlib integration	Real-time log analysis, online recommender systems
Apache Storm	Low-latency, real-time processing, distributed topology	Social media sentiment analysis, customer behavior tracking

Framework	Key Features	Use Case Examples
Google Cloud Dataflow	Unified batch and stream processing, integration with Google AI	Clickstream analysis, financial market monitoring

Real-time anomaly detection frameworks are critical for managing the flow of large volumes of data and detecting irregularities as they occur. Frameworks like Apache Kafka, Flink, Spark Streaming, Storm, and Google Cloud Dataflow provide robust solutions for processing data streams and enabling real-time anomaly detection. Each framework has its strengths and is suited for different use cases, depending on factors such as data volume, latency requirements, and integration with machine learning models. By leveraging these frameworks, organizations can enhance their ability to detect and respond to anomalies promptly, reducing risks and improving operational efficiency.

5. APPLICATIONS OF REAL-TIME ANOMALY DETECTION

Real-time anomaly detection plays a critical role across various industries, helping organizations quickly identify and mitigate unusual events, behaviors, or patterns that deviate from the norm. Below are some key applications of real-time anomaly detection in different domains:

Application Area	Description	Example Use Cases
Cybersecurity	Detecting security breaches, unauthorized access, and abnormal network behavior in real time.	Intrusion detection, DDoS attack detection, malware detection
Fraud Detection	Identifying fraudulent activities in financial transactions, online payments, and banking.	Credit card fraud, insurance claim fraud, identity theft detection
Network Monitoring	Monitoring network traffic to detect unusual patterns or potential cyberattacks.	Network intrusions, data exfiltration, network performance anomalies
Industrial IoT (IIoT)	Detecting faults or failures in industrial equipment based on real-time sensor data.	Predictive maintenance, machinery failure detection, energy consumption anomalies
Healthcare	Identifying abnormal patient health data in real time for early diagnosis and intervention.	Patient monitoring, disease outbreak detection, medical device anomalies
E-Commerce and Retail	Monitoring user activity on e-commerce platforms to identify abnormal behavior such as fraud or system malfunction.	Cart abandonment, price manipulation, inventory anomalies
Telecommunications	Real-time detection of anomalies in network performance or customer usage patterns.	Network congestion, service disruption, unauthorized usage detection
Social Media	Monitoring social media platforms for sudden changes in sentiment or unusual activities.	Sentiment analysis, crisis detection, social media manipulation
Financial Markets	Analyzing financial data to identify unusual market movements or fraudulent trading activities.	Stock market anomalies, insider trading detection, high-frequency trading irregularities

These applications demonstrate the wide-ranging impact of real-time anomaly detection in various sectors, helping organizations to enhance security, improve operational efficiency, and reduce risks. By leveraging machine learning and real-time processing frameworks, these systems can quickly identify issues and trigger automatic responses, enabling faster decision-making and mitigating potential threats.

6. CHALLENGES IN REAL-TIME ANOMALY DETECTION

While real-time anomaly detection offers significant benefits, several challenges must be addressed to ensure its effectiveness in big data environments. These challenges stem from the complexity of handling vast amounts of data, ensuring the accuracy of anomaly detection models, and maintaining system performance. This section highlights the key challenges faced in real-time anomaly detection.

6.1 Data Volume and Velocity

Challenge: In big data environments, the sheer volume and velocity of incoming data can overwhelm detection systems. Real-time anomaly detection requires the ability to process large amounts of data quickly, which can strain computational resources and impact performance.

Impact: The detection system may experience latency issues or be unable to process the data fast enough to identify anomalies in real time. This can lead to missed opportunities for intervention, especially in critical applications such as fraud detection or cybersecurity.

Solution: Scalable architectures, such as distributed systems (e.g., Apache Kafka, Apache Flink), can be employed to handle high-volume and high-velocity data streams efficiently.

6.2 Data Quality and Noise

Challenge: The data being processed may contain noise, missing values, or inconsistencies, making it difficult to distinguish genuine anomalies from random fluctuations or errors. Poor data quality can lead to false positives (incorrectly identifying normal behavior as anomalous) and false negatives (failing to detect true anomalies).

Impact: False positives can lead to unnecessary alarms and resource wastage, while false negatives can result in undetected issues, such as fraud or security breaches, which could have serious consequences.

Solution: Data preprocessing techniques, such as data cleaning, normalization, and outlier detection, can help improve data quality. Moreover, employing robust machine learning models that can handle noise effectively, such as ensemble methods or deep learning approaches, can improve anomaly detection accuracy.

6.3 Model Complexity and Interpretability

Challenge: Machine learning models used for real-time anomaly detection, especially deep learning models, can be highly complex. These models may deliver high accuracy, but they are often seen as "black boxes" with limited interpretability. This lack of transparency can make it difficult for users to understand why an anomaly was detected, which is critical in high-stakes environments like healthcare or finance.

Impact: The inability to explain why a model flagged certain behavior as anomalous can erode trust in the system and hinder decision-making. In regulated industries, such as finance or healthcare, the lack of interpretability can also lead to compliance issues.

Solution: Techniques such as model explainability (e.g., LIME, SHAP) and the use of simpler, more interpretable models (e.g., decision trees or linear models) can help provide insights into how the model arrived at its predictions.

6.4 Real-Time Processing Latency

Challenge: Real-time anomaly detection systems must process incoming data streams with minimal latency to ensure timely identification of anomalies. However, achieving low-latency processing can be difficult, especially when dealing with large and complex datasets.

Impact: High latency can delay the detection of anomalies, rendering the system ineffective in environments where fast decision-making is critical, such as fraud detection or cybersecurity monitoring.

Solution: Optimizing the system architecture, utilizing low-latency processing frameworks (e.g., Apache Flink, Apache Kafka Streams), and employing distributed computing resources can help reduce processing latency.

6.5 Imbalanced Data

Challenge: Anomalies are, by definition, rare events, and in many real-world applications, datasets are highly imbalanced, with normal data instances vastly outnumbering anomalous ones. This imbalance can make it difficult for machine learning models to effectively learn the characteristics of anomalous data.

Impact: Models trained on imbalanced data may struggle to accurately detect anomalies, leading to a higher risk of false negatives (missed anomalies) and reduced overall detection performance.

Solution: Techniques such as oversampling, undersampling, or using synthetic data (e.g., SMOTE) can help balance the dataset. Additionally, anomaly detection algorithms that focus on detecting rare events, such as one-class SVM or autoencoders, can be more effective in imbalanced scenarios.

Real-time anomaly detection systems face a variety of challenges, ranging from handling large data volumes to ensuring the accuracy and interpretability of machine learning models. Overcoming these challenges requires a combination of advanced techniques in machine learning, system design, and data processing. By addressing issues such as data quality, latency, and model adaptability, organizations can build more robust and efficient anomaly detection systems that provide actionable insights and timely responses to emerging threats.

7. CONCLUSION

Real-time anomaly detection is a critical component in the modern data-driven landscape, where organizations must quickly identify and respond to unusual events, system failures, or potential security threats. As big data environments continue to grow in complexity and scale, traditional methods of anomaly detection struggle to keep pace with the volume, velocity, and variety of data. Machine learning-based approaches offer a promising solution, enabling the automated detection of anomalies in real time with higher accuracy and efficiency than conventional methods.

However, deploying real-time anomaly detection frameworks presents several challenges, including handling large and dynamic datasets, ensuring scalability, minimizing false positives, and adapting to evolving data patterns. Overcoming these challenges requires the integration of advanced machine learning techniques such as incremental learning, deep learning, and ensemble methods, alongside robust data processing frameworks capable of handling high-throughput streams.

Despite these hurdles, the applications of real-time anomaly detection are vast and impactful, spanning industries like cybersecurity, fraud detection, healthcare, industrial IoT, and finance. These systems enable faster decision-making, improve security, and ensure the smooth functioning of critical infrastructure.

As technology continues to evolve, the development of more efficient, scalable, and interpretable anomaly detection models will be crucial. By focusing on overcoming the challenges highlighted in this paper, organizations can unlock the full potential of real-time anomaly detection, leading to enhanced risk management, operational efficiency, and better decision-making in an increasingly data-driven world.

REFERENCES

[1] . Kaushik Reddy Muppa, Analysis on Cyber Risk Exposures and An Evaluation of The Elements That Go into Being Ready to Deal with Cyber Threats, International Journal of Computer Engineering and Technology (IJCET), 15(3), 2024, pp. 12-20

[2] . Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19-31. https://doi.org/10.1016/j.jnca.2015.11.005

[3] . Venkat Nutalapati. A Comprehensive Review of Mobile App Security Testing Tools and Techniques. International Research Journal of Engineering & Applied Sciences (IRJEAS). 8(1), pp. 10-15, 2020.

[4] . Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1-58. https://doi.org/10.1145/1541880.1541882

[5] . Kaushik Reddy Muppa, Analysis on the Role of Artificial Intelligence and Identity and Access Management (IAM) In Cyber Security, International Journal of Artificial Intelligence Research and Development (IJAIRD), 2(1), 2024, pp. 113-122. DOI 10.17605/OSF.IO/76DG5.

[6] . Iglewicz, B., & Hoaglin, D. C. (1993). How to detect and handle outliers. SAGE Publications. ISBN 978-0761904061

[7] . Kshetri, N. (2017). 1. Cybersecurity in the Age of Big Data: Implications for Businesses. International Journal of Information Management, 37(1), 45-59. https://doi.org/10.1016/j.ijinfomgt.2016.10.003

[8] . Venkat Nutalapati. Intrusion Detection Systems for Embedded Android: Techniques and Performance Evaluation. International Research Journal of Engineering & Applied Sciences (IRJEAS). 7(4), pp. 18-25, 2019.

[9] . Lakhina, A., Crovella, M., & Diot, C. (2004). Minimizing the impact of network anomalies on real-time anomaly detection. Proceedings of the 3rd ACM SIGCOMM Workshop on Internet Measurement, 211-224. https://doi.org/10.1145/1028788.1028816

[10] . Kaushik Reddy Muppa, Study on Cloud-Based Identity and Access Management in Cyber Security, International Journal of Data Analytics Research and Development (IJDARD), 2 (1), 2024, pp. 40–49. DOI 10.17605/OSF.IO/J93FR.

[11] . Luo, J., & Huang, Y. (2018). Deep learning-based anomaly detection: A survey. ACM Computing Surveys (CSUR), 51(6), 1-41. https://doi.org/10.1145/3189857

[12] . Mahmood, A. N., & Ganaie, M. A. (2021). A review of machine learning techniques for anomaly detection in network traffic. Computers, 10(1), 25. https://doi.org/10.3390/computers10010025

[13] . Palaniappan, S., & Lee, M. (2008). Anomaly detection in data mining: A survey. The International Journal of Computer Science and Applications, 5(3), 1-12.

[14] . Kaushik Reddy Muppa, Advancing Cloud Security with AI-Enhanced AWS Identity and Access Management, International Research Journal of Engineering & Applied Sciences (IRJEAS). 10(1), pp. 25-08, 2022.

[15] . Venkat Nutalapati. Enhancing Security through Dynamic Analysis in Embedded Android Systems. International Research

Journal of Engineering & Applied Sciences (IRJEAS). 8(4), pp. 29-35, 2020.

[16] . Qiao, J., Wang, X., & Liu, H. (2018). Real-time anomaly detection using machine learning: A survey. Journal of Big Data, 5(1), 1-18. https://doi.org/10.1186/s40537-018-0146-x

[17] . Sculley, D., Holt, G., & O'Callaghan, S. (2015). Hidden technical debt in machine learning systems. Proceedings of the 28th International Conference on Neural Information Processing Systems, 2503-2511. https://doi.org/10.1145/2981748.2981767

[18] . Sahoo, S., & Murugesan, R. (2020). Anomaly detection in time-series data using machine learning techniques. Machine Learning and Applications: An International Journal (MLAIJ), 7(1), 21-34.

[19] . Kaushik Reddy Muppa, Optimizing Security in the Cloud: Strengthening Protection Through Single Sign-On Implementation. International Research Journal of Engineering & Applied Sciences (IRJEAS). 11(2), pp. 01-03, 2023.

[20] . Venkat Nutalapati. Implementing End-to-End Encryption in Mobile Applications: Challenges and Solutions. International Research Journal of Engineering & Applied Sciences (IRJEAS). 9(2), pp. 29-33, 2021.

[21] . Zimek, A., Schubert, E., & Kriegel, H.-P. (2012). A survey on unsupervised anomaly detection. Proceedings of the International Conference on Data Mining, 31, 1-20. https://doi.org/10.1109/ICDM.2012.161

[22] . Cingireddy, A. R., Ghosh, R., Melapu, V. K., Joginipelli, S., & Kwembe, T. A. (2022). Classification of Parkinson's Disease Using Motor and Non-Motor Biomarkers Through Machine Learning Techniques. *International Journal of Quantitative Structure-Property Relationships (IJQSPR), 7*(2), 1-21. https://doi.org/10.4018/IJQSPR.290011

[23] . Zhang, C., & Zhao, Y. (2017). Machine learning-based real-time anomaly detection in network traffic. International Journal of Network Management, 27(6), e1920. https://doi.org/10.1002/nem.1920

[24] . Venkat Nutalapati. Secure Coding Practices in Mobile App Development. International Research Journal of Engineering & Applied Sciences (IRJEAS). 10(1), pp. 29-34, 2022. 10.55083/irjeas.2022.v10i1010

[25] . Zhao, H., & Wei, Z. (2019). Real-time anomaly detection for data streams using machine learning techniques. International Journal of Computer Applications, 177(13), 31-39. https://doi.org/10.5120/ijca2019919060

Conflict of Interest Statement: *The authors declare that there is no conflict of interest regarding the publication of this paper.*

Cite this Article

Anurag Shrivashtav, Amit Kushwaha. Machine Learning-Based Frameworks for Real-Time Anomaly Detection in Big Data Environments. International Research Journal of Engineering & Applied Sciences (IRJEAS). 13(1), pp. 29-38, 2025. 10.55083/irjeas.2025.v13i01004