ARTIFICIAL INTELLIGENCE ANALYSIS OF BIG DATA COLLECTED THROUGH IOT DEVICES

Qurbonov Behruz Amrulloyevich
Tashkent University of Information Technologies
named after Muhammad al-Khwarizmi 3rd year student
Faculty of Software Engineering
Recipient of the Muhammad al-Khwarizmi scholarship
Yondoshaliyev Alisher Elyorjon o'g'li
Tashkent University of Information Technologies
named after Muhammad al-Khwarizmi 2rd year student
Faculty of Software Engineering

Abstract: The Internet of Things (IoT) has revolutionized data collection by enabling billions of interconnected devices to generate vast amounts of data, often referred to as big data. These devices, ranging from smart sensors in industrial systems to wearable health monitors, produce high-volume, high-velocity, and high-variety data that require advanced analytical techniques for meaningful insights. Artificial Intelligence (AI), with its capabilities in machine learning (ML), deep learning (DL), and predictive analytics, is uniquely suited to process and analyze IoT-generated big data. This article explores the fundamentals of AI-driven analysis of big data from IoT devices, addressing methods, challenges, solutions, and mathematical formulations to quantify performance and efficiency.

Keywords: Analysis of IoT Big Data pandas, scikit-learn, TensorFlow, Anomaly Detection, Autoencoders , PySpark.

Data Ingestion and Preprocessing

IoT devices generate streaming data in formats like JSON, CSV, or time-series logs. Preprocessing is critical to handle noise, missing values, and heterogeneity.

• Streaming Data Ingestion: Tools like Apache Kafka or paho-mqtt in Python enable real-time data collection from IoT devices. The throughput of data ingestion is:

$$\Theta = \frac{D}{T}$$

where Θ is throughput, D is data volume, and T is processing time

Missing Value Imputation: Missing data is common in IoT due to connectivity issues. Imputation uses mean or time-series interpolation:

$$\hat{x}_t = \frac{1}{k} \sum_{i=t-k}^{t+k} x_i$$

where xˆ_t is the imputed value at time t, and x_i are neighboring values.

Normalization: IoT data often spans different scales. Min-max normalization is used:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where x ′ is the normalized value, and x_min, x_max are the minimum and maximum values.

Exploratory Data Analysis (EDA)

EDA identifies patterns in IoT data using visualization tools like matplotlib and seaborn. Correlation analysis quantifies relationships:

$$\rho = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

where ρ is the Pearson correlation coefficient, Cov(X, Y) is the covariance, and σX, σY are standard deviations.

Anomaly Detection Anomaly detection identifies unusual patterns in IoT data, such as equipment failures or cyber threats. Unsupervised learning algorithms like Isolation Forest or Autoencoders are effective.

• Isolation Forest: This algorithm isolates anomalies by randomly partitioning data. The anomaly score is based on path length:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

where s(x, n) is the anomaly score, E(h(x)) is the average path length, and c(n) is the average path length for n samples.

• Autoencoders: Deep learning models reconstruct normal data, with high reconstruction error indicating anomalies:

$$L = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2$$

where L is the reconstruction loss, xi is the input, and xˆi is the reconstructed output.

Predictive Modeling

Predictive models forecast future trends or events, such as equipment maintenance needs. Time-series models like ARIMA or Long Short-Term Memory (LSTM) networks are used.

ARIMA: Models time-series data:

$$\phi(B)(1 - B)^d y_t = \theta(B)\epsilon_t$$

where ϕ(B) and θ(B) are autoregressive and moving average polynomials, B is the backshift operator, d is the differencing order, y_t is the time series, and ϵ_t is white noise.

LSTM: Captures long-term dependencies in sequential data:

$$h_t = o_t \cdot \tanh(C_t)$$

where ht is the hidden state, ot is the output gate, and Ct is the cell state.

Data Heterogeneity

IoT data varies in format, scale, and quality, complicating analysis.

– Problem: Heterogeneous data reduces model accuracy, quantified by variance:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

where σ^2 is variance, xi are data points, and μ is the mean.

– Solution: Use data integration techniques, such as schema mapping in pandas, and feature engineering to standardize data.

IoT data often includes sensitive information, raising privacy and security concerns.

– Problem: Centralized data storage risks breaches, with privacy loss measured by differential privacy:

$$\epsilon = \ln\left(\frac{P(M|D)}{P(M|D')}\right)$$

where ϵ is the privacy budget, P(M|D) and P(M|D′) are model output probabilities for datasets D and D′ .

– Solution: Implement federated learning, where models are trained locally:

$$\Delta W = \sum_{i=1}^{k} \nabla L_i(W)$$

where ΔW is the aggregated model update, ∇Li(W) is the gradient from device i, and k is the number of devices. Use encryption for data transmission.

Key Algorithms for AI Analysis of IoT Data

Algorithm 1 Isolation Forest for Anomaly Detection

Input: Data points $X = \{x_1, \ldots, x_n\}$, number of trees T, sample size s

Output: Anomaly scores $s(x, n)$

for $t = 1$ to T **do**

Sample s points randomly from X

Build isolation tree by recursive random splits

Compute path length $h(x)$ for each $x \in X$

end for

Compute average path length: $E(h(x)) = \frac{1}{T}\sum_{t=1}^{T} h_t(x)$

Compute anomaly score: $s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$

Return: $s(x, n)$

Algorithm 2 Autoencoder for Anomaly Detection

Input: Data $X = \{x_1, \ldots, x_n\}$, epochs E, learning rate η

Output: Trained autoencoder, anomaly thresholds

Initialize encoder and decoder parameters θ

for each epoch $e = 1$ to E **do**

  for each $x_i \in X$ **do**

    Encode: $z_i \leftarrow f_{\theta_e}(x_i)$

    Decode: $\hat{x}_i \leftarrow g_{\theta_d}(z_i)$

    Compute loss: $L \leftarrow \|x_i - \hat{x}_i\|^2$

    Update θ using gradient descent: $\theta \leftarrow \theta - \eta \nabla L$

  end for

end for

Compute threshold: $T_h = \mu_L + k\sigma_L$

Return: θ, T_h

Algorithm 3 Stochastic Gradient Descent for Predictive Modeling

Input: Data $D = \{(x_i, y_i)\}$, learning rate η, epochs E

Output: Model parameters θ

Initialize θ randomly

for each epoch $e = 1$ to E **do**

  for each $(x_i, y_i) \in D$ **do**

    Compute prediction: $\hat{y}_i \leftarrow f_\theta(x_i)$

    Compute loss: $L \leftarrow L(\hat{y}_i, y_i)$

    Compute gradient: $\nabla L \leftarrow \frac{\partial L}{\partial \theta}$

    Update parameters: $\theta \leftarrow \theta - \eta \nabla L$

  end for

end for

Return: θ

AI-driven analysis of big data from IoT devices enables transformative applications in smart cities, healthcare, and manufacturing. Methods like anomaly detection, predictive modeling, and distributed computing, supported by Python libraries, handle the complexity of IoT data. Challenges such as data volume, heterogeneity, privacy, and computational complexity are mitigated through sampling, federated learning, and GPU acceleration. Mathematical formulations and algorithms, including Isolation Forest, Autoencoders, and SGD, provide a rigorous foundation for these solutions. By leveraging AI and IoT, organizations can unlock actionable insights, driving efficiency and innovation.

REFERENCES

1. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks* , 54(15), 2787–2805.
2. Zanella, A., et al. (2014). Smart cities: A literature review. *IEEE Internet of Things Journal* , 1(1), 20–31.
3. Al-Turjman, F. (2020). Artificial intelligence-enabled smart things in the internet of things era. *IEEE Access* , 8, 98516–98525.
4. Mao, Y., et al. (2017). Mobile edge computing: Survey and research outlook. *IEEE Communications Surveys & Tutorials* , 22(3), 1624–1657.
5. Chen, M., Yang, Z., Saad, W., & Yin, C. (2020). A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications* , 19(10), 6576–6590.
6. Xu, L. D., He, W., & Li, S. (2014). Internet of Things in industries: A survey. *IEEE Transactions on Engineering Management* , 61(4), 868–880.
7. Zhang, Y., et al. (2018). Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Mobile Computing* , 21(5), 1467–1480.
8. Ning, Z., et al. (2020). Intelligent resource scheduling in vehicular fog networks with reinforcement learning. *IEEE Transactions on Vehicular Technology* , 69(9), 9915–9926.
9. Bahl, P., Han, R., Li, F., & Satyanarayanan, M. (2002). Challenges in developing context-aware computing applications. *Wireless Communications, IEEE* , 9(5), 34–42.
10. IBM Research. (2021). *AI and IoT Convergence: Smarter Decisions at the Edge* . IBM White Paper.