nigeria computer society
www.ncs.org.ng

International Journal of Information Security, Privacy and Digital Forensics
Vol. 8, No. 2, December, 2024

A Hybrid Deep Learning and Symbolic AI for Anomaly Detection in Heterogeneous High-Performance Computing Systems

O.B. Okunoye
Department of Computer Science, University of Lagos, Akoka, Nigeria.
bokunoye@unilag.edu.ng

S. O. Oyebola
Department of Computer Science, University of Lagos, Akoka, Nigeria.
simeonoyekunleoyebola@gmail.com

Abstract: Anomaly detection in heterogeneous High Performance Computing (HPC) systems is challenging because the environment of multi-component hardware is complex. Traditional detection methods are not compatible with high-dimensional data and are heterogeneous. This study proposes a new hybrid approach by combining deep learning and symbolic artificial intelligence techniques to improve the accuracy and interpretability of anomaly detection. In this paper, the pattern recognition strengths of deep learning with the knowledge representation and reasoning of symbolic AI are combined, aiming at a unified framework that can achieve better precision in detection while realizing explainability. The framework is evaluated with HPC datasets based on performance metrics and qualitative expert assessments, bearing potential generalizability to other complex systems such as cyber-physical networks and IoT infrastructures.

Keywords: Anomaly detection, HPC systems, hybrid AI, deep learning, symbolic artificial intelligence, interpretability, heterogeneous computing.

1. Introduction

According to Siavash Ghiasvand and Ciorba (2019), an anomaly is any observed behaviour that differs from what is predicted. In contrast to failures, unexpected behaviours are not always wrong. Anomaly detection is a widely used method for identifying faults in computer systems (Park et al., 2018). Log analysis in HPC systems helps identify anomalies and avert failures (Siavash Ghiasvand et al., 2016). The identification of abnormal states is a difficult job. Real-time detection might be challenging due to streaming data's limitations (Velasquez et al., 2022).

High-performance computing systems have taken computational capabilities to new dimensions, unleashing a new generation of truly pioneering scientific discovery and industrial innovation. Their complex architecture is then fabricated on heterogeneous hardware comprising processors—CPUs, GPUs, accelerators—storage devices, and interconnected networks. The heterogeneity has some critical issues connected with it, especially in anomaly detection, because it influenced the system security, reliability, and optimal performance (Navaux, Lorenzon and Serpa, 2023). Traditional rule-based systems and statistical methods for anomaly detection in HPC systems usually fail because of the complexity management and high-dimensional data of modern HPC environments (Molan et al., 2023). Two major reasons this is so are that creating rules for rule-based systems requires much manual effort in system development and maintenance, which does not provide much flexibility or the ability to adapt to the continuous evolution of HPC systems (Thudumu et al., 2020). Meanwhile, although useful in some scenarios, statistical methods miss the complexity of the patterns and interconnections within the data. As a reaction to such inefficiencies, different researchers have explored the potential of advanced artificial intelligence techniques—deep learning and symbolic AI—to provide more effective solutions for anomaly detection (Saeed and Omlin, 2023).

The growth of high-performance computing systems has made reliability a challenge. In recent years, as high-performance computer systems have grown in size, their dependability has steadily declined. As a result, system resilience has been identified as one of the most significant difficulties for large-scale HPC

systems (Molan et al., 2021). Several sorts of fault situations or anomalies may occur during operation, ranging from failing hardware to incorrect setups or defective software. Currently, system administrators and end users need to manually find it. Automated fault detection is necessary for large-scale supercomputers and facilities (Borghesi et al., 2019b), since the current methodology does not scale. Several strategies and systems have been presented to guarantee that parallel programs are executed correctly and completed successfully.

Several anomaly detection approaches have been presented for several application areas (Selicato et al., 2021). LOF, kNNW, and ODIN are examples of neighborhood-based anomaly detection systems that use data points to identify abnormalities (Zhou et al., 2020). These strategies perform poorly, and are aware of their limitations. Another example is the use of ensemble-based anomaly detection algorithms. Zimek et al. (2014) and PasillasDíaz et al. (2016) developed an effective ensemble-based anomaly detection approach.

In our current high computing environments, the greatest challenge faced has become anomaly detection, which is described in terms of operation as the process of detecting deviations from desired behavior, mainly because they are heterogeneous and complicated (Li and Jung, 2023). In traditional methodologies, the complexity of these HPC higher-dimensional mechanisms has a lot of intricacies of the system. Subsequently, integrating techniques from deep learning and symbolic artificial intelligence resolved the problems caused by the limitations of individual approaches and elevated the synergy (Elhanashi et al., 2023).

Li et al. (2017) proposed a scheme to use deep learning to learn normal behavior from training data into some kind of cognitive representation and symbolic rule mining/pattern matching in rule(s) for anomaly detection via Autoencoder. ADDLR. ADDLR used an auto-encoder-based deep learning model to learn normal behavioral patterns from historical data. Then rules would be mined to detect anomalies according to deviations from these patterns. On the other hand, Zhang et al. (2018) introduced Deep Symbolic Anomaly Detection, or DSAD for short. This approach is based on rule mining in a convolutional neural network for anomaly detection in time series data of HPC applications. More recently, Liu et al. (2020) presented the Hybrid Anomaly Detection based on Deep Learning and Symbolic Rule Mining (HADDLR), which uses an ensemble of deep learning models to learn normal patterns of behaviors, followed by rule mining to spot anomalies.

These findings indicate that applying hybrid deep learning-symbolic AI approaches to anomaly detection in heterogeneous HPC systems performs well in terms of accuracy compared to classical statistical techniques, interpretability compared to pure deep learning ones, and reasoning emanating from symbolic AI components.

2. Related Works

Hybrid deep learning is an exciting approach to anomaly detection in Heterogeneous HPC systems. This could be made possible by looking for patterns within deep learning methods coupled with symbolic AI's interpretability, knowledge representation, and reasoning capabilities (AlHaddad et al., 2023). Well-designed hybrid models have the potential to deliver state-of-the-art performance with an appropriate balance between detection accuracy and interpretability for the approach selected. This impetus will be key to disentangling the challenges of current computing environments.

Velasquez et al. (2022) studied the hybrid machine-learning ensemble for detecting anomalies in a real-time Industry 4.0 system. This ensemble comprises two phases based on a normal industrial system: manufacturing and operations. The ensemble system outperformed individual ML sub-models (LOF, OCSVM, and Autoencoder) regarding the F1-score and AUC across three computers. The suggested approach supports the vertical scalability of the number of algorithms employed in the ensemble. The hybrid model can identify anomalies in real time, with a maximum calculation time of around 190 milliseconds. The autoencoder's poor F1-score suggests testing various algorithms (e.g., isolation forest, elliptic envelope) to enhance assembly performance. Further research is needed to assess the computing costs of retraining various algorithms.

The research suggested investigating the system of created machines at various degrees of deterioration. Data imputation research is recommended to produce synthetic samples for systems with missing information since communication failures are typical in industrial settings. The research did not use deep learning approaches while developing meta-classifiers utilising basic classifiers such as recurrent neural networks (LSTMs).
In addition, the study focuses on detecting single-type anomalies, which remains a challenge that needs to be addressed.

Chliah, Battou, Ait el Hadj, and Laoufi (2023) conducted a study on a hybrid strategy that uses both supervised and unsupervised learning to find abnormalities using the Apache Spark big data engine.

The Sparks MLlib first clustered the network data using the K-means technique. After that, the K-nearest neighbours (KNN) method was used to identify and discover abnormalities in each cluster. The model was trained and verified using a real dataset from Ibn Zohr University. Their findings revealed that the suggested

model beat certain well-known algorithms in correctly identifying irregularities on the dataset. The experimental findings demonstrate that the recommended hybrid strategy achieves a maximum accuracy of 99.94% when the k-fold cross-validation procedure is applied to the whole dataset encompassing all 48 features. Pazho et al. (2023) studied the ability of graph-based methods to identify abnormalities in distributed systems. This was a combination of distinct kinds or made up of the same type, resulting in different sets of criteria. It dived precisely into graph-based approaches to identify possibilities for dealing with the difficult parts of the actual world within a conceptual analysis marked by heterogeneity and dynamic structure. The research conducted an in-depth analysis of current and advanced writings on the issue, determining similarities and contrasts based on their aspects. It provided three systems at various levels of abstraction to facilitate a more thorough understanding. The research further delved into the unique issues of identifying anomalies in these systems.

Gao et al. (2019) compare the traditional threshold-based anomaly detection approach for HPC systems with a hybrid approach using isolation forests, an anomaly detection algorithm with an additional overlay for domain knowledge encoded in rules. In doing so, they can demonstrate that such a hybrid approach ensures higher accuracy in anomaly detection, particularly new previously unseen anomalies. The method created data preparation models to manage outliers and missing data. As a result, an imbalance removal model known as GRU-MACGANS was developed to supplement minority category data. The SA-CNN-BiGRU-FCN model, which included a knowledge transfer technique, was used to detect anomalous behaviour. Finally, the findings of the SA-CNN-BiGRU-FCN model were analysed using the contrastive explanations approach. The study employed a case study of a low-voltage distribution system to show the usefulness of this strategy, indicating that data pretreatment models and GRU-MACGANS improve data quality and relieve category imbalance difficulties in anomaly detection.

Ferrag, Shu, Djallel, and Choo (2021) examined the effectiveness of deep learning-powered intrusion detection systems (IDS) in countering distributed denial of Service (DDoS) attacks within the framework of Agriculture 4.0. The study suggested and assessed three different Intrusion Detection System (IDS) models: a model based on convolutional neural network (CNN), a model based on deep neural network (DNN), and a model based on recurrent neural network (RNN). These models were evaluated using two well-known real-world traffic datasets: the CIC-DDoS 2019 dataset and the TON_IoT dataset. The test used binary and multiclass classification scenarios to thoroughly assess the models' performance across various types of traffic. Regarding performance evaluation, the CNN-based IDS model demonstrated exceptional results, obtaining impressive accuracy rates of 99.95% for binary traffic identification and 99.92% for multiclass traffic detection. This performance's superiority was especially notable compared to conventional machine learning methods, which included decision trees, random forests, naive Bayes, and logistic regression. The evaluation criteria encompassed a range of crucial performance measures for IDS systems, such as detection rate, false alarm rate, precision, F-score, recall, true negative rate, false accept rate, ROC Curve analysis, and overall accuracy. The results highlighted the greater effectiveness of deep learning algorithms in accurately detecting and reducing DDoS assaults in agricultural IoT systems.

3. Workings of Hybrid Deep Learning Model on Multi-Sensor Dataset

The architecture that integrates Hybrid Deep Learning with Symbolic AI for Anomaly Detection in Heterogeneous HPC Systems follows a structured approach consisting of multiple layers. Each layer has some specific functionality to get the data effectively processed, analyzed, and detect anomalies. The hybrid approach follows the strengths of deep learning and symbolic AI. Deep learning has already delivered unrivalled performance related to pattern recognition and the extraction of features from raw data. To a great extent, it often lacks interpretability, which is one of the key factors in decision-making. Symbolic artificial intelligence deals with knowledge representation and reasoning; interpretability is the cornerstone for offering insights into domains. Combining these two paradigms will make the system balance accuracy with interpretability, improving anomaly detection's reliability and effectiveness on HPC systems.

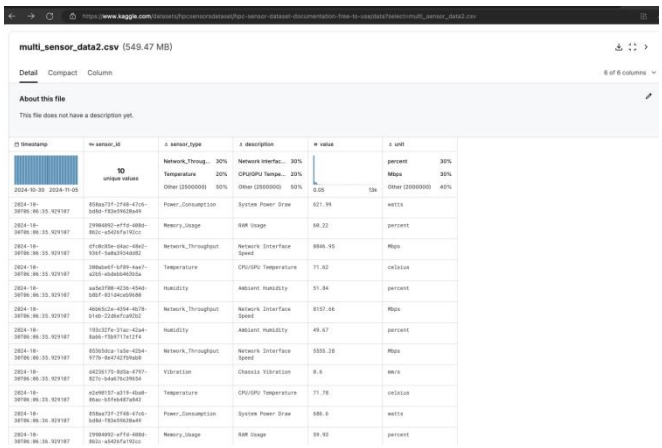


Figure 1: Multi-sensor dataset with over 5 million columns (source:https://www.kaggle.com/datasets/hpcsensorsdataset/hpc-sensor-dataset-documentation-free-to-use/data?select=multi_sensor_data2.csv)

The system would involve eight critical layers as follows:
1. Data Collection Layer
2. Data Preprocessing Layer
3. Feature Engineering Layer
4. Hybrid AI Model Layer
5. Model Training and Optimization Layer
6. Anomaly Detection Layer
7. Alerting and Visualization Layer
7. Alerting and Visualization Layer
8. Feedback Loop and Adaptation

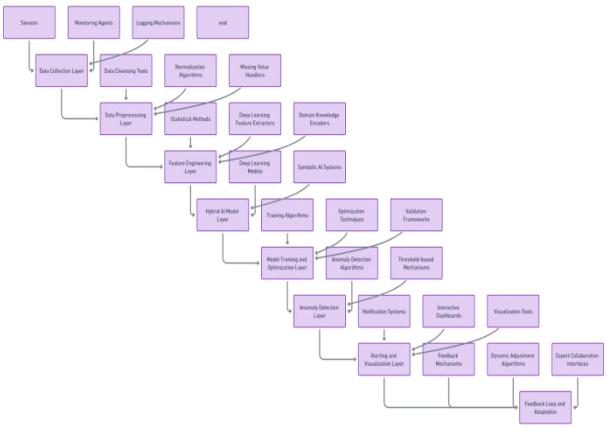


Figure 2: Architecture of the hybrid deep learning and symbolic AI for anomaly detection in heterogeneous high-performance computing systems.

Data Collection Layer
Function: This layer is supposed to gather all data sources in the HPC system.
Components: The components in this layer are sensors, monitoring agents, and logging mechanisms.

Data Preprocessing Layer
Function: Cleaning, filtering, and aggregation of raw data. The components include data cleansing tools, normalization algorithms, and missing value handlers.

Feature Engineering Layer
Function: Extract relevant features from the preprocessed data.
Components: Statistical methods, deep learning feature extractors, and domain knowledge encoder.

Hybrid AI Model Layer
Function: Development and integration of deep learning and symbolic AI models.
Components: Deep learning models, such as CNNs and RNNs; symbolic AI systems, including rule-based and expert systems.

Model Training and Optimization Layer
Function: This layer trains and optimizes our hybrid AI model.
Components: The training algorithms, optimization techniques, and validation frameworks.

Anomaly Detection Layer
Function: The layer detects anomalies against the data on the streaming system.
Components: Anomaly detection algorithms, such as Isolation Forest, One-Class SVM, autoencoders, and threshold-based mechanisms.

Alerting and Visualization Layer
Function: Alert and visualize anomalies detected.
Components: Notification systems, interactive dashboards, visualization tools.

Feedback Loop and Adaptation
Function: Constantly renew and adapt the hybrid AI model.
Components: Feedback mechanisms, dynamic adjustment algorithms, expert collaboration interfaces

WORKFLOW:
 # Data Collection Layer
Initialize sensors, monitoring agents, and logging mechanisms
Collect data from sensors
Collect data from monitoring agents
Collect data from logging mechanisms
Store collected data

 # Data Pre-processing Layer
Initialize data cleansing tools
Initialize normalization algorithms
Initialize missing value handlers
Cleanse the collected data using data cleansing tools
Normalize the cleansed data using normalization algorithms
Handle missing values in the normalized data

 # Feature Engineering Layer
Initialize statistical methods for feature extraction
Initialize deep learning feature extractors
Initialize domain knowledge encoders
Extract features from preprocessed data using statistical methods
Extract features from preprocessed data using deep learning feature extractors
Encode domain-specific knowledge into features using domain knowledge encoders
Combine all extracted features

 # Hybrid AI Model Layer
Initialize deep learning models (e.g., CNN, RNN, Transformers)

Initialize symbolic AI systems (e.g., rule-based systems, expert systems)
Process features using deep learning models
Process features using symbolic AI systems
Integrate deep learning and symbolic AI outputs to form a hybrid model

 # Model Training and Optimization Layer
Initialize training algorithms
Initialize optimization techniques
Initialize validation frameworks
Train the hybrid AI model using historical data
Optimize the trained model using optimization techniques
Validate the optimized model using validation frameworks

 # Anomaly Detection Layer
Initialize anomaly detection algorithms (e.g., Isolation Forest, One-Class SVM, Autoencoders)
Initialize threshold-based mechanisms
Deploy the trained hybrid AI model for real-time data analysis
Detect anomalies using anomaly detection algorithms
Apply threshold-based mechanisms to filter detected anomalies

 # Alerting and Visualization Layer
Initialize notification systems
Initialize interactive dashboards
Initialize visualization tools
Trigger alerts and notifications for detected anomalies
Update interactive dashboards with real-time anomaly data
Visualize anomalies using visualization tools

 # Feedback Loop and Adaptation Layer
Initialize feedback mechanisms
Initialize dynamic adjustment algorithms
Initialize expert collaboration interfaces
Collect feedback from system administrators and domain experts
Adjust hybrid AI model parameters based on feedback and new data
Collaborate with experts to refine anomaly detection strategies
Update the hybrid AI model continuously with new data and insights

 # Main Workflow
Begin the main workflow
Collect raw data from the data collection layer
Preprocess collected data in the data preprocessing layer
Perform feature engineering on preprocessed data
Create a hybrid AI model using the feature engineering layer
Train and optimize the hybrid AI model with historical data
While (system is running):
  Collect real-time data
  Detect anomalies using the anomaly detection layer
  Alert and visualize detected anomalies
  Update the hybrid AI model based on feedback and new data
End the main workflow

4. Simulations and Results

4.1. Simulation setup

1 Data Processing Layer

1.1 Data Cleansing

The data cleansing step ensures the data's integrity by removing duplicates and correcting data types. Formally, given a dataset $D = \{x_1, x_2, \ldots, x_n\}$, we apply the cleansing operation:

$$D' = \text{cleanse}(D) = \{x \in D \mid x \text{ is unique and correctly typed}\}$$

1.2 Missing Value Imputation

Let X be a matrix representing our data with missing entries. We use an imputer I such that:

$$X_{imputed} = I(X) = \begin{cases} \bar{x}_j, & \text{if } x_{ij} \text{ is missing} \\ x_{ij}, & \text{otherwise} \end{cases}$$

where $\bar{x}_j$ is the mean of the j-th column for numerical columns.

1.3 Normalization

Normalization is applied using a StandardScaler:

$$X_{normalized} = \frac{X - \mu}{\sigma}$$

where μ is the mean and σ is the standard deviation of the dataset.

2 Feature Engineering Layer

2.1 Statistical Feature Extraction

Given a feature vector $X = \{x_1, x_2, \ldots, x_n\}$, the statistical features are:

$$\text{Mean: } \mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

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

$$\text{Standard Deviation: } \sigma = \sqrt{\sigma^2}$$

2.2 Autoencoder for Feature Extraction

An Autoencoder consists of an encoder-decoder structure. Given input data X, the encoding operation is:

$$h = f_\theta(X) = \text{ReLU}(W_e X + b_e)$$

where W_e and b_e are the weights and biases of the encoder. The decoder reconstructs the input:

$$\hat{X} = g_\phi(h) = \text{Sigmoid}(W_d h + b_d)$$

2.3 Rule-based Feature Encoding

Rule-based features use conditional logic:

$$\text{High CPU Usage} = \begin{cases} 1, & \text{if CPU} > 90\% \\ 0, & \text{otherwise} \end{cases}$$

3 Hybrid Model Layer

3.1 CNN-RNN Architecture

The CNN applies convolutional operations:

$$y_{ij}^{(l)} = \sigma\left(\sum_{m=0}^{k-1}\sum_{n=0}^{k-1} w_{mn}^{(l)} x_{(i+m)(j+n)}^{(l-1)} + b^{(l)}\right)$$

where $w_{mn}^{(l)}$ are the convolution filter weights, and σ is the activation function.
The RNN processes the sequence output:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b_h)$$

$$y_t = W_{hy} h_t + b_y$$

4 Model Optimization Layer

Model optimization minimizes the loss function:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

y

5 Anomaly Detection Layer

5.1 Isolation Forest

Isolation Forest isolates observations by random partitions. The anomaly score is:

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

where $E(h(x))$ is the average path length, and $c(n)$ is the average path length of a binary search tree.

5.2 Autoencoder-Based Detection

Anomaly score based on reconstruction error:

$$\text{Error} = \|X - \hat{X}\|^2$$

6 Alert Visualization Layer

6.1 Real-time Monitoring

The system reads data points and checks for anomalies:

$$\text{Anomaly Alert} \Rightarrow \text{Send Email if } s(x) > \text{Threshold}$$

4.2 Results

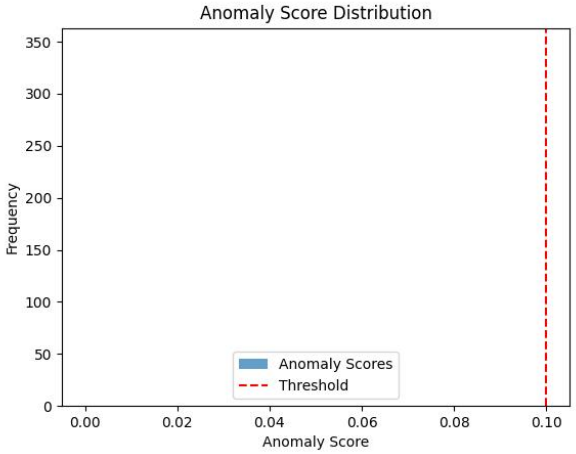


Figure 3: Anomaly score distribution plot from hybrid deep learning and symbolic AI for anomaly detection in HPC systems (Source: Oyebola 2024) trained model

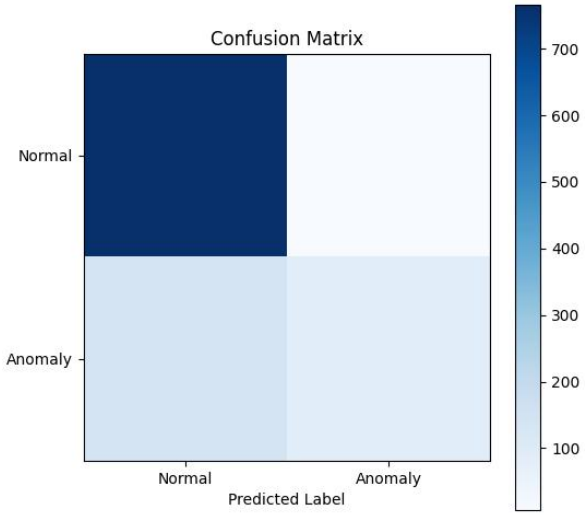


Figure 4: Confusion matrix plot from hybrid deep learning and symbolic AI for anomaly detection in HPC systems trained model showing normal data against anomalies from the trained model and dataset (Source: Oyebola 2024)

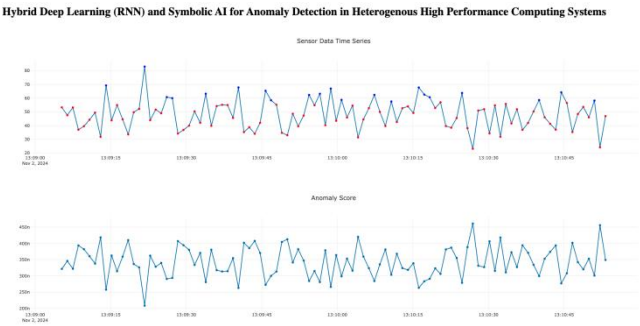


Figure 5: Real-Time dashboard reading sensor values in real-time using a time-series dataset from hybrid deep learning and symbolic AI for anomaly detection in HPC systems (Source: Oyebola 2024).

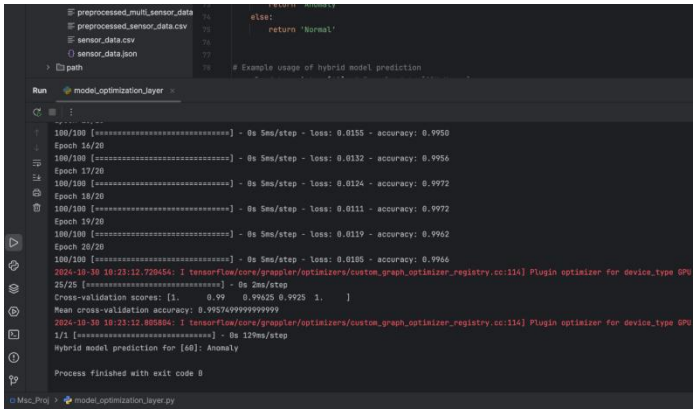


Figure 6: Model prediction score after training provided dataset (Source: Oyebola 2024)

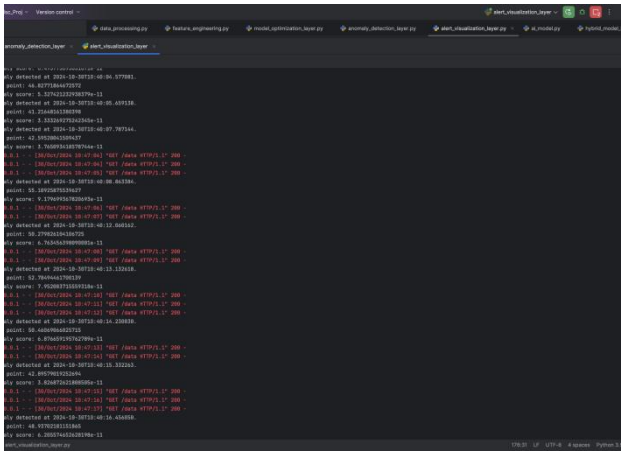


Figure 6: Real-time logs feeding dashboard from trained model (Source: Oyebola 2024)

Table 1: Performance comparison of hybrid deep learning and symbolic AI models and OCSVM, LOF algorithms in low-traffic load.

Model	Accuracy Score
LOF	0.924
OCSVM	0.922
Hybrid Model	0.9957

5. Conclusions

The architecture for integrating hybrid deep learning and symbolic AI in anomaly detection for heterogeneous HPC systems provides a comprehensive framework that addresses the limitations of traditional methods. By leveraging the strengths of both deep learning and symbolic AI, the system aims to achieve enhanced accuracy, interpretability, and adaptability. This hybrid approach is poised to significantly improve the reliability and performance of anomaly detection in complex HPC environments, contributing to more secure and efficient computing infrastructures with a mean accuracy score of 0.9927, significantly outperforming other models such as the OCSVM and LOF models.

References

AlHaddad, U., Basuhail, A., Khemakhem, M., Eassa, F.E. and Jambi, K. (2023). Ensemble Model Based on Hybrid Deep Learning for Intrusion Detection in Smart Grid Networks. Sensors, [online] 23(17), p.7464. doi:https://doi.org/10.3390/s23177464.

Borghesi, A., Libri, A., Benini, L. and Bartolini, A. (2019b). Online Anomaly Detection in HPC Systems. arXiv (Cornell University), [online] 15, pp.229–233. doi:https://doi.org/10.1109/aicas.2019.8771527.

Elhanashi, A., Dini, P., Saponara, S. and Zheng, Q. (2023). Integration of Deep Learning into the IoT: A Survey of Techniques and Challenges for Real-World Applications. Electronics, [online] 12(24), p.4925. doi:https://doi.org/10.3390/electronics12244925.

Ferrag, M.A., Shu, L., Djallel, H. and Choo, K.-K.R. (2021). Deep Learning-Based Intrusion Detection for Distributed Denial of Service Attack in Agriculture 4.0. Electronics, 10(11), p.1257. doi:https://doi.org/10.3390/electronics10111257.

Gao, B., Kong, X., Li, S., Chen, Y., Zhang, X., Liu, Z., & Lv, W. (2024). Enhancing anomaly detection accuracy and interpretability in low-quality and class imbalanced data: A comprehensive approach. Applied Energy, 353(B), 122157. Available at: https://doi.org/10.1016/j.apenergy.2023.122157

Li, G. and Jung, J.J. (2023). Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. Information Fusion, 91, pp.93–102. doi:https://doi.org/10.1016/j.inffus.2022.10.008

Molan, M., Borghesi, A., Cesarini, D., Benini, L. and Bartolini, A. (2023). RUAD: Unsupervised anomaly detection in HPC systems. Future Generation Computer Systems, 141, pp.542–554. doi:https://doi.org/10.1016/j.future.2022.12.001.

Navaux, P.O.A., Lorenzon, A.F. and Serpa, M. da S. (2023). Challenges in High-Performance Computing. Journal of the Brazilian Computer Society, [online] 29(1), pp.51–62. doi:https://doi.org/10.5753/jbcs.2023.2219.

Park, B.H., Hui, Y., Boehm, S., Ashraf, R.A., Layton, C. and Engelmann, C. (2018). A Big Data Analytics Framework for HPC Log Data: Three Case Studies Using the Titan Supercomputer Log. OSTI OAI (U.S. Department of Energy Office of Scientific and Technical Information). [online] doi:https://doi.org/10.1109/cluster.2018.00073.

Pasillas-Díaz, J.R. and Ratté, S. (2016). Bagged Subspaces for Unsupervised Outlier Detection. Computational Intelligence, 33(3), pp.507–523. doi:https://doi.org/10.1111/coin.12097.

Pazho, A. D., Noghre, G. A., Purkayastha, A. A., Vempati, J., Martin, O., & Tabkhi, H. (2023). A Survey of Graph-based Deep Learning for Anomaly Detection in Distributed Systems. arXiv:2206.o49v2[cs.LG] 1 June 2023.

Saeed, W. and Omlin, C. (2023). Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. Knowledge-Based Systems, 263, p.110273. doi:https://doi.org/10.1016/j.knosys.2023.110273.

Siavash Ghiasvand, Ciorba, F.M., Tschuter, R. and Nagel, W.E. (2016). Lessons Learned from Spatial and Temporal Correlation of Node Failures in High Performance Computers. edoc (University of Basel). doi:https://doi.org/10.1109/pdp.2016.101.

Srivastava, A., & Bharti, M. R. (2023). Hybrid Machine Learning Model for Anomaly Detection in Unlabelled Data of Wireless Sensor Networks. Springer Link Volume 129,2693–2710. https://link.springer.com/article/10.1007/s11277-023-

Siavash Ghiasvand and Ciorba, F.M. (2019). Anomaly Detection in High Performance Computers: A Vicinity Perspective. arXiv (Cornell University). doi:https://doi.org/10.1109/ispdc.2019.00024.

Thudumu, S., Branch, P., Jin, J. and Singh, J. (Jack) (2020). A comprehensive survey of anomaly detection techniques for high dimensional big data. Journal of Big Data,

Selicato, L., Esposito, F., Gargano, G., Maria Carmela Vegliante, Opinto, G., Gian Maria Zaccaria, Ciavarella, S., Guarini, A. and Nicoletta Del Buono (2021). A New Ensemble Method for Detecting Anomalies in Gene Expression Matrices. Mathematics, 9(8), pp.882–882. doi:https://doi.org/10.3390/math9080882.

Velasquez, D., Perez, E., Oregui, X., Artetxe, A., Manteca, J., Mansilla, J.E., Toro, M., Maiza, M. and Sierra, B. (2022). A Hybrid Machine-Learning Ensemble for Anomaly Detection in Real-Time Industry 4.0 Systems. IEEE Access, 10, pp.72024–72036. doi:https://doi.org/10.1109/access.2022.3188102.

Zhou, X., Hu, Y., Liang, W., Ma, J. and Jin, Q. (2020). Variational LSTM Enhanced Anomaly Detection for Industrial Big Data. IEEE Transactions on Industrial Informatics, pp.1–1. doi:https://doi.org/10.1109/tii.2020.3022432.

Zimek, A., Campello, R.J.G.B. and Sander, J. (2014). Ensembles for unsupervised outlier detection. ACM SIGKDD Explorations Newsletter, 15(1), pp.11–22. doi:https://doi.org/10.1145/2594473.2594476

10253-2

7(1). doi:https://doi.org/10.1186/s40537-020-00320-x.