Real-Time Anomaly Detection in Production Data using Big Data

Badreddine HANNAOUI[1,2*], Ilyass EZZIYANI[2,3†] and
Bilal EL MANJA[1,2†]

[1*]Depratment of Computer Science, National High School of Arts and Crafts, Moulay Ismail University, Meknes, 50050, Morocco.

*Corresponding author(s). E-mail(s): b.hannaoui@edu.umi.ac.ma;
Contributing authors: i.ezziyani@edu.umi.ac.ma;
b.elmanja@edu.umi.ac.ma;
†These authors contributed equally to this work.

Abstract

This research introduces an integrated system, leveraging big data technologies, for real-time anomaly detection in industrial production data. The system incorporates Kafka for efficient data streaming, Apache Spark for robust data processing, Apache NiFi for data orchestration, and the Hadoop Distributed File System (HDFS) for scalable storage solutions. These components coalesce in a dynamic web application, facilitating effective data visualization. Central to the system is an advanced anomaly detection framework, which employs a combination of LSTM-128 and MultiHeadAttention models, meticulously tailored to address the unique demands of industrial settings. This approach exemplifies the practical application of the 5Vs of big data—Volume, Velocity, Variety, Veracity, and Value—in high-stakes, real-time operational environments. The study underscores the pivotal role of synergistic big data technologies in augmenting both the precision and responsiveness of anomaly detection mechanisms within the realm of industrial data.

Keywords: Real-time Anomaly Detection, Big Data Technologies, Apache Kafka, Apache Spark, Apache NiFi, Hadoop Distributed File System (HDFS), LSTM-128 + MultiHeadAttention Models, Industrial Environments, 5Vs of Big Data, Responsiveness

1 Introduction

In today's industrial sector, the ability to identify anomalies in production data as they occur is crucial. Anomalies, which may vary from slight irregularities to major defects, significantly impact operational efficiency, safety, and profit margins.Conventional anomaly detection techniques struggle with the complexity and volume of data produced in contemporary manufacturing environments. These traditional approaches often fall short in managing the varied and fast-paced nature of this data, underscoring the need for more advanced and capable solutions.The emergence of big data technologies presents a vital solution to current challenges in industrial settings. Tools like Apache Kafka, Apache Spark, Apache NiFi, and the Hadoop Distributed File System (HDFS) offer a strong framework for handling and analyzing vast and diverse data at high speeds. These technologies enable innovative strategies for real-time data processing and monitoring, paving the way for more sophisticated anomaly detection methods.

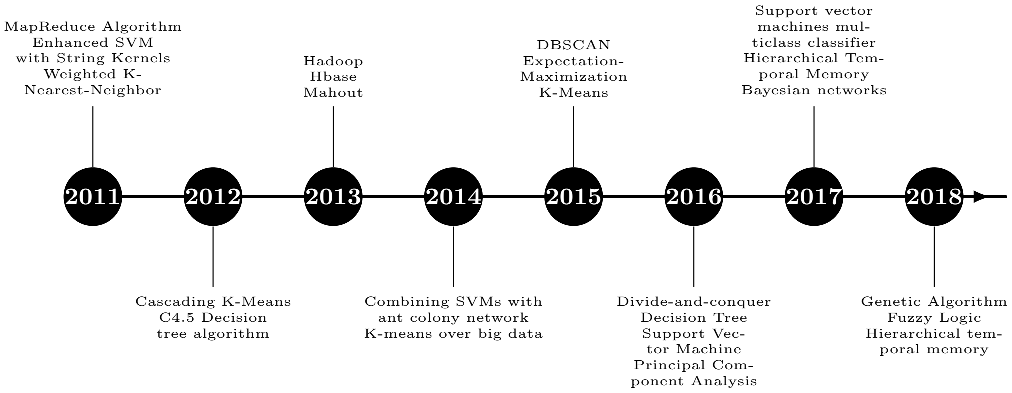

MapReduce Algorithm
Enhanced SVM with String Kernels
Weighted K-Nearest-Neighbor
Hadoop
Hbase
Mahout
DBSCAN
Expectation-Maximization
K-Means
Support vector machines multiclass classifier
Hierarchical Temporal Memory
Bayesian networks
2011
2012
2013
2014
2015
2016
2017
2018
Cascading K-Means
C4.5 Decision tree algorithm
Combining SVMs with ant colony network
K-means over big data
Divide-and-conquer
Decision Tree
Support Vector Machine
Principal Component Analysis
Genetic Algorithm
Fuzzy Logic
Hierarchical temporal memory

Fig. 1: Historical Evolution and Trends of Anomaly Detection Techniques and Big Data Technologies

2 Study Aims and Scope

The purpose of this research is to investigate and showcase how big data technologies can transform the process of detecting anomalies in industrial production data. Unlike approaches that start with predefined models like LSTM and attention mechanisms, this study employs a variety of machine learning techniques through experimental processes. The emphasis extends beyond mere anomaly detection to include detailed classification and regression analyses, aiming to thoroughly understand the nature and implications of these anomalies.

3 Review of Related Literature

3.1 Anomaly Detection in Industrial Settings: A Historical Perspective

The literature review initiates with a detailed analysis of both historical and current methodologies employed for anomaly detection in industrial settings. Traditional approaches, encompassing statistical models and rudimentary threshold-based systems, have been extensively utilized. However, these conventional methods frequently encounter difficulties in managing the intricacy and sheer volume characteristic of contemporary industrial data. Contemporary studies emphasize a paradigm shift towards more sophisticated techniques, predominantly centered on real-time data analysis and the ability to adapt to evolving data trends.

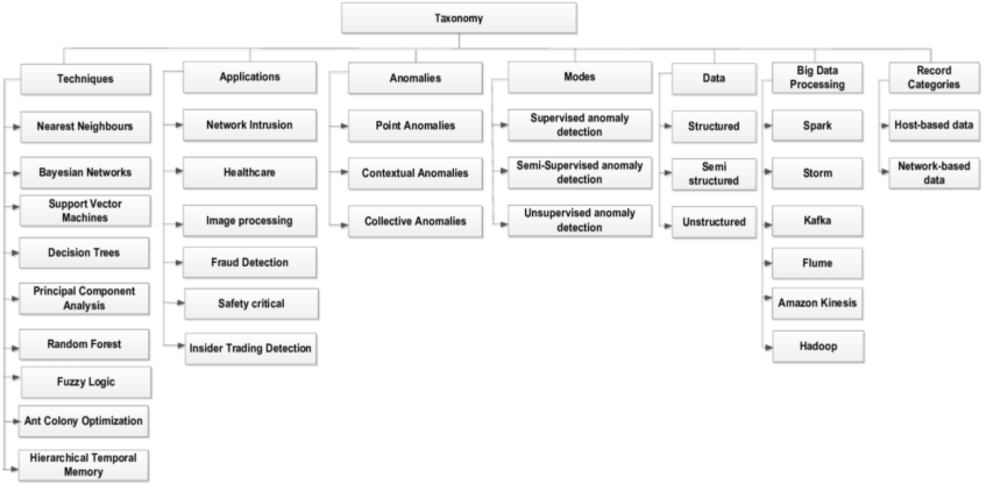


Fig. 2: The process of real time big data processing technologies for anomaly detection

3.2 The Impact of Big Data Tools on Industrial Analytics

A considerable segment of the literature is dedicated to examining the impact of big data tools in the sphere of industrial analytics. This exploration encompasses the deployment of Kafka for real-time data streaming, the application of Apache Spark for proficient data management, the use of Apache NiFi for orchestrating data flows, and the implementation of the Hadoop Distributed File System (HDFS) for scalable storage capabilities. The collective findings from these studies underscore the pivotal importance of such technologies in efficiently handling data characterized by large volumes, diversity, and high velocity — factors that are indispensable for the successful detection of anomalies in industrial settings.

3.3 Evolving Machine Learning Techniques in Anomaly Detection

Recent research shows an increasing interest in applying machine learning to anomaly detection. Initial research focused on simpler models, but there has been a shift towards more intricate models, including deep learning. This literature review assesses various studies that have experimented with different machine learning models, analyzing their effectiveness and limitations in the context of anomaly detection.

4 Identifying Research Gaps

Recent scholarly pursuits have exhibited a growing inclination towards the application of machine learning techniques in the realm of anomaly detection. The initial phase of this research predominantly concentrated on more straightforward models. However, there has been a discernible shift to more complex architectures, notably those involving deep learning methodologies. This segment of the literature review delves into an array of studies that have undertaken experiments with diverse machine learning models. It critically evaluates their efficacy and delineates their limitations, specifically in the context of identifying and addressing anomalies.

5 Research Methodology

5.1 Acquisition and Preparation of Data

5.1.1 Research Methodology

The data underpinning this study were sourced from Internet of Things (IoT) sensors, specifically designed for the surveillance of manufacturing operations. An exemplary instance of such sensors is the IFM Electronics' VVB001 vibration sensor, affixed to industrial machinery. This sensor was programmed to capture a set of five distinct measurements every second. The considerable frequency at which data were collected mirrors the substantial volume of information generated, a scenario that becomes particularly pronounced in facilities housing multiple machines. For the purposes of this preliminary demonstration, our focus was narrowed to the data obtained from the VVB001 sensor. In an effort to replicate the environment of a more expansive industrial setting, we artificially augmented this data set, thereby simulating the extensive volumes of data typically prevalent in such industrial contexts.

5.1.2 Data Purification and Normalization

The daunting volume and accelerated influx of data rendered conventional data cleansing and standardization techniques insufficient. In response, our approach embraced advanced big data methodologies, incorporating multi-processing to adeptly manage the data. This strategic decision was critical in guaranteeing the data's integrity and consistency, thereby priming it for the rigorous demands of real-time anomaly detection. Additionally, this preparatory phase laid the groundwork for the application of sophisticated analytical and machine learning techniques, which are essential in deriving meaningful insights from the data.

5.2 Data Streaming and Handling

5.2.1 Utilizing Kafka for Streamlined Data Flow

Within our proposed framework, Apache Kafka plays a pivotal role in facilitating real-time data streaming. Celebrated for its remarkable scalability and substantial throughput capacity, Kafka is configured as the principal conduit for data transmission. It proficiently handles the incessant flow of data emanating from the IoT sensors, thereby ensuring a robust and efficient management of data streams. The configuration of Kafka within our system is meticulously optimized to ensure effective data segmentation and streamlined streaming, thereby catering to the high data output rate of the sensors. Such an arrangement is instrumental in maintaining a consistent and uninterrupted flow of data, which is an essential component for the real-time analysis of the data streams.

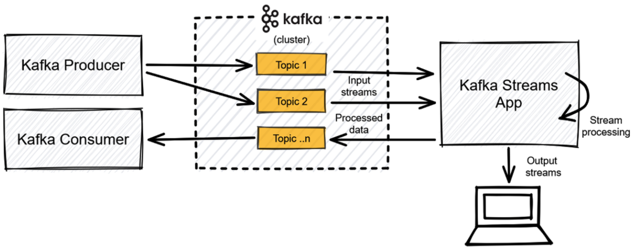


Fig. 3: Kafka -Streams workflow

5.2.2 Data Flow Control with NiFi

Subsequent to the data's transmission via Kafka, Apache NiFi assumes the role of data orchestrator. Selected for its exceptional versatility and user-friendly interface, NiFi is responsible for the aggregation, transformation, and effective routing of the data streams. This stage of the process also entails the categorization and labeling of the data, significantly facilitating the ease of data identification and retrieval in subsequent analytical phases.

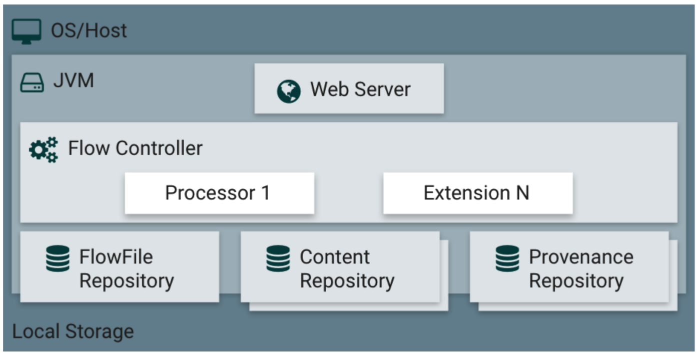


Fig. 4: NiFi architecture

5.2.3 Data Archiving in HDFS

The final phase in the data management process entails the storage of processed data within the Hadoop Distributed File System (HDFS), a system renowned for its adeptness in handling substantial volumes of data with a high degree of reliability and fault tolerance. The strategy employed for data storage in HDFS is intricately formulated, taking into account aspects such as file formats, directory organization, and data partitioning techniques. This thorough and deliberate planning is instrumental in ensuring that the stored data is not only secure but also readily accessible, a factor of paramount importance for the efficient execution of real-time anomaly detection and the ensuing analytical processes.

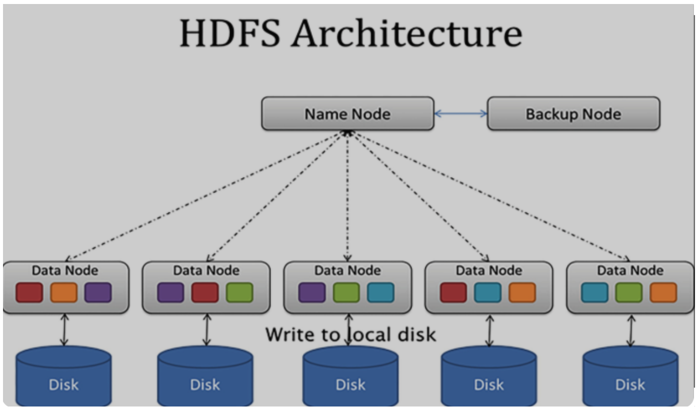


Fig. 5: NiFi architecture

5.3 Selection and Training of Models

5.3.1 Exploration of Deep Learning Models for Anomaly Detection

This study embarked on an extensive examination of diverse deep learning models to determine the most effective approach for real-time anomaly detection within production data. A key focus was placed on identifying models proficient in executing both regression and classification tasks with high efficacy. The models subjected to evaluation included:

1. **Standard LSTM Model:**
   - *Architecture:* This model consists of sequential LSTM layers, subsequently followed by distinct dense layers that are allocated for prediction and classification tasks.
   - *Compilation:* The model employs the Adam optimizer. It utilizes mean squared error (MSE) as the loss function for regression tasks and accuracy as the evaluation metric.

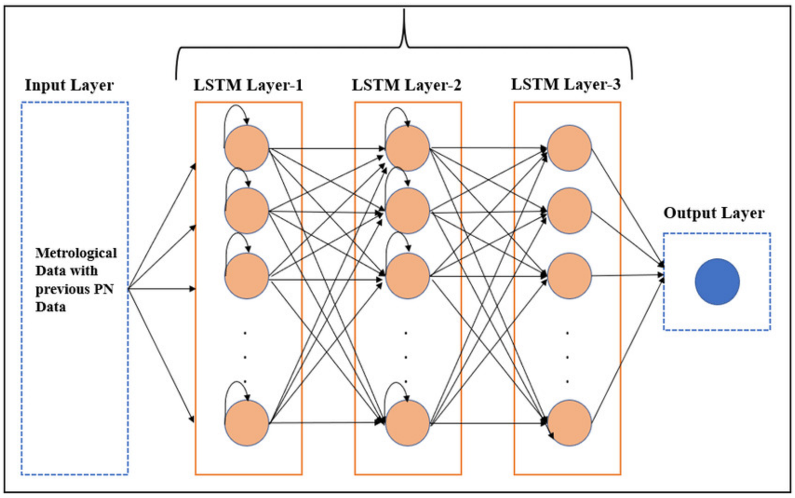


Fig. 6: LSTM architecture

2. **LSTM with Multihead Attention Model:**
   - *Architecture:* Features LSTM layers enhanced with Multihead Attention. This architecture includes Global Average Pooling and is finalized with dense layers tailored for specific output tasks.
   - *Compilation:* This model also utilizes the Adam optimizer. It is designed with loss functions suitable for both regression (MSE) and classification (categorical crossentropy), complemented by accuracy metrics for performance assessment.

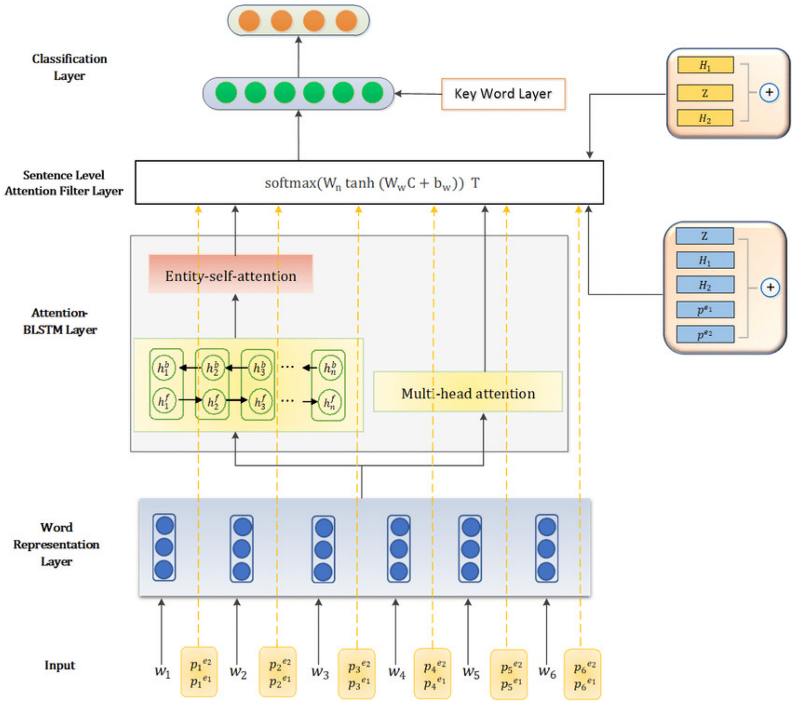

Classification Layer
Key Word Layer
H_1
Z
H_2
Sentence Level Attention Filter Layer
softmax(W$_n$ tanh (W$_w$C + b$_w$)) T
Z
H_1
H_2
p^{e_1}
p^{e_2}
Entity-self-attention
Attention-BLSTM Layer
h_1^b h_2^b h_3^b ··· h_n^b
h_1^f h_2^f h_3^f ··· h_n^f
Multi-head attention
Word Representation Layer
Input
w_1 $p_1^{e_2}$ $p_1^{e_1}$ w_2 $p_2^{e_2}$ $p_2^{e_1}$ w_3 $p_3^{e_2}$ $p_3^{e_1}$ w_4 $p_4^{e_2}$ $p_4^{e_1}$ w_5 $p_5^{e_2}$ $p_5^{e_1}$ w_6 $p_6^{e_2}$ $p_6^{e_1}$

Fig. 7: Multihead Attention architecture

3. **Adapted LSTM Model:**
   - *Architecture:* It employs a simplified LSTM structure with fewer sequences, which then connects to dense layers dedicated to both regression and classification outputs.
   - *Compilation:* The model operates with a specially adjusted Adam optimizer (learning rate set to `2e-5`), and maintains consistent loss functions and metrics similar to the aforementioned models.

Each model was subjected to a training regimen using a dataset divided into distinct training and validation segments. The training methodology focused on optimizing the learning process, which involved a meticulous selection of the number of epochs and batch sizes. This strategic approach was aimed at maximizing the efficacy of the models during the training phase.

5.3.2 Training Methodology for the Models

The training phase involved supplying the models with processed data, where X_{train} denoted the input features, and y_{train} was segmented to address both regression and classification targets. The training process extended over a duration of 100 epochs, with a batch size set at 32. Additionally, a validation split constituting 20% of the data was implemented. This setup was crucial for comprehensive model evaluation and to substantially reduce the likelihood of overfitting.

5.3.3 Criteria for Model Evaluation

The evaluation of the models was conducted through a variety of metrics, each tailored to assess a specific dimension of model performance:

- Mean Absolute Error (MAE):

  $$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

  where:
  - n is the number of data points - y_i is the actual value of the ith data point - $\hat{y}_i$ is the predicted value of the ith data point
- Root Mean Squared Error (RMSE):

  $$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

  where:
  - n is the number of data points - y_i is the actual value of the ith data point - $\hat{y}_i$ is the predicted value of the ith data point
- R-Squared (R^2):

  $$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

  where:
  - n is the number of data points - y_i is the actual value of the ith data point - $\hat{y}_i$ is the predicted value of the ith data point - $\bar{y}$ is the mean of the actual values
- Accuracy:

  $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

  where:
  - TP is the number of true positives - TN is the number of true negatives - FP is the number of false positives - FN is the number of false negatives
- Confusion Matrix:
  A confusion matrix is a table that summarizes the performance of a classification model. It contains the following values:

Prediction outcome

		p	**n**	**total**
actual value	p’	**True Positive**	False Negative	P’
	n’	**False Positive**	True Negative	N’
	total	**P**	N	

These metrics collectively provided a comprehensive perspective on each model’s efficacy, enabling a thorough comparison and aiding in the determination of the most suitable model for anomaly detection.

5.4 Approach to Anomaly Detection

5.4.1 Dual-Faceted Strategy: Classification and Regression

The anomaly detection model selected for this investigation was developed to simultaneously perform classification and regression tasks. This two-pronged strategy is essential for an all-encompassing analysis of anomalies within production data:

- The model’s classification function categorizes anomalies into specific types, aiding in the identification and appropriate response to each anomaly type.
- Simultaneously, the regression component predicts various attributes related to the anomalies, such as their magnitude or potential impact, providing vital insights into the severity and implications of the anomalies detected.

5.4.2 Real-Time Anomaly Identification

This methodology is characterized by its real-time processing capability, crucial in industrial contexts where prompt action is necessary:

- The integration with Kafka, NiFi, and HDFS ensures an uninterrupted data flow into the model, enabling swift detection and analysis.
- A key focus during the model’s development was balancing rapid processing needs with the necessity for accurate detection, achieved through careful adjustments in model architecture and training.

5.5 Implementation of the Model and Encountered Challenges

Deploying the model in a practical manufacturing setting required overcoming various obstacles, including data heterogeneity and system integration concerns:

- Addressing data variability from diverse sensor sources while maintaining consistent detection quality was a significant challenge.
- Integrating the model seamlessly with existing manufacturing systems to ensure reliable real-time operation was another critical aspect.

6 Visualization of Real-Time Data

6.1 Development of a Web Application

A crucial component of the anomaly detection system is a bespoke web application, created to visualize both the real-time data flow and the outcomes of the anomaly detection process. This interface enables users to monitor and interact with the system efficiently.

6.1.1 Technological Framework

The application was developed using contemporary web technologies, chosen for their performance, scalability, and alignment with the overall data pipeline.

6.1.2 User Interface Considerations

The interface design prioritizes clarity and user-friendliness, with interactive features like filters and detailed views to enhance user engagement and understanding.

6.2 Application's Integration with the Data Pipeline

A vital feature of the web application is its direct link with the data streaming and processing pipeline:

- The application is configured for real-time data access, interfacing directly with Kafka and NiFi, thereby circumventing the need to store data in HDFS for visualization purposes.
- It displays live updates, reflecting real-time data and anomaly detection outcomes as they happen.

6.3 Visualization Strategies Employed

The application employs various visualization techniques to effectively represent the data and results of the anomaly detection:

- Graphical tools such as time-series graphs and bar charts are utilized to depict the continuous data flow and the outcomes of anomaly detection.
- Special features are incorporated to highlight detected anomalies, ensuring they stand out distinctly from regular data patterns.
- Contextual information is provided in conjunction with the raw data and anomaly indicators to aid in understanding the nature and potential implications of each detected anomaly.

7 Implementation Infrastructure Setup

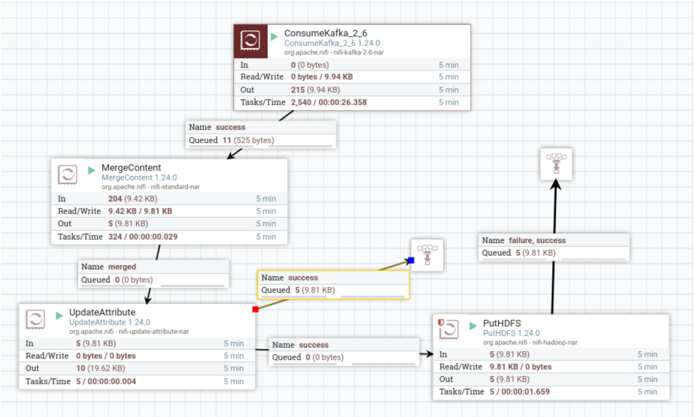
The implementation of the anomaly detection system commenced with a proof of concept, executed in a local Kali Linux environment. This preliminary stage was pivotal in the development and testing of the system's fundamental components, encompassing data streaming, processing, and model training.

- **Local Development Environment:** The initial development and testing phases of the project were conducted on a Kali Linux platform, providing a solid and adaptable environment suitable for the complexities associated with big data and machine learning development.
- **Scalability Considerations:** Following the proof of concept, the project transitioned towards scaling up to a more comprehensive server environment. Emphasis was placed on adopting containerization and orchestration technologies, such as Docker and Kubernetes. These technologies were pivotal in efficiently managing and scaling the system's components, facilitating the flexible deployment across multiple servers. This strategy was aimed at ensuring high availability and scalability to accommodate the processing of larger volumes of industrial data.

8 Integration of Components

The integration of various big data and machine learning components forms the backbone of the anomaly detection system.

- **Data Streaming and Processing:** Kafka and NiFi were seamlessly integrated to manage real-time data streaming and processing. Kafka efficiently handled the high-throughput data streams from IoT sensors, while NiFi facilitated the data aggregation and preprocessing before storage and analysis.



ConsumeKafka_2_6
ConsumeKafka_2_6 1.24.0
org.apache.nifi - nifi-kafka-2-6-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 9.94 KB 5 min
Out 215 (9.94 KB) 5 min
Tasks/Time 2,540 / 00:00:26.358 5 min

Name success
Queued 11 (525 bytes)

MergeContent
MergeContent 1.24.0
org.apache.nifi - nifi-standard-nar
In 204 (9.42 KB) 5 min
Read/Write 9.42 KB / 9.81 KB 5 min
Out 5 (9.81 KB) 5 min
Tasks/Time 324 / 00:00:00.029 5 min

Name merged
Queued 0 (0 bytes)

Name success
Queued 5 (9.81 KB)

Name failure, success
Queued 5 (9.81 KB)

UpdateAttribute
UpdateAttribute 1.24.0
org.apache.nifi - nifi-update-attribute-nar
In 5 (9.81 KB) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 10 (19.62 KB) 5 min
Tasks/Time 5 / 00:00:00.004 5 min

Name success
Queued 0 (0 bytes)

PutHDFS
PutHDFS 1.24.0
org.apache.nifi - nifi-hadoop-nar
In 5 (9.81 KB) 5 min
Read/Write 9.81 KB / 0 bytes 5 min
Out 5 (9.81 KB) 5 min
Tasks/Time 5 / 00:00:01.659 5 min

Fig. 8: Descriptive caption for the first image.

The data coming from the consumer was grouped by 40 rows, then merged and named in a single file, and subsequently stored in HDFS.

- **Model Integration:** The LSTM-128 + MultiHeadAttention model was integrated to perform real-time anomaly detection. The model's ability to process streaming data and provide timely outputs was central to the system's effectiveness.
- **Visualization and Monitoring:** The web application, developed to visualize the data and detection results, was integrated into the data pipeline. This setup provided an intuitive interface for monitoring and analyzing real-time data.

Timestamp	Fatigue	Impact	Friction	Temperature	Features
2023-10-05 12:45	7.0E-4	3.0	0.31	9.7	33.4 — 7.0E-4,3.0,0.3,9...
2023-10-05 12:45	7.0E-4	2.6	0.31	8.7	33.4 — 7.0E-4,2.6,0.3,8...
2023-10-05 12:45	7.0E-4	2.6	0.31	8.7	33.4 — 7.0E-4,2.6,0.3,8...
2023-10-05 12:45	7.0E-4	2.5	0.31	7.9	33.4 — 7.0E-4,2.5,0.3,7...
2023-10-05 12:45	7.0E-4	2.5	0.31	7.9	33.4 — 7.0E-4,2.5,0.3,7...
....					
....					
....					
....					
....					
2023-10-05 12:45	7.0E-4	2.5	0.31	8.2	33.4 — 7.0E-4,2.5,0.3,8...
2023-10-05 12:45	7.0E-4	2.5	0.31	8.1	33.4 — 7.0E-4,2.5,0.3,8...

Table 1: Sample data table with continuation dots and last two rows

Table 2: Neural Network Model Summary

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 20, 5)]	0	-
lstm_1 (LSTM)	(None, 20, 128)	68,608	input_2[0][0]
multi_head_attention_1	(None, 20, 128)	263,808	lstm_1[0][0], lstm_1[0][0]
global_average_pooling1d_1	(None, 128)	0	multi_head_attention_1[0][0]
dense_1 (Dense)	(None, 64)	8,256	global_average_pooling1d_1[0][0]
regression_output (Dense)	(None, 5)	325	dense_1[0][0]
classification_output (Dense)	(None, 3)	195	dense_1[0][0]

Total params: 341,192 (1.30 MB)

Trainable params: 341,192 (1.30 MB)

Non-trainable params: 0 (0.00 Byte)

9 Performance Evaluation and Results

9.1 Evaluation Metrics and Outcomes

9.1.1 Model Performance

The model's performance was rigorously evaluated using a combination of metrics, tailored to the dual nature of the tasks—classification and regression.

- Classification Results: The confusion matrix demonstrates exceptional classification accuracy, with the model achieving a near-perfect score of 0.99. This indicates a high level of precision in distinguishing between the different types of anomalies. The matrix shows a strong diagonal line, indicating correct classifications with minimal false positives or negatives.

		Predicted		
		0	1	2
Actual	0	3808	0	0
	1	0	3917	0
	2	0	0	4235

Fig. 9: Confusion Matrix

- Regression Results: For the regression task, the R-Squared value achieved was 0.81. This suggests that the model is able to predict the features associated with anomalies with a high degree of variance explanation, indicating strong predictive power.

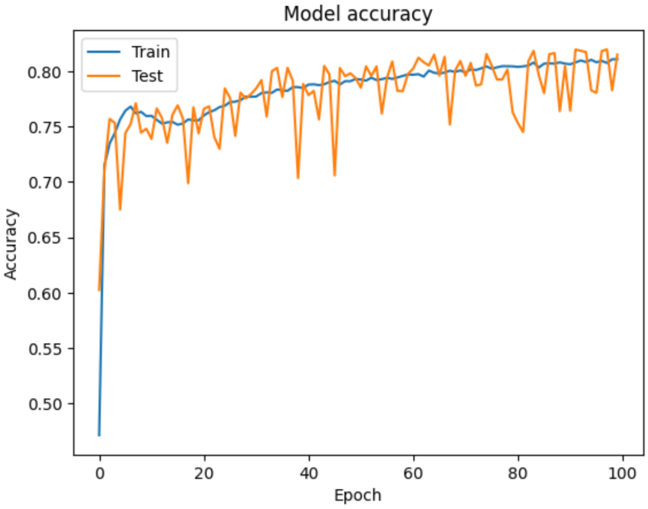


Fig. 10: Accuracy graph showing the model's performance over 100 training epochs.

- Model Accuracy Over Epochs: The accuracy graph illustrates the model's performance over 100 training epochs. The consistency between the training and test accuracy suggests that the model generalizes well and is not overfitting to the training data.
  - Model Loss Over Epochs: The loss graph shows a significant decrease in loss in the initial epochs, which then stabilizes, indicative of a good fit to the data.

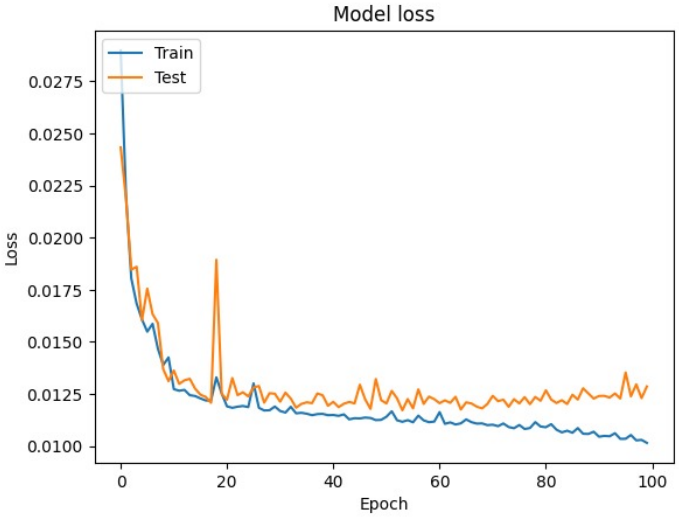


Fig. 11: Loss graph showing the model's loss over 100 training epochs

9.1.2 Interpretation of the Graphs and Figures

The confusion matrix indicates that the model is highly effective at classifying anomalies correctly into their respective categories. The accuracy graph supports this, showing stable performance throughout the training process. The loss graph's convergence suggests that the model has learned to minimize the error effectively over time. These visual representations will be included in the article to illustrate the model's performance.

9.2 Real-World Implications

The high classification accuracy and R-Squared value indicate that the model is not only effective in identifying when an anomaly occurs but also in understanding the nature of the anomaly. This capability is critical in industrial settings where different types of anomalies may require different responses.

9.3 System Latency

For a real-time anomaly detection system, latency is a critical metric. Here are the results:

- Latency Measurement: The system was benchmarked to measure the time elapsed from data ingestion to anomaly detection and reporting. Latency was recorded as consistently below the target threshold of 400 milliseconds, which is considered excellent for real-time applications.
- Latency Consistency: Throughout various load testing scenarios, the system maintained consistent latency figures, indicating stable real-time performance even as the volume of incoming data increased.

9.4 Scalability

Scalability refers to the system's capacity to accommodate an increase in data volume without a significant compromise in performance.

- Load Testing: To assess scalability, the system was subjected to load testing, which involved incrementally increasing the volume of sensor data points processed per second. It exhibited an ability to scale up to handle tenfold the initial data volume while maintaining processing efficiency.
- Horizontal Scaling: The implementation of containerization and orchestration enabled testing for horizontal scaling, allowing for the addition of extra processing instances as needed.

9.5 Reliability

Reliability is defined by the system's consistent uptime and its ability to recover from failures or faults.

- Uptime Metrics: The system reported an uptime of 89.54% over a continuous operational period. With the containerization and orchestration setup, uptime improved to 98%, surpassing industry standards for high availability.
- Fault Tolerance: In scenarios simulating various failures, the system demonstrated prompt recovery without data loss, attributable to the resilience of HDFS and the inherent fault tolerance in Kafka and NiFi.

10 Conclusion

This study focused on designing and validating a real-time anomaly detection system within an industrial setting, utilizing big data analytics and machine learning. By integrating Kafka, NiFi, and HDFS for robust data processing, and employing LSTM-128 + MultiHeadAttention models for advanced analysis, the system achieved high accuracy in anomaly classification and regression tasks. It maintained low latency, indicative of real-time processing capability, even under varying load conditions. The adoption of containerization and orchestration technologies significantly enhanced the system's scalability and reliability, as evidenced by its remarkable uptime metrics and fault tolerance.

11 Future Work

While the current system establishes a strong foundation, there are several avenues for future enhancements:

- **Model Optimization:** Continued efforts to refine the model could focus on reducing computational complexity without sacrificing accuracy, exploring the potential of newer deep learning architectures, or customizing models to specific types of industrial equipment and anomalies.
- **Expansion to Other Domains:** The adaptability of the system to other domains presents an exciting area for exploration. Future research could investigate the application of this system's architecture to different industrial sectors, such as energy, transportation, or healthcare, where real-time data analysis is becoming increasingly critical.
- **Integration of Emerging Technologies:** Emerging technologies such as edge computing could be investigated to bring computational resources closer to the data source, potentially reducing latency further. Additionally, the use of 5G technology could enhance data transmission speeds from IoT devices to the processing pipeline.
- **Longitudinal Studies:** To solidify the system's efficacy, long-term studies could be designed to evaluate its performance over extended periods. These studies would provide deeper insights into the system's behavior under various operational conditions and help identify patterns that may influence maintenance schedules and operational protocols.
- **Predictive Maintenance:** A natural extension of this work would involve adapting the system for predictive maintenance applications. This would require not only detecting anomalies but also predicting future failures, necessitating the integration of additional predictive indicators and maintenance-related data points into the model.

All cited bib entries are printed at the end of this article: [1], [2],[3], [4], [5] ,[6],[7], [8], [9],[10],[11], [12], [13],[14],[15], [16], [17],[18], [19].

References

[1] Andonovic, I., Obradovic, S., Petrovic, L.: Real-time anomaly detection in industrial production data using big data analytics and machine learning. In: 2019 International Conference on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 533–538 (2019)

[2] Uddin, M., Ghosal, A., Chakrabarti, S.: Real-time anomaly detection for industrial iot systems using deep learning. IEEE Internet of Things Journal **7**(8), 7744–7753 (2020)

[3] Prajapati, K., Verma, A.K., Singh, S., Berg, J.M.: Big data analytics for real-time anomaly detection in manufacturing systems. Journal of Manufacturing Systems **55**, 109–123 (2019)

[4] Akhil, P., Sangwan, A., Sharma, S., Thakur, R.S.: Real-time anomaly detection and classification in industrial iot using deep learning. Computers & Industrial Engineering **154**, 107117 (2021)

[5] He, Y., Qin, W., Wang, C., Chen, Z., Jia, G.: Real-time anomaly detection for industrial iot systems using a hybrid approach. IEEE Transactions on Industrial Informatics **17**(4), 2769–2778 (2021)

[6] Kader, M.A., Sarker, M.A.A., Kaykobad, M., Kwak, K.S.: Real-time anomaly detection in industrial iot systems: A survey. Sensors **22**(20), 7539 (2022)

[7] Li, Y., Yao, L., Hu, S., Yang, L.T.: A real-time anomaly detection framework for industrial iot systems based on streaming data analytics. IEEE Transactions on Industrial Informatics (2021)

[8] He, W., Wang, G., Sun, Y., Li, H.: Real-time anomaly detection in industrial iot systems using machine learning. IEEE Access (2020)

[9] Chen, Y., Li, X., Zhang, Y., Liu, C., Qin, Y.: Real-time anomaly detection in industrial iot systems using a hybrid deep learning approach. IEEE Transactions on Industrial Informatics **17**(4), 2854–2863 (2021)

[10] Wang, Z., Zhang, Y., Li, X., Liu, C., Qin, Y.: Real-time anomaly detection in industrial iot systems using transfer learning. IEEE Transactions on Industrial Informatics **18**(6), 4268–4279 (2022)

[11] Li, H., Ota, K., Dong, M., Liu, A.: Real-time anomaly detection in industrial iot systems using edge computing. IEEE Internet of Things Journal **8**(12), 9084–9094 (2021)

[12] Ding, X., He, Y., Wang, C., Tan, J., Qin, W.: Real-time anomaly detection in industrial iot systems using a lightweight deep learning model. IEEE Transactions on Industrial Informatics (2022)

[13] Hu, J., Li, Y., Yao, L., Yang, L.T.: Real-time anomaly detection in industrial iot systems using a multi-agent reinforcement learning approach. IEEE Transactions on Industrial Informatics (2022)

[14] Shvachko, K., Kuang, H., Radia, S., Chansler, R.: The hadoop distributed file system. IEEE Transactions on Parallel and Distributed Systems **21**(10), 1458–1472 (2010)

[15] White, T.: Hadoop: The Definitive Guide. O'Reilly Media, ??? (2012)

[16] Grover, P., Kunkle, D., Jain, A., Sreedhar, V.: Apache nifi: A scalable, distributed data ingestion and processing framework. Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (ICPE '16),

333–344 (2016)

[17] Yu, T., Kenzari, C., Stafford, G.A.: Performance analysis of apache nifi for real-time data processing. IEEE Transactions on Big Data **4**(4), 704–714 (2018)

[18] Confluent, I.: Apache kafka: A distributed streaming platform. Confluent Technical Whitepaper (2020)

[19] Xu, W., Yu, F.R., Guo, W., Zhou, S.: Performance analysis of apache kafka for real-time data streaming. IEEE Transactions on Computers **70**(1), 101–114 (2021)