SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Robust Anomaly Detection In Large-Scale Streaming Data Systems Using Deep Learning

Haripriya V[1], Ankita Gandhi[2], Aranganathan A[3], Saraswati B[4], Jeevajothi R[5], S. Balaji[6]

[1] Assistant Professor, Department of CS & IT, JAIN (Deemed-to-be University), Bengaluru, Karnataka, India. Email: V.haripriya@jainuniversity.ac.in, ORCID: 0000-0003-2035-2452
[2] Assistant Professor, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India. Email: ankita.gandhi@paruluniversity.ac.in, ORCID: 0000-0002-3516-2187
[3] Associate Professor, Department of Electronics and Communication Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India. Email: aranganathan.etc@sathyabama.ac.in, ORCID: 0000-0002-5986-6031
[4] Assistant Professor, Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: saraswatib@maher.ac.in
[5] Assistant Professor, Department of Management Studies, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: rjeevajothimba@maher.ac.in
[6] Professor, Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, Tamil Nadu, India. Email: balajiit@gmail.com, ORCID: 0000-0003-3208-1637

Abstract
Streaming data generated from sensors, Internet of Things (IoT) devices, and digital platforms demands continuous monitoring, yet identifying anomalous patterns in high-velocity environments remains challenging due to evolving data distributions and scale. Existing approaches often struggle with adaptability and robustness. This research aims to design a robust deep learning (DL)model for anomaly detection in large-scale streaming systems. A total of 6543 data points were gathered from the open source Streaming Anomaly Dataset, followed by Z-score normalizationand feature extraction using Principal Component Analysis (PCA). The proposed Billiards Optimizer-driven Adaptive Convolutional Neural Network (BO-ACNN) model employs the Billiards Optimizer to fine-tune hyperparameters and enhance convergence efficiency, while the Adaptive CNN dynamically adjusts convolutional filters to capture evolving temporal–spatial features. This combination enables precise detection of irregular patterns in streaming environments. The model effectively identifies anomalies in large-scale data streams by learning complex patterns and adapting to distribution shifts. Experimental evaluation shows better performance in detection rate (98.60%), accuracy (98.90%), false positive rate (1.79%), recall (97.26%), precision (96.45%), F1-score (97.85%), ROC-AUC (98.45%), and latency (20ms) compared to conventional approaches, which were implemented in Python. The approach ensures scalable, adaptive, and reliable anomaly detection, making it suitable for intelligent monitoring applications.

Keywords: Anomaly Detection, Streaming Data Analytics, Large-Scale Data, Monitoring, Adaptive Learning.

1. Introduction

The enormous streaming data that has been introduced since individuals started utilizing the internet to a greater degree led to the emergence of time window models that comprise sliding windows, landmark windows, and damped windows [1]. The established techniques that can be used to identify the presence of anomalies in streaming data were also challenged when they tried to deal with large amounts of industrial data due to the unpredictable nature of data streams and the underlying distribution is difficult to model [2]. More robust techniques were thus needed to adequately detect anomalies, particularly in industrial scenarios where timely detection of anomalous behaviors could prevent failures and achieve optimal performance of the system. To further improve the detection of anomalies in the case of large-scale streaming data systems with the help of DL methods, it is necessary to guarantee the proper and efficient monitoring that would contribute to the improved maintenance and minimized downtime in the industrial setting [3]. Robust anomaly detection in large-scale streaming data systems was performed using AI-based models. The data analysis methods from AI-

enabled platforms, which operate in real time, make it possible to find unusual trends through historical data and system behavior knowledge [4]. It proved significant because standard supervised methods failed to work and acquiring labeled logs proved difficult. The models showed strong performance because they successfully tracked both time-based and sequential developments in system log data and IoT device monitoring data. The researchers achieved accurate detection through their work, which optimized DL algorithms for latency reduction [5]. The online learning capability of DL methods enables models to adjust continuously to new data streams while maintaining current knowledge about system changes over time [6]. The computational demands of DL models stem from their need for extensive resources to execute both training and inference processes. The systems need extensive labeled data, which they cannot access while facing difficulties to identify anomalies in environments where data patterns change constantly [7].

Research aim: The researchers aim to create a deep learning framework, which uses the BO-ACNN model to achieve accurate real-time anomaly detection in large-scale streaming data systems. The research aims at improving scalability, detection accuracy, and being able to capture dynamic temporal-spatial patterns.

Research organization: The research framework consists of two main parts, which include background information about the study in Section 1. Section 2 and 3 provide a summary of existing research about anomaly detection and DL methods, data processing and proposed BO-ACNN method. Section 4 and 5 provide findings and their subsequent analysis and conclusions.

2. Related Works

Table 1 presents a summary of current anomaly detection methods, which use ML and DL techniques to address various domains because it shows their research goals and methodological approaches and their achieved results.

Table 1. Summary of Existing Anomaly Detection Methods and Limitations

Ref	Objective	Method	Result	Limitations
[8]	Identify abnormalities in LSSPV power stations	K-Means clustering coupled with LSTM	Better results compared to ANN; lower maintenance cost and shorter duration	Scalability and generalizability were not addressed
[9]	Enhance electrical load anomaly detection using edge computing	Edge Industrial Unit Detector (EIUD) with unsupervised clustering	Achieved high accuracy with approximately 1 ms processing time per data point	Strong dependency on edge devices; scalability challenges
[10]	Improve anomaly detection in Log-based Cloud Systems (LCS)	ML-based anomaly detection on telemetry data	Improved anomaly detection performance	Data complexity and scalability issues remain
[11]	Address anomaly detection gaps in Big Data quality assessment	Quality Anomaly Score (QAS) using six quality parameters	Provided a unified anomaly detection model across domains	High computational complexity and scalability issues with large datasets
[12]	Develop secure anomaly detection in Industrial Internet of Things (IIoT)	Isolation Forest (IF) integrated with Transport Layer Security (TLS) in edge/cloud environments	Achieved effective anomaly detection with secure data transmission	Security robustness and scalability concerns in heterogeneous deployments
[13]	Recommend federated learning for anomaly detection in large IoT deployments.	Federated Learning (FL) coupled with TCN-ACNN (FedLog)	Outperform DeepLog and LogAnomaly under centralized and federated architectures	Issues with non-IID data and scalability
[14]	Review Graph Anomaly Detection (GAD) using GNNs	Taxonomy of 13 Graph Neural Networks (GNN)-based methods	Delivered comprehensive insights and comparative performance evaluation	Several open research challenges remain unresolved
[15]	Perform anomaly detection using deep	VGG-based CNN architecture	Achieved good detection accuracy	High computational cost, poor scalability, and limited

	convolutional neural networks			adaptability for streaming data
[16]	Detect anomalies using conventional ML techniques	Traditional ML-based anomaly detection methods	Moderate performance with lower accuracy compared to DL approaches	Ineffective for high-dimensional data, poor scalability, and inability to capture temporal dependencies

3. Methodology

The proposed solution uses datasets collected from benchmark sources or IoT devices, where the data can have normal or unusual streaming. Z-score normalization is used as a preprocessing method to standardize values. PCA is used for feature extraction to minimize noise and dimensionality. The BO-ACNN model receives the processed data, where the BO adjusts hyperparameters, and an Adaptive CNN finds patterns. Figure 1 shows the workflow architecture of BO-ACNN for Robust Anomaly Detection in Large-Scale Streaming Data Systems.

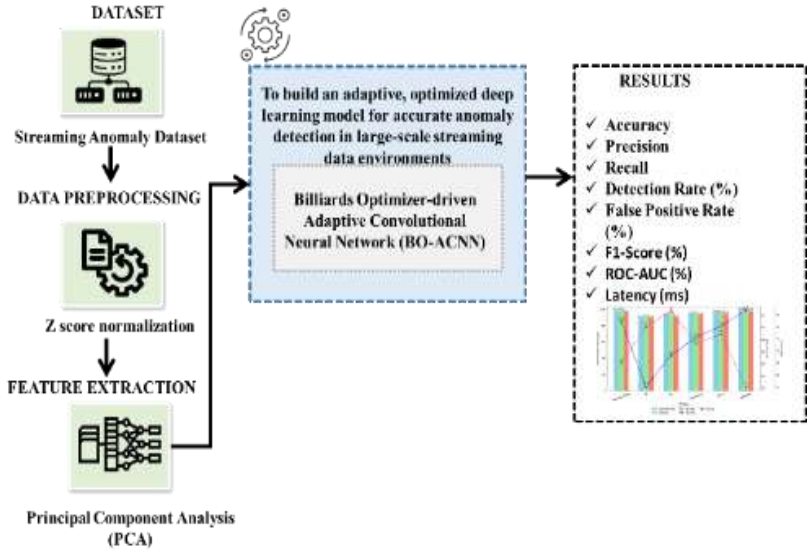

Figure 1: DL-Driven Workflow for Adaptive Anomaly Detection in High-Volume Data Streams

Data collection: The dataset consists of 6,543 time-stamped multivariate observations of interconnected devices that record the environmental conditions, system performance, and network behavior. It has 13 feature columns, including temperature, pressure, humidity, CPU and memory usage, network latency, packet loss, throughput, error rate, voltage level, vibration signal, a device ID, and a timestamp. All records are the synchronized measurements at a certain time point. Both normal and abnormal system operation are captured in the dataset, thus it is possible to analyze the dynamic behavior of the system. The target column classifies each instance as either normal or anomalous, making it suitable for anomaly detection and predictive modeling tasks.

Kaggle Source:https://www.kaggle.com/datasets/colabsss/real-time-streaming-anomaly-dataset
Data feature exploration: Figure 2(a) illustrates the temporal variation of CPU usage, memory consumption, and throughput, which highlights dynamic changes and potential anomalies. Figure 2 (b) depicts feature connections and separability between normal and anomalous data.

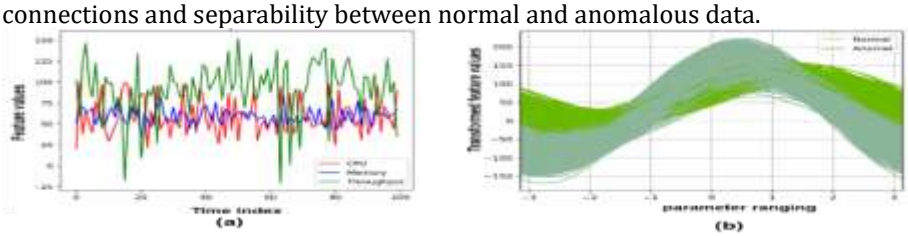

Figure 2: Streaming anomaly detection (a) Streaming Metrics (b) Normal and Abnormal Patterns

The characteristics of raw data before preprocessing, highlighting system behavior and demonstrating the need for the proposed method to improve anomaly detection in large-scale streaming environments. Figure 3(a) shows a normal distribution pattern of memory usage, which shows that system performance, remains stable with only minor fluctuations. Figure 3(b) shows paired value comparisons between different indices, which show both existing differences, and possible anomalous data points.

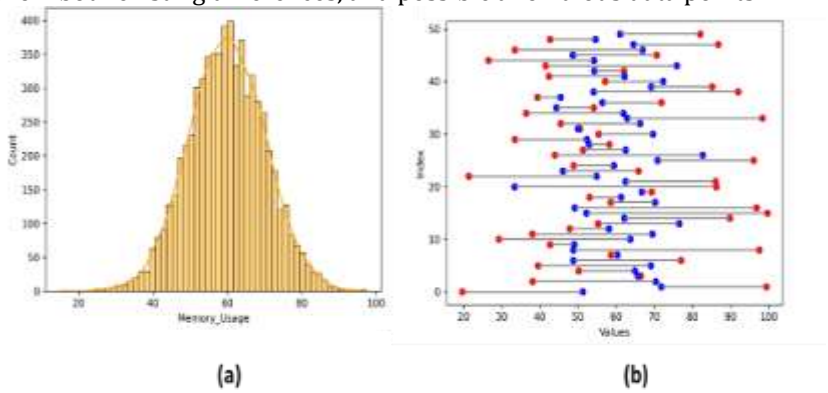

Figure 3: Streaming data patterns and feature distributions (a) Memory Consumption (b) Comparative Variation

3.2 Data preprocessing using Z-score normalization

The research project intends to develop an anomaly detection system, which uses Z-score normalization to monitor extensive streaming data streams. The goal of the project requires researchers to establish uniform data processing standards by transforming all data elements to a common scale with a mean value of 0 and a standard deviation of 1. This normalization enhances the data quality as well as the stability and consistency of the models in dynamic environments. Reducing variability among features using the model improves the detection of irregularities and anomalies in data streams. Equation (1) illustrates how training data statistics are used to normalize both training and test data for streaming datasets:

$$Y - Scored\ EMG_{s,c,t} = (EMG_{s,c,t} - \mu_{train,t})/\sigma_{train,t} \quad (1)$$

In Equation (1), t represents the current discrete time step in the streaming data, c indicates whether the data belongs to the training or test set, and s denotes the subject, sensor, or data source.$EMG_{s,c,t}$ is the original signal or feature value at time t, while $Scored\ EMG_{s,c,t}$is the normalized signal after applying the Z-score.$\mu_{train,t}$and $\sigma_{train,t}$are the mean and standard deviation calculated using the training set of subject Y.

3.3 Extracting the feature using PCA

To enhance the process of detecting anomalies in large data streams, a two-stage supervised model employing PCA is presented. First, the contaminated data is classified by means of binary classification. The second step involves finding the timestamp where the anomaly occurs by means of a multi-class model. This process iterates until all the anomalies are detected and corrected.

$$W = (W^1, W^2, \dots, W^j, \dots, W^m) \quad (2)$$

Equation (2) defines the complete dataset W as a collection of n streaming instances. W^j controls how much influence a specific feature has. In large-scale streaming environments, data arrives continuously, and each W^m represents a snapshot or time window of system behavior.

$$W^j = \left(w^j_{s_1}, w^j_{s_2}, \dots, w^j_{s_i}, \dots, w^j_{s_o}\right) \in \mathbb{R}^o \quad (3)$$

Equation (3) represents each streaming instance as a multivariate time series over s timestamps,means belongs to, and $\mathbb{R}^o$means an o-dimensional valued vector space. Each value $w^j_{s_o}$ captures system behavior at a specific time.

$$B^j = \begin{cases} 1 & if\ there\ exists\ i\ such\ that\ w^j_{s_i}\ is\ an\ anomaly, \\ 0 & otherwise \end{cases} \quad (4)$$

In equation (4), $B^j$is is the contamination label for the j-th time series: 1 indicates an anomaly exists, otherwise 0.

3.4 Anomaly detection in Large-Scale Streaming Data using BO-ACNN

To develop an ACNN-BO method for robust anomaly detection in large-scale streaming data systems. The ACNN dynamically extracts evolving temporal and spatial features from high-velocity data streams. BO Algorithm is an effective algorithm to optimize network parameters with collision, exploration and memory-driven exploration. The combined system ensures stability, high accuracy and resistance to high dimensional, noisy streaming data. The solution enhances responsiveness and precision with complex anomaly detection scenarios.

ACNN Module for Robust Anomaly Detection: To create a scalable anomaly detection ACNN module that can be effectively used to detect anomalies in large-scale streaming data systems. The module contains convolutional filters which are dynamically changed to suit the changing data patterns. It integrates the time and space features extraction in order to enhance detection of big and small feature anomalies. The ACNN takes advantage of streaming normalization and adaptive learning to stabilize high-dimensional space data which improving the accuracy, robustness and responsiveness of anomaly detection.

$$\mathcal{P}_l = \begin{bmatrix} w_1 & z_1 & 1 \\ w_2 & z_2 & 1 \\ \vdots & \vdots & \vdots \\ w_M & z_M & 1 \end{bmatrix} \quad (5)$$

In Equation (5), the samples are arranged into a matrix $_{,pl}$us features $w, z,$ and a bias term of 1 across M observations.

$$\mathcal{P}_l^s = \mathcal{P}_l \mathcal{M} \quad (6)$$

In Equation (6), the augmented representation $P_l^s$is is computed by multiplying the matrix P_l with M, thereby scaling the feature space for normalization and ensuring consistent dimensional transformation.

$$\mathcal{D}_b = \{P_1, \ldots, P_L\} = \cup_{l=1}^{L} \cup_{j=1}^{3} \mathcal{T}_j (J_l) \quad (7)$$

In Equation (7), the final augmented dataset $\mathcal{D}_b$ is formed by applying the transformations T_1, $T2, T_3$on J_l, generating expanded samples for robust learning through a union operation. Here, $\mathcal{D}_b$ denotes the dataset, P_L represents the $l - th$ sample, and union denotes their operation.

BO for Robust Anomaly Detection: The C-BO (Chaotic) system together with its DL models and BO method creates an advanced anomaly detection technology for handling large-scale streaming data systems. The ball–pocket interaction method enables the DL models to explore and utilize their feature space through their combination with BO strategy. The three equations 8 through 12 provide instructions for establishing setup procedures and evaluating system performance together with creating adaptive solution movement strategies. The method increases convergence success rates while stopping local minimum issues during model optimization. It enhances accuracy, stability, and resiliency to anomalies detection in streaming data settings. The algorithm determines the best parameters using its solution evaluation mechanism that involves an update of the ball positions and control of boundaries without a lot of exploration and exploitation.

$$A_{K,L}^{old} = rand_{[0,1]}\left(Var_L^{max} - Var_L^{min}\right) - Var_L^{min} \quad (8)$$

In Equation (7), $rand_{[0,1]}$ generates random numbers, and the initial solution matrix $A_{K,L}^{old}$ is created using these values scaled between the variable bounds Var_L^{min} and Var_L^{max} for population initialization.

$$T_j = \frac{f^{-\gamma E_j}}{\sum_j f^{-\gamma E_j}} \quad j = 1,2,3,\ldots \quad (9)$$

The selection probability T_j is computed using exponential fitness weighting $f^{-\gamma E_j}$, where E_j represents the error and γcontrols the selection pressure. The term $\sum_j$ denotes the aggregation over all error values associated with state j.

$$O = \frac{iter}{iter_n} \quad (10)$$

The iteration control factor O is defined as the ratio of the current iteration to the maximum number of iterations $iter_n$, guiding the convergence progress of the optimization process.

$$A_{K,L}^{new} = rand_{[-ER,ER]}(1 - O)\left(A_{K,L}^{old} - T_{K,L}^{new}\right) - T_{j,L}^{K} \quad (11)$$

The solution $A_{K,L}^{new}$ is updated using a random perturbation, where $rand_{[-ER,ER]}$generates a uniformly distributed random number within the range to introduce stochasticity. The convergence factor O regulates the update process. Here, $A_{K,L}^{old}$ denotes the previous value of the parameter, $T_{K,L}^{new}$ represents the updated target value at position (K, L), and the difference between the old solution and the target T guides the adjustment. Additionally, $T_{j,L}^{K}$ denotes the interaction or contribution term from index j, layer L, and state K.

$$A_{K+M}^{new} = \frac{\overrightarrow{u'_{K+M}}}{2b} u'_{K+M} + \overrightarrow{A_K^{old}} \quad (12)$$

The final solution $A_{K+M}^{new}$is is obtained by combining the normalized transformed vector with the previous state, balancing exploration and exploitation. Here, $\frac{\overrightarrow{u'_{K+M}}}{2b} u'_{K+M}$ represents the scaled self-interaction term that promotes exploitation, while $\overrightarrow{A_K^{old}}$ denotes the previous vector (best solution) at position K, contributing to exploration.

Algorithm 1:BO-ACNN for robust anomaly detection

1. ***Input***: *Streaming dataset WWW, normalization parameters* $\mu, \sigma \backslash mu, \backslash$ *sigma*μ, σ, *maximum iterations iterniter_nitern, threshold* $\theta \backslash theta\theta$
2. *Load streaming dataset WWW*
3. *Normalize data using* $Z -$ *score normalization*
4. *Apply PCA to extract feature set WjW^{j}Wj*
5. *Initialize ACNN model* (*convolution layers, filters, parameters*)
6. *Initialize BO optimization and generate initial solution AoldA^{old}Aold*
7. *Evaluate fitness EoldE_{old}Eold using ACNN performance*
8. ***For*** $iter = 1iter = 1iter = 1$ *to iterniter_nitern* ***do***
9. *Update BO solution AnewA^{new}Anew*
10. *Train ACNN using AnewA^{new}Anew and compute fitness EnewE_{new}Enew*
11. ***If*** $Enew < EoldE\_\{new\} < E\_\{old\}Enew < Eold$ ***then***
12. $Update\ Aold = AnewA\^{}\{old\} = A\^{}\{new\}Aold = Anew$
13. $Update\ Eold = EnewE\_\{old\} = E\_\{new\}Eold = Enew$
14. ***Else***
15. *Retain previous solution AoldA^{old}Aold*
16. ***End If***
9. ***End For***
10. *Train final ACNN model using optimized parameters*
11. ***For each*** *data instance* $x \in Wx \in Wx \in W$ ***do***
12. *Predict output* y
13. ***If*** $y \geq \theta y \geq \backslash thetay \geq \theta$ ***then***
14. $Label = Anomaly$
15. ***Else***
16. $Label = Normal$
17. ***End If***
18. ***End For***
19. ***Output***: *Anomaly labels for each data instance* (***Anomaly*** / ***Normal***)

Algorithm 1 begins by initializing the ACNN model to learn temporal and spatial features from streaming data, following preprocessing and feature extraction. An iterative process of optimizing ACNN parameters with BO is then used to solve the fitness values and update solutions.

4. Result and Discussion

The Result section shows the performance evaluation of the proposed BO-ACNN model based on experimental analysis. This model is built using the Python programming language. The model is trained and tested on a Windows-based system equipped with an Intel® Core™ i7-10875H processor running at 2.30 GHz.

Detection Rate: This is a measure of the model's ability to detect real anomalies. The greater the value, the better the ability to detect abnormal events, which is important in reducing missed anomalies.

False Positive Rate: This is the rate of the number of times when normal cases are mistaken for anomalies. A smaller value is preferable to minimize unwanted alerts and overhead in the system.

Accuracy: The general accuracy of the model, which is the measure of the percentage of correctly classified instances. **Precision:** Measures the consistency of the prediction of anomalies, the number of anomalies identified that are actually anomalies. False alarms are minimized by high precision.

Recall: Determines how well the model is able to describe all the real anomalies. High recall implies that there are less critical anomalies that are ignored.

F1-Score: It is a balanced metric, assessing both false results and true results, which is suitable in those situations when it is necessary to check false positives and false negatives.
ROC-AUC: It measure assesses the degree to which the model is able to differentiate between normal and abnormal classes at various threshold levels.
Latency: The model must be able to process incoming data in a certain amount of time to be classified, whereas streaming environments require lower latency to reliably detect anomalies.

Performance evaluation based on an existing dataset: The BO-ACNN model performance assessment is carried out on existing dataset [15] VGG-Net-based DCNN to test it in a controlled experimental setting. The data corresponds to real network traffic patterns, which contain different cyber-attacks that happen in real life scenarios. Table 2 demonstrates that the BO-ACNN model improves the learning of features by optimizing the DL and augmenting features, which leads to high detection performance. It has a low false positive rate of 1.89 and an accuracy of 98.85%.

Table 2: Comparative Analysis of Present VGG-Net-Based DCNN, and Proposed BO-ACNN Model.

Method	Detection Rate (%)	False Positive Rate (%)	Accuracy (%)	Latency (ms)
VGG-Net-Based DCNN[15]	98.47	2	98.79	30
BO-ACNN[Proposed]	98.52	1.89	98.85	25

The proposed BO-ACNN model is promising to ensure the consistency and equal performance comparison between all the approaches in existing network traffic pattern data [16]. Conventional ensemble models like RF, Autoencoder (DL), LSTM, and GB have low effectiveness in dealing with temporal dependencies in the dataset network traffic patterns. Table 3 shows that BO-ACNN achieves superior feature learning of complex streaming anomaly patterns, achieving greater performance with 98.5% accuracy and a higher F1-score (97.7%).

Table 3: Comparative Evaluation of Classification Models for Anomaly Detection

Model	Accuracy (%)	Precision (%)	Recall (%)	F1- Score (%)	ROC-AUC (%)
RF [16]	91.2	89.8	87.5	88.6	92.1
GB[16]	93.5	91.2	89.4	90.3	94.7
Autoencoder (DL) [16]	95.1	93.8	92.3	93.0	96.2
LSTM [16]	96.4	94.7	93.9	94.3	97.1
BO-ACNN [Proposed]	98.5	96.3	97.1	97.7	98

Performance evaluation using the Streaming Anomaly Dataset: This research evaluates temporal and anomaly detection models using continuously generated data streams that include both normal and abnormal behavior. All existing models were retrained on the streaming anomaly dataset to ensure a fair comparison. The proposed BO-ACNN model outperforms these methods, achieving an accuracy of 98.85% and a detection rate of 98.52% are presented in Table 4 and Figure 4.

Table 4: Evaluation of Existing Methods and Proposed BO-ACNN for Anomaly Detection

Method	Detection Rate (%)	False Positive Rate (%)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)	Latency (ms)
VGG-Net-Based DCNN	98.47	2.00	98.79	95.6	96.8	96.2	97.5	30
RF	89.2	7.8	91.2	89.8	87.5	88.6	92.1	45
GB	92.6	6.1	93.5	91.2	89.4	90.3	94.7	52
Autoencoder (DL)	95.1	4.3	95.1	93.8	92.3	93.0	96.2	38
LSTM	96.4	3.5	96.4	94.7	93.9	94.3	97.1	42
BO-ACNN [Proposed]	98.60	1.79	98.90	96.45	97.26	97.85	98.45	20

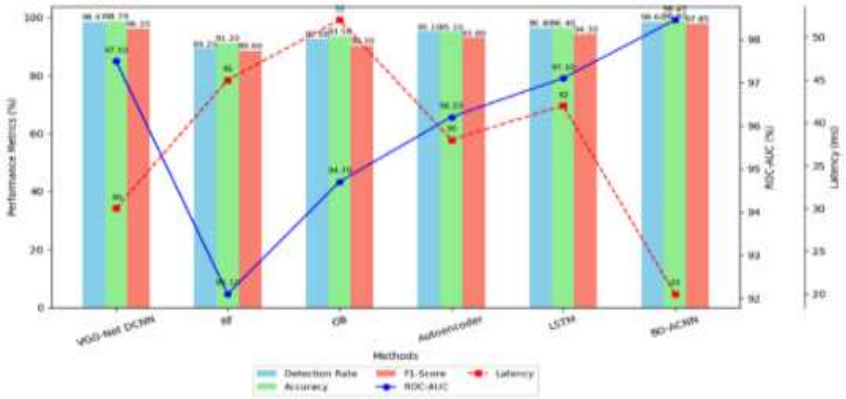

Figure 4: Graphical representation of various metrics comparison achieved in the proposed data

The proposed BO-ACNN model demonstrates superior performance for real-time anomaly detection by effectively capturing dynamic temporal–spatial patterns and adapting to streaming data variations. The traditional models VGG-Net-based DCNN [15], RF [16], GB [16], Autoencoder [16], and LSTM [16] face multiple limitations, which include their inability to adapt quickly and their slow processing speed and their limited capability to learn from changing time intervals and their inability to manage developing data patterns. The BO-ACNN system achieves its objectives by using two methods, which are adaptive convolutional learning and Billiards Optimizer-based hyperparameter tuning to achieve better accuracy and system reliability and quick detection results.

4. Conclusion

The research focuses on developing efficient methods to detect unusual activities in extensive continuous data acquisition systems, which track changes in operational environments. The BO-ACNN model achieves high detection efficiency through its 98.60% detection rate, 98.90% accuracy, 96.45% precision, 97.26% recall, 97.85% F1-score, and 98.45% ROC-AUC measurement. The system also demonstrates a 1.79 false positive rate and maintains operational performance with 20-millisecond response time. The model demonstrates successful performance in anomaly detection because it can handle multiple types of streaming data and it maintains performance when data patterns change. The solution provides dependable smart monitoring capabilities with efficient predictive maintenance tools, which work in fast-moving operational settings. The model requires high processing power for its operations but becomes less efficient when handling datasets that contain many dimensions or show constant changes. The system struggles to manage several various data streams which have different sources. The research develops effective systems outlines, which can facilitate the instantaneous educational progress and facilitate gathering of data through different sources whilst being in a position to handle a large number of real-life systems.

Reference

1 Qiu, Z., 2023. A Multi-Scale Deep Learning and Uncertainty Estimation Framework for Comprehensive Anomaly Detection in Cloud Environments. Transactions on Computational and Scientific Methods, 3(2). https://doi.org/10.5281/zenodo.18278012

2 Zhang, H., Jia, X., Chen, C., Bachani, S., Goel, J.K., Tarun, M., Mahmood, H.A.R., Abdullah, B.A., Talib, R.M., Nasir, N.A., and Merie, G.M.S., 2025. Deep learning-based real-time data quality assessment and anomaly detection for large-scale distributed data streams. International Journal of Medical and All Body Health Research, 6(1), pp.01-11.https://doi.org/10.54660/IJMBHR.2025.6.1.01-11

3 Sen, R.K., 2023. Detailed Process for Developing an Efficient Anomaly Detection Algorithm for Real-Time Streaming Data in Large-Scale Industrial Systems. Processes, 11(1), p.101.https://doi.org/10.55041/IJSREM23710

4 Dewi, D.A., Singh, H.K.R., Periasamy, J., Kurniawan, T.B., Henderi, H., and Hasibuan, M.S., 2024. Scalable Machine Learning Approaches for Real-Time Anomaly and Outlier Detection in Streaming Environments. Journal of Applied Data Sciences, 5(4), pp.1949-1962. https://doi.org/10.47738/jads.v5i4.444

5 Korba, A.A., Diaf, A., Bouchiha, M.A. and Ghamri-Doudane, Y., 2025. Mitigating IoT botnet attacks: An early-stage explainable network-based anomaly detection approach. Computer Communications, p.108270.https://doi.org/10.1016/j.comcom.2025.108270
6 Feng, Y., Cai, W., Yue, H., Xu, J., Lin, Y., Chen, J. and Hu, Z., 2022. An improved X-means and isolation forest-based methodology for network traffic anomaly detection. Plos one, 17(1), p.e0263423https://doi.org/10.1371/journal.pone.0263423
7 Moallemi, A., Burrello, A., Brunelli, D. and Benini, L., 2022. Exploring scalable, distributed real-time anomaly detection for bridge health monitoring. IEEE Internet of Things Journal, 9(18), pp.17660-17674. https://doi.org/10.1109/JIOT.2022.3157532
8 Zulfauzi, I.A., Dahlan, N.Y., Sintuya, H. and Setthapun, W., 2023. Anomaly detection using K-Means and long-short term memory for predictive maintenance of large-scale solar (LSS) photovoltaic plant. Energy Reports, 9, pp.154-158. https://doi.org/10.1016/j.egyr.2023.09.159
9 Li, C., Li, Y., Cao, Y., Zhang, Z., Wan, J., and Shahidehpour, M., 2025. Anomaly Detection of Cross-Operational State Streaming Data in an Edge Computing Platform. IEEE Transactions on Industry Applications.https://doi.org/10.1109/TIA.2025.3556663
10 Karshiyev, Z., Sattarov, M., and Erkinov, F., 2025. ADAPTIVE HYBRID ENSEMBLE FRAMEWORK FOR REAL-TIME ANOMALY DETECTION IN LARGE-SCALE DATA STREAMS. Techscience. uz-Texnikafanlariningdolzarbmasalalari, 3(12), pp.74-93. https://doi.org/10.47390/ts-v3i12y2025N09
11 Widad, E., Saida, E., and Gahi, Y., 2023. Quality anomaly detection using predictive techniques: an extensive big data quality framework for reliable data analysis. IEEE Access, 11, pp.103306-103318. https://doi.org/10.1109/ACCESS.2023.3317354
12 Bin Mofidul, R., Alam, M.M., Rahman, M.H. and Jang, Y.M., 2022. Real-time energy data acquisition, anomaly detection, and monitoring system: Implementation of a secured, robust, and integrated global IIoT infrastructure with edge and cloud AI. Sensors, 22(22), p.8980. https://doi.org/10.3390/s22228980
13 Li, B., Ma, S., Deng, R., Choo, K.K.R., and Yang, J., 2022. Federated anomaly detection on system logs for the internet of things: A customizable and communication-efficient approach. IEEE Transactions on Network and Service Management, 19(2), pp.1705-1716.https://doi.org/10.1109/TNSM.2022.3152620
14 Qiao, H., Tong, H., An, B., King, I., Aggarwal, C., and Pang, G., 2025. Deep graph anomaly detection: A survey and new perspectives. IEEE Transactions on Knowledge and Data Engineering. https://doi.org/10.1109/TKDE.2025.3581578
15 Siale, A.Y.D., Hassan, Q.M.Z., Kadekle, M.F.A.S., and Veena, B.S., 2025. Enhancing large-scale network security with a VGG-Net-based DCNN: a deep learning approach to anomaly detection. Journal of Robotics and Control (JRC), 6(3), pp.1316-1331..https://doi.org/10.18196/jrc.v6i3.25169
16 Tewari, S., 2022. Anomaly Detection in Large Scale Data Platforms with Machine Learning,7(2), p. 2456-4184. https://doi.org/10.13140/RG.2.2.11600.21762