International Journal of Engineering Sciences & Research Technology

(A Peer Reviewed Online Journal)
Impact Factor: 5.164

IJESRT

RESEARCHERID
THOMSON REUTERS

[Srinivas *al.*, 10(11): November, 2021]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

EMPOWERING MAINFRAMES WITH AI/ML CAPABILITIES: REIMAGINING WHAT'S POSSIBLE

Srinivas Adilapuram
Lead Application Developer, ADP Inc, USA

DOI: 10.5281/zenodo.14619498

ABSTRACT

Mainframes are efficient systems with immense storage and processing power. They support critical applications in industries like banking, healthcare, and logistics. These systems excel in batch processing, transaction handling, and managing large datasets. However, they lack modern analytics capabilities. Advanced tasks such as real-time fraud detection and predictive maintenance expose inefficiencies in legacy systems. The gap widens with rising data volumes and evolving operational demands. Integrating Artificial Intelligence (AI) and Machine Learning (ML) offers a transformative solution. AI/ML models empower mainframes to perform advanced analytics, optimize processes, and enhance customer experiences. This article looks at how AI/ML integration can modernize mainframes.

KEYWORDS: Mainframe systems, AI/ML integration, fraud detection, predictive maintenance, data analytics, operational optimization, hybrid solutions, legacy modernization, TensorFlow.

1. INTRODUCTION

Mainframes have long been the backbone of enterprise operations. They manage critical workloads and process vast amounts of data. Industries such as finance, government, and healthcare depend on them. These systems handle batch processing, transaction logs, and secure data storage. However, the digital age demands more. Businesses need insights in real time. They require advanced data analytics, predictive models, and faster processing.

Artificial Intelligence (AI) and Machine Learning (ML) have changed the game. These technologies introduce automation, enhanced security, and intelligent systems. With AI/ML, businesses analyze data faster and detect anomalies in real-time. Tasks like risk management, fraud detection, and process optimization become efficient. Mainframes, despite their power, fall short in adopting these capabilities.

The limitation lies in their design. They rely on legacy protocols and frameworks. These worked well years ago but struggle with modern requirements. The result is inefficiency in areas like fraud detection and operational optimization. Additionally, they lack real-time data analytics.

Integrating AI/ML capabilities bridges this gap. AI/ML tools use advanced models for tasks like fraud detection and predictive maintenance. Frameworks such as TensorFlow and PyTorch train these models. APIs allow seamless integration with mainframes, ensuring smooth data flow. Hybrid solutions maintain core functionalities while introducing modern capabilities. This article discusses these methods, challenges, and solutions in detail.

2. LITERATURE REVIEW

Mainframe systems have been pivotal in critical enterprise applications for decades, but their legacy architectures now face growing challenges in adapting to modern data analytics requirements. The operationalization of AI has been extensively discussed by Hechler et al. [2], who emphasized the importance of integrating AI/ML capabilities into enterprise systems to enhance decision-making and operational efficiency.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Shestak et al. [1] highlighted data structuring as a vital component in improving processing efficiencies, which aligns with the AI/ML need for high-quality training data. Raghavan and El Gayar [7] showed the utility of machine learning algorithms, particularly in fraud detection, showcasing the superiority of ensemble models over traditional rule-based systems. Their work shows the effectiveness of applying real-time ML-driven fraud detection in mainframe environments; especially in supercomputer environments [3].

Predictive maintenance has gained traction as an essential application of ML, as highlighted by McKay et al. [4], who used advanced anomaly detection techniques for system reliability. Autoencoders, as discussed by Ziegler et al. [9], offer a unique t solution for taming hardware complexity, which can be adapted for identifying failure patterns in mainframes.

Xue et al. [5] discussed the security threats posed by static risk models, advocating for adaptive ML approaches to strengthen cybersecurity. These insights are critical in designing AI/ML-enabled mainframes to detect and respond to cyber threats in real time.

Apruzzese et al. [6] discussed the application of deep learning in cybersecurity, further emphasizing its relevance for mainframes dealing with sensitive transactional data. Their work provides a foundation for integrating LSTM and autoencoder models for fraud detection and predictive analytics.

3. PROBLEM STATEMENT: LEGACY MAINFRAMES FACE OBSOLESCENCE IN MODERN ANALYTICS-DRIVEN ECOSYSTEMS

Mainframes often have vast resources and data storage capabilities, being used for a broad range of activities supporting organizations. However, many systems, especially those in government or manufacturing settings, adopt an "if it isn't broken, don't fix it" policy toward modernization. It is technology's nature that if not modernized, it becomes obsolete. The same is applicable for mainframes as well. While they used to be efficient using the same resources and protocols a few years ago, modern capabilities and requirements mean that compared to modern requirements, there may be inefficiencies in the processes.

3.1 Limited Real-Time Data Processing

Mainframes rely on batch processing architectures designed for sequential workloads. This framework lacks the agility required for real-time data analysis. As a result, latency issues arise when processing high-velocity data streams, such as those generated by IoT sensors or financial transactions.

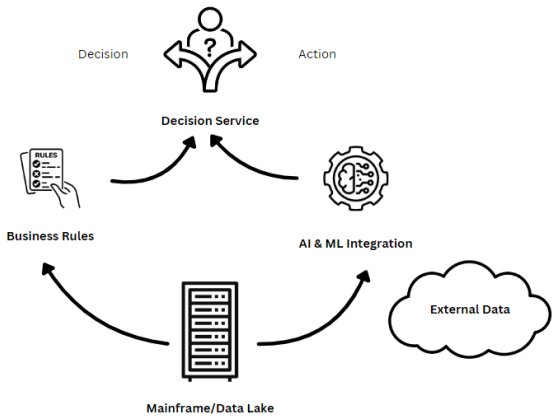

Figure 1: Real-time Decision making based on mainframe data

Without real-time capabilities, fraud detection systems fail to identify anomalies during live transactions, increasing operational risk. Similarly, predictive analytics requiring instantaneous insights are rendered ineffective. This limitation undermines competitive positioning in data-driven markets where real-time decision-making is critical. [8]

3.2 Absence of Predictive Maintenance Mechanisms

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Legacy mainframes operate on reactive maintenance strategies. These systems detect failures post-occurrence, leading to extended downtimes. Reactive methodologies depend heavily on predefined thresholds and do not adapt to evolving system dynamics. Predictive maintenance relies on ML-driven anomaly detection and pattern recognition to preempt potential disruptions. Without AI/ML integration, mainframes fail to detect non-linear trends in hardware wear-and-tear or usage patterns, leading to suboptimal resource allocation and avoidable system failures.

3.3 Inefficient Fraud Detection Systems

Traditional fraud detection in mainframes depends on deterministic rule-based algorithms. These static models require frequent manual updates to account for emerging threat vectors. Rule-based systems cannot analyze multifactorial data relationships or adapt to novel fraud schemes in real time. This may introduce several risks to file transfers, including Man-In-The-Middle attacks.

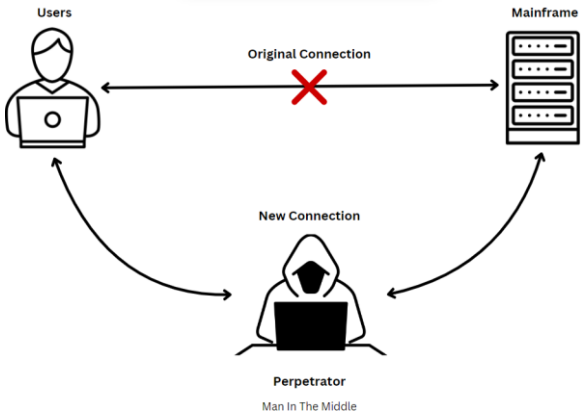

Figure 2: Illustration of a Man-In-The-Middle attack.

To prevent this, there must be a system that can detect latent patterns within multi-dimensional datasets. Without such capabilities, mainframes struggle to counter sophisticated cyberattacks, leaving sensitive transactional data vulnerable to breaches.

3.4 Lack of Scalability for Modern Workloads

Mainframes operate within fixed resource constraints defined by their monolithic architectures. Modern applications generate exponentially growing data volumes and require elastic scalability to manage dynamic workloads. AI/ML workloads, such as deep learning model training, demand high computational resources and memory bandwidth. Mainframes cannot efficiently scale to accommodate such demands, creating bottlenecks. This restricts their ability to support advanced analytics applications, such as customer segmentation or churn prediction, that require high-frequency model retraining.

3.5 Ineffective Customer Segmentation for Personalization

Customer data processed by mainframes is often siloed and lacks multi-channel integration. This prevents the creation of unified customer profiles essential for effective segmentation. Legacy analytical tools cannot incorporate unstructured data, such as social media sentiment or real-time behavioral metrics, into segmentation models. Machine learning techniques like clustering algorithms (e.g., K-Means or DBSCAN) enable granular customer classification. Without these, mainframes deliver generalized insights, reducing their utility in delivering personalized user experiences critical to customer retention.

3.6 Security Vulnerabilities Due to Static Risk Models

Cybersecurity risk assessment on mainframes often uses static, threshold-based models. These models cannot adapt to evolving threat environment or detect zero-day vulnerabilities. Advanced security frameworks utilize ML algorithms for anomaly detection, risk scoring, and intrusion prediction. For instance, unsupervised learning models, such as Autoencoders, identify hidden attack patterns that evade traditional rule-based mechanisms. Without such defenses, mainframes expose organizations to substantial compliance risks and reputational damage.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

4. SOLUTION: AI/ML MODELS FOR MAINFRAME OPTIMIZATION

Integrating AI and ML technologies addresses critical vulnerabilities in mainframes; especially those operating with legacy technologies or settings. The goal is to allow for dynamic threat detection, adaptive security policies, and scalable resource optimization for modern workloads.

4.1 Fraud Detection Using Real-Time Transaction Analysis

Fraud detection in mainframes involves monitoring live transactions for anomalies. AI/ML can revolutionize this by employing real-time supervised learning models. Using libraries like TensorFlow, an LSTM (Long Short-Term Memory) neural network can analyze sequential transaction data to predict fraudulent behavior.

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Define the model
model = Sequential([
    LSTM(128, input_shape=(30, 10), return_sequences=True),
    LSTM(64),
    Dense(1, activation='sigmoid')  # Output layer for binary
classification
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

# Dummy training data (transaction sequences)
# X_train: 3D array (samples, time steps, features)
# y_train: Labels (1 for fraud, 0 for non-fraud)
X_train = ...  # Replace with transaction data
y_train = ...

# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32)
```

Figure 3: Setting up an LSTM network

This code sets up an LSTM network. The input layer accepts time-series data (e.g., a sequence of 30 transactions with 10 features each). The output layer uses a sigmoid function to predict whether a transaction is fraudulent. Here;

1. LSTMs excel at handling sequential data, crucial for transaction patterns.
2. The binary_crossentropy loss function optimizes fraud/no-fraud classification.
3. Once trained, the model is deployed via APIs connected to mainframe systems.

The implementation allows for dynamic fraud detection, reducing reliance on static rules. Challenges include ensuring latency is minimal and securing the data pipeline.

4.2 Predictive Maintenance to Minimize Downtime

Predictive maintenance uses unsupervised learning models to detect anomalies in system behavior before failure occurs. Autoencoders, which compress and reconstruct input data, are ideal for anomaly detection in mainframe logs.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense

# Define the autoencoder model
input_layer = Input(shape=(20,))  # 20 features from
mainframe logs
encoded = Dense(10, activation='relu')(input_layer)
decoded = Dense(20, activation='sigmoid')(encoded)

autoencoder = Model(input_layer, decoded)
encoder = Model(input_layer, encoded)  # Encoder for
compressed representation

# Compile the model
autoencoder.compile(optimizer='adam', loss='mse')

# Train the autoencoder
X_logs = ...  # Mainframe log data
autoencoder.fit(X_logs, X_logs, epochs=20, batch_size=64)
```

Figure 4: Setting up an AI/ML enabled autoencoder

The autoencoder set up above learns to compress log data into a reduced representation with the help of an ML model. Of course, it will need to be set up with the relevant database as well to store what it learns.
Reconstruction errors indicate anomalies in patterns, signaling potential issues. Engineers analyze high-error instances to preempt system failures.

Predictive maintenance via this model minimizes unscheduled downtimes. However, integrating real-time log processing into legacy mainframes may require significant resource investment.

4.3 Customer Segmentation for Personalized Experiences

Clustering algorithms like K-Means can create customer segments based on purchasing behavior, demographics, and engagement.

```
from sklearn.cluster import KMeans
import pandas as pd

# Sample customer data
data = pd.DataFrame({
    'age': [25, 45, 34, 65, 23],
    'spending_score': [80, 40, 70, 20, 90]
})

# Apply K-Means
kmeans = KMeans(n_clusters=3, random_state=0).fit(data)

# Assign cluster labels
data['cluster'] = kmeans.labels_

print(data)
```

Figure 5: K-Means clustering model for mainframe

Here, the fit method clusters customers into groups based on input features like age and spending. Labels assigned to clusters provide actionable insights for personalization. Segmentation data, on the other hand, integrates into mainframe-driven marketing campaigns.

AI-driven segmentation enhances personalization but necessitates continuous updates to clusters based on new data.

4.4 APIs for Seamless AI/ML Integration

APIs act as the bridge between AI/ML models and mainframe systems. A RESTful API example using Flask:

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

```
from flask import Flask, request, jsonify
import pickle

app = Flask(__name__)

# Load a pre-trained model
model = pickle.load(open('fraud_model.pkl', 'rb'))

@app.route('/predict', methods=['POST'])
def predict():
    data = request.json
    prediction = model.predict([data['transaction_features']])
    return jsonify({'fraudulent': bool(prediction[0])})

if __name__ == '__main__':
    app.run(port=5000)
```

Figure 6: Setting up a RESTful API via Flask

Here, the API accepts transaction data via POST requests. This way, it invokes the model for predictions, ensuring seamless mainframe integration. The response informs the system whether to flag a transaction.

4.5 Edge Devices for Distributed Processing

Offloading inference to edge devices reduces computational load on mainframes. An example setup involves deploying TensorFlow Lite models on edge hardware for fraud detection.

Edge inference accelerates real-time analytics and lowers latency. However, it necessitates synchronization mechanisms with centralized mainframes to maintain consistency.

These solutions collectively modernize mainframes, addressing the specific issues of latency, inefficiency, and lack of scalability.

5. DISCUSSION AND RECOMMENDATIONS

The integration of AI/ML models into mainframes addresses critical limitations, but implementing this effectively requires targeted actions across multiple domains. Below are specific, actionable recommendations:

5.1 Enhance Real-Time Data Processing Capabilities

Mainframes must adopt hybrid architectures combining traditional batch processing with real-time data handling. This involves:

- Deploying Event-Driven Frameworks: Use Apache Kafka or RabbitMQ as message brokers to ingest and process high-velocity data streams in real time.
- Optimizing Models for Low Latency: Train and deploy optimized models using TensorFlow Lite or ONNX, enabling faster inferences.
- Establishing a High-Speed Data Pipeline: Implement fast communication interfaces like gRPC between the AI models and mainframe applications.

5.2 Strengthen Predictive Maintenance Mechanisms

To reduce downtimes, implement predictive maintenance tools by:

- Integrating Scalable Log Collectors: Use systems like Elastic Stack (ELK) for log aggregation.
- Automating Data Labeling: Use weak supervision tools like Snorkel for creating labeled datasets from mainframe logs.
- Running Models Continuously: Deploy autoencoders in streaming mode to monitor and reconstruct system logs in real time.

5.3 Advance Fraud Detection Systems

Dynamic and adaptive fraud detection is essential for mainframes managing financial transactions. Recommendations include:

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

- Deploying Multi-Layered AI Models: Combine supervised (LSTMs) and unsupervised (Autoencoders) learning for improved accuracy.
- Implementing Incremental Learning Pipelines: Update models dynamically as new transaction data arrives to account for emerging fraud patterns.
- Introducing Anomaly Scoring: Integrate anomaly scores into risk assessment dashboards for immediate fraud detection.

5.4 Scale for Modern Workloads

Modern workloads demand elastic scalability in mainframes, achievable through:

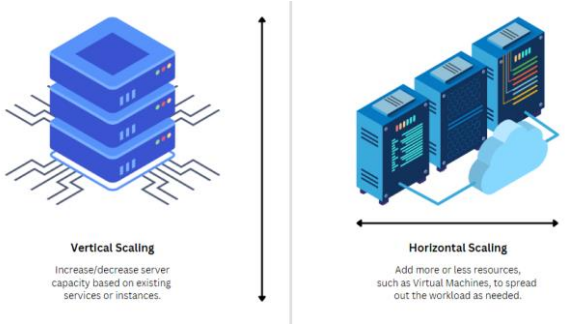

Figure 7: Horizontal and vertical scalability of mainframes

- Using Containerization: Deploy AI/ML models in Docker containers to isolate resource-intensive workloads from core processes.
- Integrating Cloud-Native Solutions: Offload compute-intensive tasks to cloud services like AWS SageMaker or Google AI Platform.
- Adopting Modular Architectures: Enable seamless plug-and-play functionality for new AI tools.

5.5 Improve Customer Segmentation Capabilities

AI-enhanced customer insights are achievable by:

- Consolidating Data Silos: Use data lakes to unify structured and unstructured customer data.
- Implementing Feedback Loops: Incorporate real-time user behavior into clustering algorithms to ensure dynamic segmentation.
- Enhancing Explainability: Use SHAP (SHapley Additive exPlanations) values to interpret model-driven segmentation.

5.6 Address Security Implications

AI-powered mainframes must address the security challenges of integration by:

- Building Governance Frameworks: Define clear protocols for AI model training, testing, and deployment to mitigate risks.
- Securing APIs and Data Pipelines: Encrypt API communications and anonymize data in transit.
- Regular Security Audits: Conduct periodic security assessments focusing on the AI/ML layers to prevent vulnerabilities.

6. CONCLUSION

Integrating AI/ML capabilities into mainframe systems is a transformative solution that addresses critical inefficiencies and enhances functionality in fraud detection, predictive maintenance, and customer segmentation. With the help of advanced frameworks and hybrid architectures, organizations can unlock new levels of scalability, security, and adaptability.

However, implementation requires a unique approach involving upskilling, governance, and continuous optimization to align legacy systems with modern requirements. The outcome is a future-ready system capable of thriving in analytics-driven ecosystems.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

REFERENCES

1. Shestak, Y., Tolupa, S., Torchylo, A., & Onyigwang, O. J. (2021, October). Analysis of Methods for Data Structuring in Data Centers. In 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 394-398). IEEE.
2. Hechler, E., Oberhofer, M., Schaeck, T., Hechler, E., Oberhofer, M., & Schaeck, T. (2020). The operationalization of AI. Deploying AI in the Enterprise: IT Approaches for Design, DevOps, Governance, Change Management, Blockchain, and Quantum Computing, 115-140.
3. Jordan, K. E. (1987). Performance comparison of large-scale scientific computers: Scalar mainframes, mainframes with integrated vector facilities, and supercomputers. Computer, 20(03), 10-23.
4. McKay, R., Pendleton, B., Britt, J., & Nakhavanit, B. (2019, March). Machine learning algorithms on botnet traffic: ensemble and simple algorithms. In Proceedings of the 2019 3rd international conference on compute and data analysis (pp. 31-35).
5. Xue, M., Yuan, C., Wu, H., Zhang, Y., & Liu, W. (2020). Machine learning security: Threats, countermeasures, and evaluations. IEEE Access, 8, 74720-74742.
6. Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., & Marchetti, M. (2018, May). On the effectiveness of machine and deep learning for cyber security. In 2018 10th international conference on cyber Conflict (CyCon) (pp. 371-390). IEEE.
7. Raghavan, P., & El Gayar, N. (2019, December). Fraud detection using machine learning and deep learning. In 2019 international conference on computational intelligence and knowledge economy (ICCIKE) (pp. 334-339). IEEE.
8. Baquero, J. A., Burkhardt, R., Govindarajan, A., & Wallace, T. (2020). Derisking AI by design: How to build risk management into AI development. McKinsey & Company.
9. Ziegler, M. M., Bertran, R., Buyuktosunoglu, A., & Bose, P. (2017). Machine learning techniques for taming the complexity of modern hardware design. IBM Journal of Research and Development, 61(4/5), 13-1.