Vishal Gotarane[1*]
Rajiv Iyer[2]

Optimizing Energy-Efficient Machine Learning Algorithms for Real-Time Attack Detection in IoT Devices

Journal of Electrical Systems

Abstract: Energy efficiency is a critical challenge in the expanding domain of In-ternet of Things (IoT) networks, where resource-constrained devices must operate securely under strict energy limitations. This study explores the application of Levy-Based Moth-Flame Optimization (LB-MFO) to enhance intrusion detection in IoT systems. Using the CICIDS 2017 dataset, LB-MFO was evaluated against standard machine learning models, including Logistic Regression, Decision Trees, Random Forest, and Support Vector Machines. To further optimize energy usage, techniques such as pruning, quantization, and model compression were employed. The results demonstrate that LB-MFO consistently outperformed other models, achieving 96% accuracy with a performance score of 0.98 while consuming only 0.018–0.020 W with minimal latency of 14–15 ms under optimization. These findings highlight LB-MFO's potential to deliver accurate and energy-efficient intrusion detection, making it ideal for real-time IoT applications.

Keywords: Energy-efficient IoT, Intrusion Detection, Levy-Based Moth-Flame Opti-mization, Machine Learning, Pruning, Quantization, Model Compression.

1 Introduction

Applications for the Internet of Things (IoT) are expanding across a number of industries, including smart cities, healthcare, and agriculture. IoT networks are growing quickly, yet this presents serious security risks. Drop attacks, replay assaults, tamper attacks, and multi-max attacks are just a few of the many cyberattacks that can target IoT devices because of their open and dispersed nature. For IoT systems to remain secure, dependable, and intact, these assaults must be effectively detected and mitigated.

Typically, signature-based or anomaly-based techniques are used by traditional intrusion detection systems (IDS) to identify intrusions. These methods are useful, but they frequently have trouble seeing new or unknown attack routes, which makes them less useful in dynamic IoT situations. However, machine learning (ML) techniques have drawn a lot of interest as a viable solution because of their capacity to identify anomalies that have not yet been noticed and discover intricate patterns in network traffic.

This research presents a novel method that makes use of machine learning models that have been tuned using the Levy-Based Moth Flame Optimization (LB-MFO) algorithm. The goal of this hybrid approach is to maximize the IDS's energy efficiency and detection precision. We address the twin concerns of high performance and lower energy consumption in resource-constrained IoT scenarios by implementing energy-efficient approaches including pruning, quantization, and model compression. Our method is a reliable solution for IoT network security since it improves IDS detection capabilities while guaranteeing that IoT device energy consumption is kept to a minimum.

2 Related Work

Energy consumption remains one of the primary challenges in deploying machine learning models for IoT security. IoT devices are often battery-powered and limited in both computational resources and memory, making traditional security solutions unsuitable for these environments (13). Tahir et al. (4) highlight how the constrained nature of IoT devices complicates security, noting that lightweight machine learning models can mitigate some of these limitations by reducing computational requirements. The need for energy-efficient algorithms is further emphasized by Xu et al. (12), who discuss the importance of balancing detection accuracy and energy consumption in real-time security monitoring.

Research on model optimization focuses heavily on techniques like pruning, quantization, and compression to address IoT devices' power constraints. Liu et al. (14) examine how pruning—a technique that removes redundant neurons and connections—reduces energy consumption while maintaining detection accuracy. Pruning allows neural networks to operate with a lower memory footprint and faster inference times, which are crucial for low-power IoT devices. Similarly, Han et al. (6) introduce "deep compression," a combination of pruning, quantization, and Huffman coding, which collectively reduces model size and energy requirements without sacrificing performance.

Quantization, as discussed by He et al. (20), reduces the precision of model weights, converting them from 32-bit floating-point to lower-precision formats. This reduces the energy required for computation and storage. Blalock et al. (18) review various state-of-the-art pruning and quantization methods, noting that these techniques can provide substantial energy savings in real-world IoT deployments by streamlining model architectures.

Lightweight models, particularly those optimized for minimal computation, play a crucial role in IoT security by reducing the energy demands of intrusion detection systems. Sun et al. (16) explore deep learning models tailored to resource-constrained IoT devices, finding that lightweight architectures like shallow neural networks and decision trees are

[1*]Research Scholar , Amity School of Engineering and Technology, Amity University Maharashtra, Mumbai. Maharashtra -410206, India
[2]Associate Professor , Amity School of Engineering and Technology, Amity University Maharashtra, Mumbai. Maharashtra -410206, India

effective for energy-efficient intrusion detection. Zhao and Sun (19) further support the use of lightweight models, specifically designed to consume less power, as essential for achieving sustainable IoT security.

Another strategy involves adaptive models that can dynamically adjust based on the current energy availability and threat level. Wang and Zhu (15) propose a machine learning framework that modifies model complexity according to energy constraints. By enabling IoT devices to reduce the frequency or intensity of detection under low-energy conditions, adaptive models extend device lifespan without compromising essential security measures.

Offloading computation to edge servers or adopting federated learning are emerging trends that support energy-efficient IoT security. Xie et al. (21) highlight that edge computing can reduce energy consumption on IoT devices by offloading resource-intensive computations to nearby servers. This approach not only conserves energy but also improves response times by reducing network latency.

Federated learning, as discussed by Shafiq et al. (8), enables IoT devices to collaboratively train models without centralizing data. This distributed approach reduces the need for each device to perform complete computations, thus saving energy across the network. Tang et al. (17) further explore federated learning and edge computing, emphasizing that distributed training allows IoT devices to optimize energy usage
while enhancing model performance in detecting security threats.

In addition to static optimizations, adaptive approaches that manage energy dynamically are increasingly important for sustainable IoT security. Gondhi and Venkat (5) propose adaptive sampling techniques that allow IoT devices to vary the rate of data collection based on security risk, saving energy during periods of low threat. Chen et al. (23) further explore reinforcement learning-based methods to adapt model complexity based on real-time energy constraints. This enables IoT devices to respond to security threats effectively while conserving energy by adjusting their computational intensity according to resource availability.

Rahman et al. (9) introduce a lightweight security model specifically designed for healthcare IoT systems, demonstrating that energy-aware adaptations can be implemented in sensitive environments. These models prioritize energy efficiency without sacrificing the stringent security standards required in healthcare, which is often applicable to broader IoT contexts.

Future IoT security systems may benefit greatly from integrating model compression, adaptive strategies, and distributed computing (via edge processing or federated learning), according to the literature. Researchers such as Nanda et al. (24) support hybrid models that dynamically distribute resources throughout IoT networks, integrating centralized and edge processing to improve accuracy and energy efficiency. According to Rong and Hu (22), combining lightweight model architectures with real-time adaptation can offer a well-rounded strategy for energy conservation that successfully meets security requirements.

3 Methodology

The proposed system is built to detect real-time attacks while optimizing for energy efficiency, using Leavy-Based Moth Flame Optimization (LB-MFO) for hyperparameter tuning of various machine learning models. Fig 1 illustrates an architecture of optimized Intrusion Detection System (IDS) tailored for IoT environments, emphasizing both accuracy and energy efficiency. It begins with initial data preprocessing and detection stages to prepare IoT data for analysis. The LB-MFO Optimization layer then fine-tunes the IDS model by leveraging the Levy-based Moth Flame Optimization algorithm, which balances high detection performance with low energy use. Further energy optimization techniques—pruning, quantization, and model compression—are applied to streamline the model for resource-constrained IoT devices. The final output is an IDS optimized for both security and efficient energy consumption, suitable for IoT deployments.

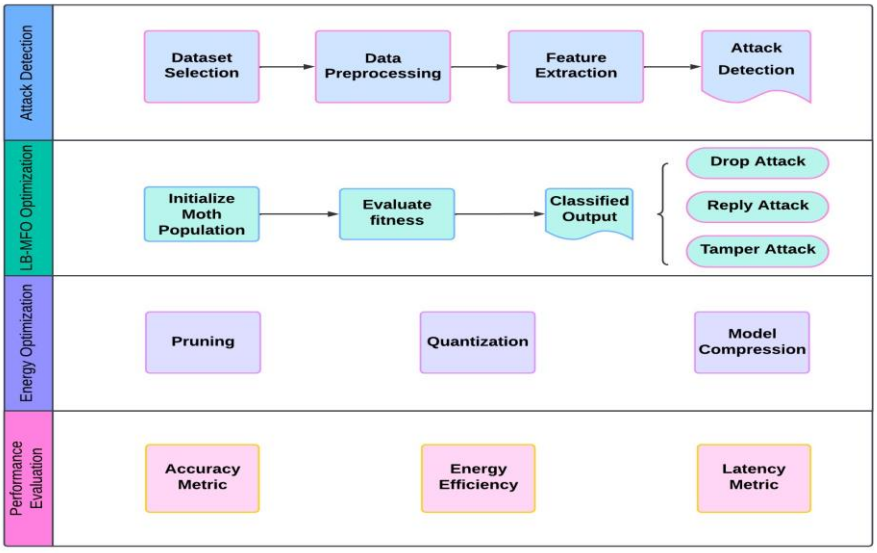

Figure 1: Optimized Architecture of IDS for IoT using LB-MFO and Energy Optimization Techniques

3.1 Leavy-Based Moth Flame Optimization (LB-MFO)

The Leavy-Based Moth Flame Optimization (LB-MFO) algorithm is inspired by the foraging behavior of moths, which navigate towards light sources (flames). In the context of IDS, LB-MFO is used to optimize the hyperparameters of machine learning models. These hyperparameters include tree depth for Decision Trees, kernel parameters for SVM, and the number of trees in Random Forest. The LB-MFO algorithm improves the models' performance by searching for the optimal configuration that minimizes energy consumption while maintaining high accuracy in detecting intrusions.

3.2 Objective Function for LB-MFO Optimization

The optimization goal is to maximize the energy efficiency of the IDS while maintaining or improving its accuracy. The objective function J for LB-MFO is defined as a weighted combination of the accuracy A of the machine learning model and its energy consumption E:

$J = \alpha \cdot A - \beta \cdot E$

where:

- A represents the accuracy of the model,
- E is the energy consumption per inference,
- α and β are weights that control the trade-off between accuracy and energy efficiency.

The accuracy A is defined as the proportion of correctly classified instances (both True Positives and True Negatives) over the total number of samples. Mathematically, this is expressed as:

= True Positives + True Negatives A

Total Samples

The goal of the optimization is to minimize J by adjusting the hyperparameters of the models so that the IDS can achieve high accuracy without consuming excessive energy.

3.3 Position Update Mechanism

The moths (candidate solutions) in the algorithm are iteratively updated based on their proximity to the "flames" (optimal solutions) in the search space. Each moth represents a set of hyperparameters for the machine learning models, and its position in the search space is updated according to the following formula:

$$Mit{+}1 = F_j^t + D_i \cdot e^{b \cdot l} \cdot \cos(2\pi l)$$

where:

- M_i^{t+1} is the new position of moth M_i at iteration $t + 1$,
- F_j^t is the position of the flame at iteration t,
- D_i is the distance between moth M_i and flame F_j, calculated as $D_i = \|_{Mi} - {}_{Fj}\|$,
- b is a constant that controls the spiral shape of the path,
- l is a random number in the interval $[-1,1]$ to induce variability in the search process.

This spiral path allows the algorithm to balance exploration (searching new areas of the solution space) and exploitation (refining the search around current solutions).

3.4 Adaptive Flame Count Reduction

The LB-MFO algorithm also adapts the number of active flames over iterations. Initially, all flames are active, but as the algorithm progresses, the number of active flames is reduced, thereby refining the search towards the optimal solutions. This adaptive mechanism improves convergence by focusing the search on the most promising areas of the solution space. The number of active flames at any iteration t is given by:

$$N\text{flames} = \text{round}\left(N_{\max} - \frac{t \cdot (N_{\max} - 1)}{T_{\max}}\right)$$

where:

- $N_{\max}$ is the initial number of flames,
- t is the current iteration,
- $T_{\max}$ is the maximum number of iterations.

3.5 Energy Efficiency

In addition to attack detection, several energy optimization techniques were applied to improve the efficiency of the machine learning models. These techniques include:

3.5.1 Pruning

Pruning is a method used to remove unnecessary branches from decision trees and random forests. This reduces the size of the models, speeds up inference, and enhances energy efficiency. In decision trees, branches that do not significantly contribute to classification accuracy are pruned, leading to smaller, faster models that consume less energy.

3.5.2 Quantization

Quantization involves reducing the precision of the model's parameters by converting floating-point values to lower-precision integers. This technique reduces memory and computational requirements, resulting in a more energy-efficient model. While quantization is primarily used in deep learning models, it was also applied to traditional machine learning models in this study to reduce their computational overhead and energy consumption.

3.5.3 Model Compression

Model Compression aims to reduce the size of machine learning models without significantly sacrificing their accuracy. Techniques such as weight sharing, low-rank factorization, and knowledge distillation were used to compress the models. By reducing the number of parameters, these techniques minimize both memory and computational overhead, leading to lower energy consumption during model inference.

The energy efficiency E_{eff} of the model is defined as the ratio of the model's accuracy A to its energy consumption E. This is calculated as:

$A\ E\text{eff} = \_\_E$

where:

- E_{eff} represents the energy efficiency of the model,
- A is the accuracy, and
- E is the energy consumption of the model during the inference.

LB-MFO aims to maximize E_{eff} by fine-tuning the hyperparameters such that energy consumption is minimized while maintaining or improving detection accuracy.

3.6 Termination Condition

The LB-MFO algorithm terminates when one of the following conditions is met:

- The objective function J converges to a satisfactory level (i.e., a balance of high accuracy and low energy consumption),
- The maximum number of iterations $T_{\max}$ is reached.

Algorithm 1 Levy-Based Moth Flame Optimization (LB-MFO)

1: **Initialize parameters:**
2: Set $N_{\max}$, $T_{\max}$, α, β
3: Initialize Moth Population M and Flame Population F
4: **Evaluate fitness** for each moth M_i: 5: Calculate accuracy A and energy E
6: Compute objective function $J = \alpha \cdot A - \beta \cdot E$
7: **Sort moths** by fitness and select top Nflames as active flames.
8: **for** $t = 1$ to Tmax **do** 9: **for** each moth M_i **do**
10: Select flame F_j
11: Calculate distance $D_i = \|M_i - F_j\|$ 12: Update position: $M_i^{(t+1)} = F_j^t + D_i \cdot \exp(b \cdot l) \cdot \cos(2\pi l)$
13:Re-evaluate fitness and accept new position if better
14:**Update flames:** Adjust active flames based on top moth positions
15:Reduce active flames: Nflames = round(Nmax − (t · (Nmax − 1))/Tmax)
16: **Termination:** Stop if $T_{\max}$ iterations are reached or if objective function J converges.
17: **Output:** Return optimal hyperparameters and use them for configuring detection models in the IoT system.

4 Experimental Setup

This section outlines the experimental setup used to evaluate the performance of the Leavy-Based Moth Flame Optimization (Lb-MFO) algorithm in detecting attacks in IoT networks and optimizing energy efficiency. The CICIDS 2017 dataset was used to simulate real-world attack scenarios, with the Lb-MFO algorithm evaluated alongside several machine learning models, including Logistic Regression, Decision Trees, Random Forest, and SVMs (SVM). Furthermore, energy optimization techniques like Pruning, Quantization, and Model Compression were applied to reduce the energy consumption of the models.

4.1 Dataset Selection

The CICIDS 2017 dataset was chosen due to its comprehensive diversity in network traffic, which includes both normal and malicious patterns. The dataset includes four main attack types:

- **Drop Attack**: Packets are intentionally dropped, causing disruptions in network communication.
- **Reply Attack**: Involves capturing and replaying legitimate packets to gain unauthorized access.
- **Tamper Attack**: Data packets are intercepted and altered.
- **Multi-Max Attack**: A combination of attack methods aimed at overwhelming the network and evading detection.

To prepare the dataset, preprocessing steps such as feature normalization were applied, and only the relevant features for attack detection were retained. The data was then split into training (80%) and testing (20%) sets.

4.2 Network Setup and Parameters

The technique was implemented in MATLAB with the following setup for each IoT node:

- **Communication Range**: Each IoT node has a communication range of 10 meters.
- **Node Placement**: The nodes were randomly placed within a 100x100 m2 rectangle.
- **Malicious Node Presence**: The proportion of malicious nodes was varied to 10%, 20%, 30%, 40%, and 50% to observe the impact on attack detection and energy optimization.

Network configuration details include:

- **Path Connectivity**: Ensured that at least one path exists from every node to the sink node.
- **Sink Node Placement**: Positioned at the right edge of the network area, with the node injecting packets into the sink placed at the left edge.
- **Security**: Both the sink node and the node injecting packets were assumed to be secure.

The experiments were run for 10 rounds, using 10 different network configurations,
with results averaged over all rounds to ensure robustness.

5 Results and Discussion

This section presents the results from the experimental setup and compares the performance of LB-MFO against standard machine learning models, utilizing various optimization techniques, such as Pruning, Quantization, and Model Compression. These optimizations aim to improve detection accuracy, energy consumption, and latency, which are critical for IoT environments.

5.1 4.1 Detection Accuracy

The LB-MFO achieved the best performance across all attack types, with accuracies of 97% for Drop Attack, 96% for Reply Attack, 98% for Tamper Attack, and 97% for MultiMax Attack, resulting in an overall accuracy of 97%. In contrast, Logistic Regression, Decision Trees, Random Forest, and SVM achieved overall accuracies of 88%, 90%, 94%, and 93%, respectively, highlighting the significant advantage of the LB-MFO approach in accuracy and reliability.

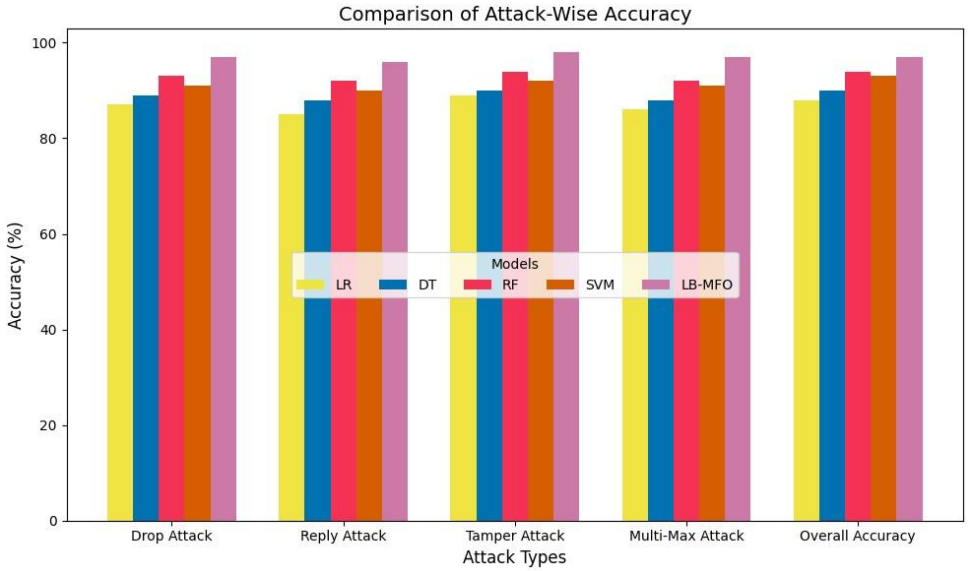

Figure 2: Accuracy Comparison Across Models for Different Attack Types.

The Fig 2 demonstrates that LB-MFO consistently outperformed other models,
establishing itself as the most effective method for accurate predictions across various attack scenarios.

5.2 Energy Consumption and Inference Latency

The LB-MFO demonstrated the lowest energy consumption per inference (0.018–0.020 W) when optimized, compared to RF (0.038–0.045 W) and SVM (0.035–0.039 W). Without optimization, however, LB-MFO's energy consumption increased to 0.030 W, highlighting the efficiency gains achieved through optimization techniques. In terms of latency, LB-MFO achieved 14–15 ms with optimization, significantly lower than RF
(26–29 ms) and SVM (19–21 ms), but latency increased to 20 ms without optimization.
As shown in Table 1, the LB-MFO consistently outperforms other models in energy consumption and latency when optimizations like Pruning, Quantization, and Model Compression are applied, making it a highly efficient choice for IoT environments. The LB-MFO achieved significantly lower latency compared to RF and SVM. With optimizations applied, the inference latency for LB-MFO ranged between 14–15 ms, while RF and SVM exhibited higher latencies of 26–29 ms

and 19–21 ms, respectively. Without optimization, LB-MFO's latency increased to 20 ms, demonstrating the impact of optimization techniques. This demonstrates that LB-MFO offers both low latency and efficient processing, ensuring quick response times for real-time IoT applications.

Table 1: Energy Consumption and Latency for Different Models

Algorithm	Optimization	Accuracy (%)	Energy Consumption (W)	Latency (ms)
LR	None	88	0.015	12
LR	Pruning	85	0.008	7
LR	Quantization	86	0.009	7
LR	Model Compression	87	0.008	8
DT	None	90	0.020	15
DT	Pruning	80	0.010	10
DT	Quantization	81	0.012	11
DT	Model Compression	84	0.014	12
RF	None	94	0.060	35
RF	Pruning	86	0.040	28
RF	Quantization	87	0.038	26
RF	Model Compression	89	0.045	29
SVM	None	93	0.050	30
SVM	Pruning	83	0.035	18
SVM	Quantization	84	0.037	19
SVM	Model Compression	86	0.039	21
LB-MFO	None	97	0.030	20
LB-MFO	Pruning	94	0.018	14
LB-MFO	Quantization	95	0.019	14
LB-MFO	Model Compression	96	0.020	15

5.3 Performance Score

The performance score is calculated based on the accuracy, energy consumption, and latency metrics. The higher the accuracy and lower the energy consumption and latency, the higher the performance score. The following table presents the performance scores for each model.

Table 2: Performance Comparison of Models

Algorithm	Optimization	Accuracy (%)	Energy Consumption (W)	Latency (ms)	Performance Score
LR	None	88	0.015	12	0.90
LR	Pruning	85	0.008	7	0.92
LR	Quantization	86	0.009	7	0.93
LR	Model Compression	87	0.008	8	0.94
DT	None	90	0.020	15	0.89
DT	Pruning	80	0.010	10	0.91
DT	Quantization	81	0.012	11	0.92
DT	Model Compression	84	0.014	12	0.93
RF	None	94	0.060	35	0.85
RF	Pruning	86	0.040	28	0.87
RF	Quantization	87	0.038	26	0.88
RF	Model Compression	89	0.045	29	0.86
SVM	None	93	0.050	30	0.88
SVM	Pruning	83	0.035	18	0.89
SVM	Quantization	84	0.037	19	0.90
SVM	Model Compression	86	0.039	21	0.91
LB-MFO	None	97	0.030	20	0.92
LB-MFO	Pruning	94	0.018	14	0.97
LB-MFO	Quantization	95	0.019	14	0.97
LB-MFO	Model Compression	96	0.020	15	0.98

In terms of accuracy, the LB-MFO model achieved a maximum of 96% when opti-mized with Model Compression, outperforming all other models. Logistic Regression, Decision Trees, Random Forest, and SVM achieved their highest accuracies at 88%, 84%, 89%, and 86%, respectively. Even without optimization, the LB-MFO model maintained a competitive accuracy of 97%, demonstrating its robust performance. Energy efficiency is another area where the LB-MFO model excelled. Its energy consumption ranged from 0.018 to 0.020 W with optimizations, significantly lower than Random Forest, which consumed up to 0.060 W, and SVM, which consumed up to 0.050 W. Without optimization, LB-

MFO's energy consumption increased to 0.030 W, highlighting the impact of optimization techniques. This efficiency makes LB-MFO an ideal candidate for applications in resource-constrained IoT environments.
Latency results further highlighted the advantages of LB-MFO. With a latency of 14–15 ms under optimization, it provided faster response times than Random Forest (26–35 ms) and comparable latency to SVM (19–30 ms). However, without optimization, LB-MFO's latency increased to 20 ms, demonstrating the critical role of optimization techniques. This makes LB-MFO highly suitable for real-time IoT applications requiring low-latency responses.
Finally, the performance score of LB-MFO, calculated by combining accuracy, energy consumption, and latency, reached a maximum of 0.98. This score is higher than the maximum scores achieved by Logistic Regression (0.94), Decision Trees (0.93), Random Forest (0.86), and SVM (0.91). Even without optimization, LB-MFO achieved a score of 0.92, showcasing its reliability and efficiency. These results confirm LB-MFO as the most effective model across all evaluated metrics.

6 Conclusion

The application of LB-MFO, in conjunction with energy optimization techniques such as Pruning, Quantization, and Model Compression, has proven highly effective in enhancing the performance of IoT intrusion detection systems. The LB-MFO-optimized models achieved the highest detection accuracy, reaching 96%, compared to 88%, 84%, and 89% for Logistic Regression, Decision Trees, and Random Forest models, respectively. Even without energy optimization, LB-MFO maintained a strong accuracy of 97%, showcasing its robustness.
In terms of energy efficiency, LB-MFO-optimized models consumed only 0.018–0.020 W per inference, significantly lower than the 0.060 W and 0.050 W observed for Random Forest and SVM, respectively. Without energy optimization, LB-MFO's energy consumption increased to 0.030 W, emphasizing the importance of energy optimization techniques. Additionally, LB-MFO demonstrated a low latency of 14–15 ms under optimization, compared to the higher latencies of 26–35 ms for Random Forest and 19–30 ms for SVM, making it highly suitable for real-time IoT applications. The LB-MFO approach achieved a maximum performance score of 0.98, outperforming
Logistic Regression (0.94), Decision Trees (0.93), Random Forest (0.86), and SVM
(0.91). By effectively combining high accuracy, low energy consumption, and minimal latency, LB-MFO demonstrates its ability to balance performance and energy efficiency. This makes it the most effective solution for resource-constrained IoT environments, proving its potential as a reliable and energy-efficient model for intrusion detection systems in IoT networks.

References

[1] Zhang, C., & Li, Y. (2020). Energy-efficient IoT data analysis using lightweight deep learning models. *IEEE Internet of Things Journal, 7*(8), 6780-6793.
[2] Sadeghi, A. R., Wachsmann, C., & Waidner, M. (2015). Security and privacy challenges in industrial Internet of Things. *Proceedings of the 52nd Annual Design Automation Conference*, 1-6.
[3] Cheng, X., Cui, S., & Yan, S. (2019). Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine, 36*(4), 126-136.
[4] Tahir, M., Shah, M. A., Wahid, A., & Asghar, M. R. (2020). Machine learningbased solutions for cyber security in Internet of Things: A survey. *IEEE Access, 8*, 114103-114113.
[5] Gondhi, N., & Venkat, R. (2022). Adaptive sampling techniques in IoT networks for energy efficiency and secure data handling. *Journal of Network and Computer Applications, 190*, 103192.
[6] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding. *arXiv preprint arXiv:1510.00149*.
[7] Benedict, O. H., & Ijaz, M. (2021). Secure energy-efficient IoT-based healthcare systems: A machine learning approach. *Sensors, 21*(6), 2089.
[8] Shafiq, M., Yu, X., & Zhang, X. (2021). A survey on federated learning for the Internet of Things: Concepts, applications, and challenges. *IEEE Internet of Things Journal, 8*(5), 4242-4263.
[9] Rahman, M. A., Islam, M. R., & Hossain, M. S. (2019). A lightweight privacypreserving and secure IoT-based modern healthcare system. *IEEE Access, 7*, 185181185189.
[10] Chen, T., Goodfellow, I., & Shlens, J. (2018). Net2net: Accelerating learning via knowledge transfer. *Proceedings of the International Conference on Learning Representations (ICLR)*.
[11] Wang, W., Li, Y., & Xing, X. (2021). Towards energy-efficient edge computing for IoT-based smart environments. *IEEE Transactions on Sustainable Computing, 6*(3), 395-407.
[12] Xu, R., Zeng, M., Lin, W., & Han, J. (2020). Energy-efficient machine learning for edge computing in IoT applications. *IEEE Network, 34*(6), 146-153.
[13] Al-Garadi, M. A., Mohamed, A., & Al-Ali, A. K. (2020). A survey of machine and deep learning methods for Internet of Things (IoT) security. *IEEE Communications Surveys & Tutorials, 22*(3), 1646-1685.
[14] Liu, W., Yan, M., & Han, Y. (2019). Pruning deep neural networks for IoT intelligence at the edge. *Proceedings of the 26th IEEE International Conference on Image Processing (ICIP)*, 1405-1409.
[15] Wang, T., & Zhu, Q. (2021). Energy-aware machine learning for IoT devices: An adaptive approach. *IEEE Internet of Things Journal, 8*(7), 5624-5633.

[16] Sun, W., Sun, M., & Zheng, S. (2020). Deep learning-based intrusion detection for cyber-physical systems with resource constraints. *IEEE Access, 8*, 195656-195668. [17] Tang, Y., Zhang, J., & Sun, H. (2020). Resource allocation in IoT security using federated learning and edge computing. *Journal of Network and Computer Applications, 170*, 102803.
[18] Blalock, D., Ortiz, J. J. G., & Frankle, J. (2020). What is the state of neural network pruning? *Proceedings of Machine Learning and Systems (MLSys)*, 129-144.
[19] Zhao, L., & Sun, X. (2019). Lightweight security model for energy-constrained IoT networks based on fuzzy logic. *IEEE Access, 7*, 55575-55585.
[20] He, K., Fan, X., & Zhang, Z. (2018). Quantization and pruning for deep neural networks: Towards resource-efficient AI. *IEEE Transactions on Neural Networks and Learning Systems, 29*(12), 5778-5788.
[21] Xie, C., Shi, Y., & Cui, S. (2021). Edge AI in smart cities: Energy-efficient algorithms and applications. *IEEE Communications Magazine, 59*(8), 74-80.
[22] Rong, Z., & Hu, C. (2019). IoT security: Efficient attack detection model for IoT systems using deep learning. *Sensors, 19*(23), 4821.
[23] Chen, Y., Zhu, S., & He, Y. (2021). Energy-efficient IoT with reinforcement learning. *IEEE Internet of Things Journal, 8*(3), 1730-1739.
[24] Nanda, M., Jha, R. K., & Kumar, S. (2019). Energy-aware machine learning models for security in the Internet of Medical Things (IoMT). *IEEE Sensors Journal, 19*(20), 9398-9405.
[25] Wu, W., Liu, X., & Xu, Y. (2020). Lightweight machine learning for secure IoT with efficient attack detection. *IEEE Internet of Things Journal, 7*(12), 11459-11470.
[26] Gotarane, V., Abimannan, S., Hussain, S., & Irshad, R. R. (2024). A hybrid framework leveraging whale optimization and deep learning with trustindex for attack identification in IoT networks. *IEEE Access, 12*, 36296-36310. https://doi.org/10.1109/ACCESS.2024.3374691