Open Access ISSN: 3007-1437 (Online), ISSN: 3007-1429 (Print)

IRAQI Academic Scientific Journals

Alkadhim Journal for Computer Science (KJCS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

KJCS ALKADHIM JOURNAL FOR COMPUTER SCIENCE

Developing an Enhanced IHBO Algorithm with Chaotic Binary Hashing to Optimize BiGRU for Industry 4.0 IIoT Anomaly Detection

[1]Saif Saad Alamshani*

[1]Cybersecurity Engineering Department , Engineering College , Al-Iraqia Science University , Baghdad , Iraq

Article information

Article history:

Received: November, 29, 2025

Accepted: March, 11, 2026

Available online: March, 25, 2026

Keywords:

Anomaly Detection

Industrial Internet of Things (IIoT)

Industry 4.0

Intrusion Detection System

BiGRU

Improved Honey Badger Optimization (IHBO)

Logistic Chaotic Map

Binary Hashing

Feature Selection

Corresponding Author:

Saif Saad Alamshani

saifalamshani@baghdadcollege.edu.iq

https://orcid.org/0009-0003-5463-6514

DOI:

https://doi.org 10.61710/kjcs.v4i1.138

Abstract

The rapid expansion of the Industrial Internet of Things (IIoT) in Industry 4.0 has increased the need for reliable anomaly detection to protect industrial services and critical operations. Deep learning–based IDS solutions can achieve strong performance, yet practical deployment may be affected by tuning effort and suboptimal search behavior when optimization is trapped in local optima, particularly when feature selection and model optimization are handled as separate stages. This paper develops an enhanced Improved Honey Badger Optimization (IHBO) algorithm and integrates it with a Bidirectional Gated Recurrent Unit (BiGRU) model for IIoT anomaly detection. To strengthen exploration and maintain population diversity during optimization, the proposed IHBO incorporates a logistic chaotic map as a controlled diversification mechanism. In addition, chaotic binary hashing is used to map continuous candidate representations into discrete binary decisions, enabling effective wrapper-based feature selection within the same optimization framework. The main novelty lies in performing integrated feature selection and BiGRU optimization within a single framework rather than treating these steps independently. Experiments on two benchmark datasets—an industrial IIoT dataset and UNSW-NB15—show that the proposed BiGRU–IHBO approach achieves 94.70% accuracy on the IIoT dataset and 99.29% accuracy on UNSW-NB15, with consistent improvements in precision, recall, and F1-score compared with baseline models reported in the same experimental setting.

1. INTRODUCTION

The Industrial Internet of Things (IIoT) has become a key enabler of Industry 4.0 by interconnecting sensors, actuators, and industrial control components to improve automation and operational efficiency. However, the same connectivity and heterogeneity that enable these benefits also introduce significant security risks, making reliable anomaly/intrusion detection an essential protection layer for modern industrial environments [7], [9]. In addition, the evolving nature of cyber threats and the operational constraints of industrial networks motivate IDS solutions that can maintain strong detection capability while remaining practical for industrial deployment [1], [4].

Intrusion Detection Systems (IDSs) are commonly categorized into signature-based and anomaly-based approaches. Signature-based IDSs are effective for known threats but become limited when attacks evolve or when previously unseen patterns appear; they also require ongoing updates to signature databases and can incur additional processing overhead as signatures grow [1], [4]. Anomaly-based IDSs aim to detect unknown or emerging behaviors, but their performance can be affected by dynamic industrial conditions and heterogeneous traffic, which may increase false alarms if the model is not robustly designed [1], [4].

To address these limitations, data-driven IDS approaches based on machine learning and deep learning have been widely investigated. Recent surveys emphasize that deep models can learn discriminative representations from complex traffic patterns, yet performance may remain sensitive to data characteristics and configuration choices, especially in high-dimensional and noisy environments [9]. In IIoT scenarios, recurrent models (RNN/LSTM/GRU variants) are particularly relevant because many attack behaviors manifest as temporal deviations that unfold across sequences rather than isolated records [18], [19]. BiGRU models are attractive for such tasks because they can capture temporal dependencies in both forward and backward directions, improving context modeling for sequential IIoT traffic [10].

Despite their promise, two recurring issues can reduce the practicality of deep-learning IDS pipelines. First, industrial datasets may contain redundant or irrelevant attributes that increase training cost and reduce generalization, motivating feature selection and dimensionality reduction [11], [8], [20]. Second, many IDS pipelines treat feature selection and model optimization as separate stages, which may limit the interaction between the chosen feature subset and the optimized classifier configuration and may increase the tuning burden [9].

Metaheuristic optimization has therefore been adopted to automate search in large, non-convex design spaces. The Honey Badger Algorithm (HBA) is a recent population-based optimizer that balances exploration and exploitation and has been used in challenging optimization problems [12]. Nevertheless, metaheuristics can still be susceptible to local optima and search instability, particularly when the optimization includes both continuous parameters and binary feature-selection decisions. These challenges motivate enhanced variants that strengthen exploration and improve search stability.

To validate the approach, we evaluate on two datasets: UNSW-NB15, a widely used intrusion detection benchmark [22], and an industrial IIoT dataset used to evaluate Industry 4.0 intrusion/anomaly detection scenarios (see Section 4.1) [5].

The main contributions of this work are as follows:

1. Integrated optimization framework: a unified BiGRU–IHBO pipeline that performs feature selection and BiGRU optimization within a single intelligent optimization process.
2. Enhanced IHBO design: incorporating logistic chaotic mapping and chaotic binary hashing to strengthen exploration and support effective binary feature selection.
3. Empirical validation: evaluation on UNSW-NB15 and an industrial IIoT dataset, demonstrating improvements over baseline models reported in the same experimental setting.

The remainder of this paper is organized as follows: Section 2 reviews related work and positions the research gap. Section 3 details the proposed BiGRU–IHBO method. Section 4 presents the experimental setup and results. Section 5 concludes the paper and outlines future directions.

2. RELATED WORK

2.1 IDS in IIoT and Industry 4.0: main directions

Recent research on intrusion/anomaly detection for IIoT emphasizes the need for methods that can handle heterogeneous industrial traffic, evolving attack behaviors, and practical deployment constraints. Systematic reviews note that learning-based IDS pipelines are increasingly adopted in IIoT, but their effectiveness depends on robust representation learning and optimization strategies that can generalize across changing industrial conditions [1], [4].

2.2 Learning-based IDS with classical ML and ensemble models

A large stream of IIoT IDS work applies classical machine learning and ensemble learning to engineered traffic features. For example, ensemble-based approaches for IIoT edge computing have been reported as effective baselines, especially when carefully configured for local environments [3]. Hybrid ML pipelines combining multiple learners (e.g., tree-based models and boosting) also appear frequently as practical alternatives when computational simplicity is prioritized [15]. However, these approaches often remain sensitive to feature redundancy and dataset-dependent characteristics, and they typically do not model temporal dependencies explicitly—an issue when attack behavior unfolds as sequences rather than isolated records.

2.3 Deep learning for IIoT IDS and sequence modeling

Deep learning methods have been widely explored to reduce reliance on handcrafted features and to learn discriminative patterns from complex traffic. Several studies report improved detection capability in IIoT contexts using deep architectures [18], [19]. In industrial settings, deep IDS approaches are also deployed within fog/edge environments to keep detection closer to data sources and reduce response latency [2]. Representation-learning methods such as optimized Autoencoder have also been used for IoT intrusion detection as part of feature learning or dimensionality reduction pipelines [13]. However, deep IDS pipelines can still require careful configuration and repeated tuning to achieve stable results across datasets and deployment conditions, which can limit practicality when frequent updates or cross-domain generalization is needed.

2.4 Feature selection, high dimensionality, and imbalance-aware learning

Feature selection remains central in IDS pipelines because network datasets can contain redundant or irrelevant attributes that inflate computational cost and harm generalization. Surveys confirm that feature selection choices can substantially affect IDS performance and efficiency [11]. Another recurring issue is class imbalance, which can inflate headline accuracy while masking reduced performance on minority attack classes. To mitigate imbalance, IoT IDS research explores augmentation and sampling strategies; for example, data augmentation is studied to enhance learning on unbalanced samples [14], and SMOTE variants are evaluated for cyber-attack prediction in IoT networks [16]. Nevertheless, many studies still provide limited class-level analysis or insufficiently connect imbalance considerations with the pipeline design, making it difficult to interpret practical detection reliability.

2.5 Metaheuristic-optimized IDS and Honey Badger Optimization

To reduce manual tuning and improve robustness, IIoT IDS research increasingly integrates metaheuristic optimization, particularly for wrapper-based feature selection and configuration search. HBA is introduced as a population-based optimizer designed to balance exploration and exploitation and has been applied across optimization tasks [12]. Beyond HBA, IDS literature also explores other optimizers and their improved variants within hybrid IDS frameworks, reflecting a trend toward optimization-assisted learning pipelines [17], [21]. However, two challenges remain common: (i) optimization may still become trapped in local optima without explicit diversity reinforcement, and (ii) many pipelines optimize feature selection and model tuning separately without clearly explaining their operational interaction or how binary feature decisions are handled inside the optimizer.

2.6 Gap analysis and positioning of this work

Based on the above literature, key gaps motivating this study are:

1. Weak operational integration: many IDS frameworks treat feature selection and model optimization as sequential, loosely connected steps rather than a unified process [11], [12].
2. Local-optima sensitivity: optimizer diversity is not always explicitly reinforced, increasing the risk of premature

convergence in complex IDS search landscapes [12].

3. Binary decision handling: feature selection is inherently binary, yet binary mapping (stable 0/1 mask generation) is often under-specified, affecting stability and reproducibility [11].
4. Evaluation realism: benchmarks such as UNSW-NB15 are central for IDS evaluation [22], but imbalance and class-level behavior must be addressed explicitly to avoid misleading conclusions [14], [16].

Positioning: To address these gaps, this work develops an enhanced IHBO mechanism (built on HBA) and integrates it with a BiGRU-based detector in a unified framework that supports (i) wrapper feature selection and (ii) BiGRU optimization within one coordinated search process, while strengthening diversity via logistic chaotic mapping and enabling binary feature decisions via chaotic binary hashing.

3. PROPOSED METHODOLOGY

3.1 Overview of the BiGRU–IHBO framework

This paper proposes an intrusion/anomaly detection framework for Industry 4.0 IIoT data that combines a deep BiGRU classifier with an enhanced IHBO strategy. The key design choice is that the same optimization engine is used for (i) selecting an informative subset of input features and (ii) optimizing the BiGRU model weights, rather than treating these steps as two loosely connected stages. We adopt HBA as the optimizer foundation [12], and BiGRU as a sequential classifier suitable for temporal patterns in IIoT traffic [10]. The overall architecture and operational flow of the proposed framework are illustrated in Figure 1.

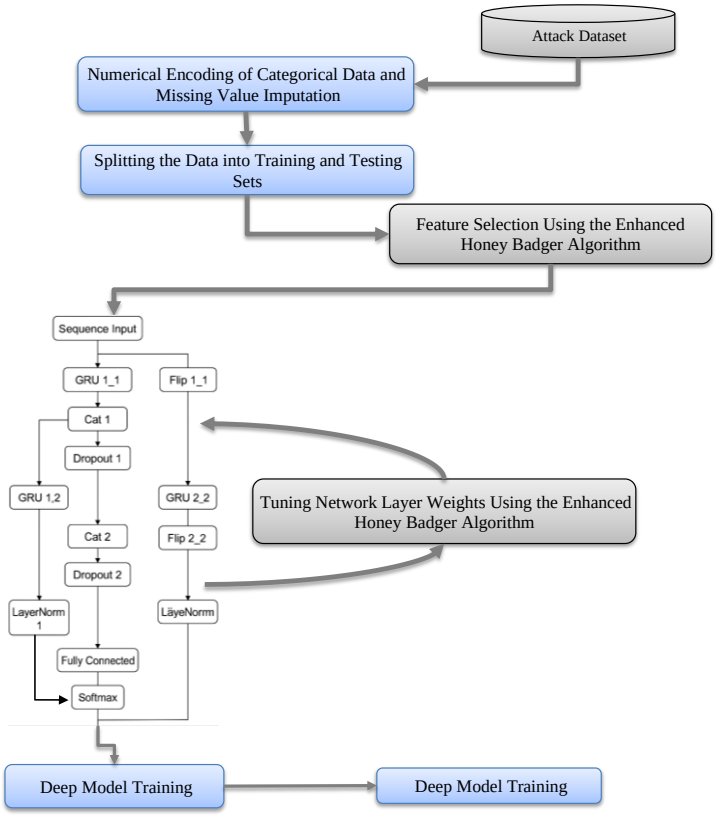

Figure. 1: The overall architecture of the proposed BiGRU-IHBO framework.

3.2 Data preparation: categorical attributes and missing values

IIoT traffic records may contain categorical fields that are not directly suitable for learning algorithms. Therefore, categorical attributes are converted into numeric representations using a one-hot index–based encoding method to preserve discriminative information while maintaining stable model performance. To address incomplete records, missing values are handled via KNN imputation **using** Euclidean distance, where each missing entry is estimated from its nearest neighbors in the feature space. This step improves data consistency and reduces the negative impact of messiness on detection performance.

3.3 Wrapper feature selection using enhanced IHBO

Feature selection is implemented as a wrapper process driven by an improved HBA variant. The optimizer iteratively searches for a feature subset that minimizes a classification error computed using a KNN classifier (details in Section 3.3.3). HBA is selected because it balances exploration and exploitation through digging and honey-search behaviors [12]. The proposed enhancement focuses on improving exploration stability and reducing local-optima trapping in metaheuristic search.

3.3.1 Logistic chaotic mapping for diversity control

To enhance exploration and avoid premature convergence, the chaotic value CH is computed using Eq. (1). In this work, the logistic chaotic map and its parameter settings are summarized in Table 1. The chaotic sequence is initialized with value 0.1, and chaotic values are regenerated at each optimization iteration to continuously perturb candidate solutions and maintain diversity.

$$\overrightarrow{CH} = logistic_{map}(Initial_{value}, iteration) \quad (1)$$

where CH denotes the chaotic value used to perturb candidate solutions during optimization, and all remaining terms follow the thesis definitions.

Table1 : Logistic Chaotic Map

Chaotic map method	Formulas
Logistic map	$x_{i+1} = ax_i(1 - x_i), a = 4$

3.3.2 Chaotic binary hashing for targeted candidate updates

Based on the chaotic sequence, a binary hash code is constructed using Eq. (2) by mapping each chaotic element to +1 if it is greater than or equal to zero and to −1 otherwise. This converts continuous chaotic values into simple discrete decisions.

$$Binary_{hashcode} = \begin{cases} 1 & x_{chaos} \geq 0 \\ -1 & otherwise \end{cases} \quad (2)$$

Operationally, the hash signs are then used to update candidate feature vectors: components corresponding to positive hash entries are increased according to the chaotic value, while components corresponding to negative entries are decreased. This position update mechanism is mathematically expressed in Eq. (3).

$$X_{new} = \begin{cases} \vec{x}_{new} + \overrightarrow{CH} & Binary_{hashcode} > 0 \\ \vec{x}_{new} - \overrightarrow{CH} & otherwise \end{cases} \quad (3)$$

Values exceeding the upper bound (ub) or lower bound (lb) are projected back into the allowed range $[lb, ub]$.

3.3.3 Binarisation and wrapper evaluation with KNN error

The feature-selection population is generated as 25 candidate solutions within the interval [−1, 1] and then binaries using Eq. (4) to produce an index matrix encoding selected features:

$$X_{index} = \begin{cases} 1 & X > 0 \\ 0 & otherwise \end{cases} \quad (4)$$

The KNN-based accuracy used for evaluating each candidate subset is computed using Eq. (5).

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} \delta(y_i, \hat{y}_i) \quad (5)$$

Finally, the cost (fitness error) is computed as the complement of accuracy using Eq. (6)

$$Error(fitness) = 1 - Accuracy \quad (6)$$

KNN is selected because it has no explicit training phase, providing computational efficiency for repeated wrapper evaluations. The optimizer then updates the population iteratively using these error values until a feature subset with minimal KNN error is obtained. (Feature selection background and motivation are aligned with common FS practice [11].)

3.4 BiGRU classifier and weight optimization using HBA/IHBO

After feature selection, the final detection component is a BiGRU-based network whose parameters are optimized using HBA. The architecture consists of a sequence input layer followed by two parallel GRU processing paths (forward and backward) implemented with a Flip Layer structure. The outputs are merged and passed through dropout to mitigate over-fitting and normalization to stabilize training, then fed to fully connected layers followed by softmax and a classification layer.

For optimization, the GRU and fully connected layer weights are initially generated randomly, then HBA refines these weights by minimizing cross-entropy loss. Each population member represents a candidate weight configuration updated by HBA movement rules and evaluated via the loss function. The reported configuration uses population size 25 and 30 optimization iterations for the weight-optimization stage.

3.5 Operational interaction and component-wise comparison (as requested by reviewers)

Compared with common IDS pipelines where feature selection and model tuning are handled as separate sequential steps, the proposed framework integrates both stages under a single optimization logic. First, logistic chaotic mapping maintains population diversity and strengthens exploration, while chaotic binary hashing enables stable discrete decisions for wrapper feature selection. Second, instead of relying only on gradient-driven updates, the framework uses HBA to refine BiGRU weights under a cross-entropy objective, exploring candidate configurations across the population.

4. EVALUATION AND RESULTS

4.1 Datasets

Two datasets were used to evaluate the proposed BiGRU–IHBO framework:

1. UNSW-NB15: a widely used benchmark dataset for intrusion detection, containing more than 700,000 records, 47 features, and 9 attack categories in addition to normal traffic [22].
2. Industrial IIoT dataset: an Industry 4.0–oriented dataset containing 133 features and diverse attack patterns, used to evaluate industrial intrusion detection scenarios.[5]

Imbalance note (important for interpretation): In IIoT intrusion detection, datasets are often severely imbalanced, where normal samples dominate attack samples. In such cases, a model can achieve high Accuracy mainly by predicting the majority “normal” class, while Precision/Recall remain lower because they are sensitive to correct detection of minority attack samples [14], [16]. No resampling (e.g., SMOTE/under-sampling) is applied in this work; therefore, F1-score and confusion-matrix analysis are emphasized to reflect minority-class behavior.

4.2 Experimental configuration

All experiments were run on a machine with 9 CPU cores, 32 GB RAM, and an NVIDIA RTX 3080 GPU. The data were split into 80% training and 20% testing. The learning setup used the Adam optimizer with an initial learning rate of 0.01, 30 epochs, and a mini-batch size of 3000 samples. For the Honey Badger optimization settings, the population size was 25. The number of iterations was set to 100 for the feature-selection phase, and 30 for optimizing the network weights. As shown in Figure 2, the configuration (population = 25 with the logistic chaotic map) provided a strong trade-off between detection performance and computational cost and was adopted for the reported experiments. Because metaheuristic tuning requires repeated evaluations, the optimization is intended for offline training, while deployment-time inference remains a standard BiGRU forward pass.

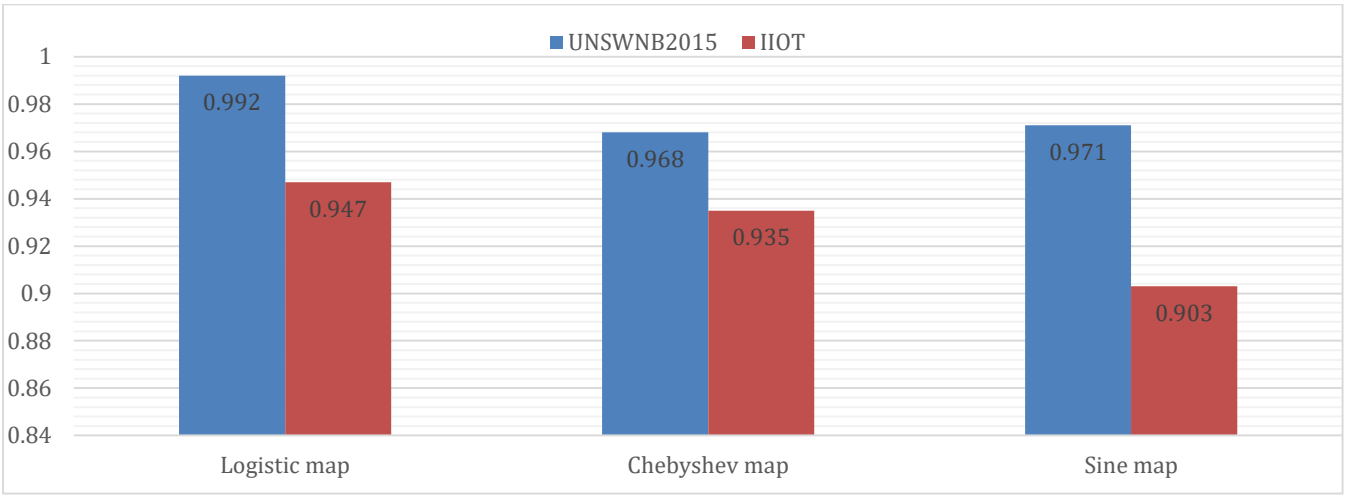

Figure 1 : Performance comparison of different chaotic maps combined with the Honey Badger algorithm.

4.3 Evaluation metrics

Performance was assessed using four standard classification metrics—Accuracy, Precision, Recall, and F1-score—as defined in Eqs. (7)–(10). The metrics are calculated based on True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$F_{score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

4.4 Results on UNSW-NB15

Table 2 summarizes the comparison on UNSW-NB15 between the proposed BiGRU–IHBO model and the baseline MH-DRNN method [6]. The proposed model achieves higher results across all metrics, with notable gains in Precision and F1-score. These improvements are also depicted in Figure 3.

Table 2 : Performance on UNSW-NB15 (baseline MH-DRNN [6])

Criterion	MH-DRNN	Proposed method (BiGRU-IHBO)	Difference
Precision	48.73%	70.28%	+21.55% better
Recall	45.15%	54.79%	+9.64% better
F1-score	46.87%	61.58%	+14.71% better
Accuracy	98.4%	99.29%	+0.89% better

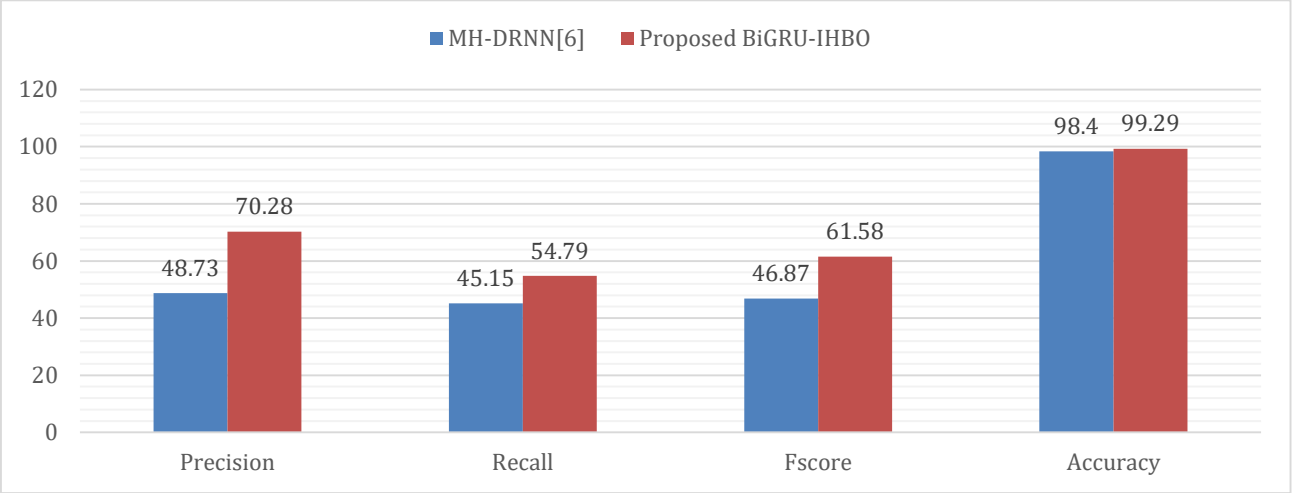

Figure 2 : Performance comparison of the proposed method with baseline models on the UNSW-NB15 dataset.

Imbalance interpretation: Although Accuracy is very high, Precision and Recall are lower. This behavior is consistent with imbalanced intrusion datasets, where correct prediction of many normal samples can inflate Accuracy even if attack detection is harder [14], [16]. The confusion matrix in Figure 4 further illustrates class-level behavior and supports a more realistic interpretation beyond Accuracy alone.

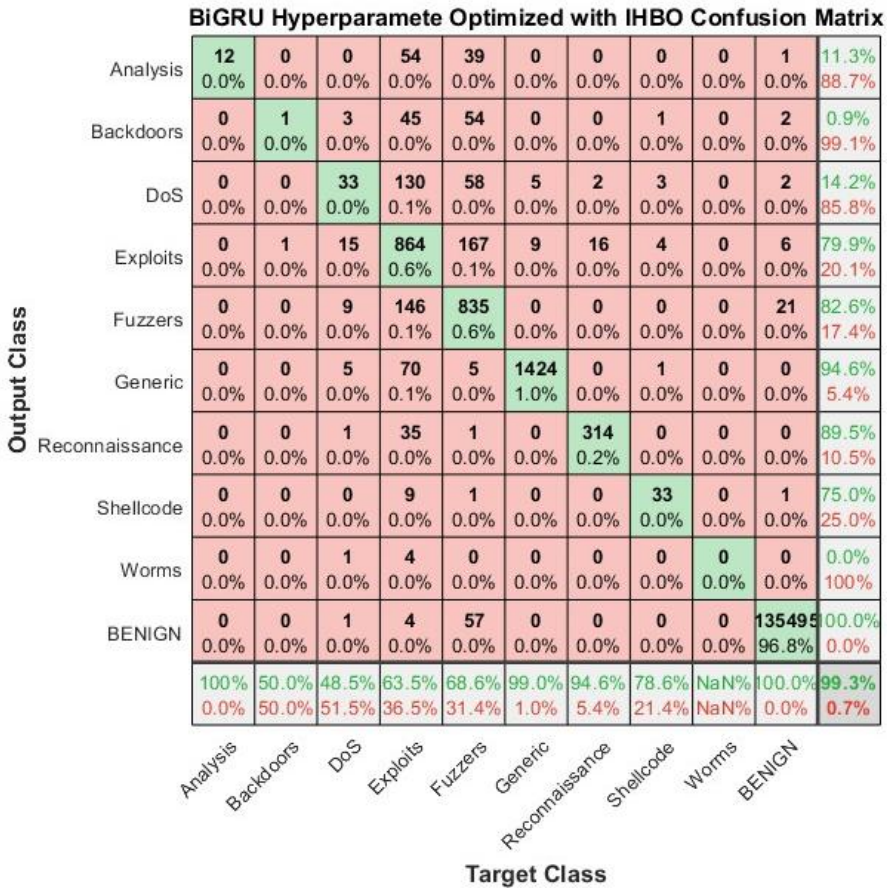

Figure 3 : Confusion matrix of the proposed BiGRU-IHBO model on the UNSW-NB15 dataset.

4.5 Results on the Industrial IIoT dataset

On the industrial IIoT dataset, the proposed model was compared with a baseline ANN approach [5]. Table 3 reports that the proposed BiGRU–IHBO method outperforms the ANN baseline across all metrics. The convergence curve in Figure 5 indicates stable convergence behavior during the BiGRU weight optimization phase.

Table 3 : Performance on the Industrial IIoT dataset (baseline ANN [5])

Criterion	ANN [5]	Proposed method (BiGRU-IHBO)	Difference
Precision	92.10%	95.74%	+3.64% better
Recall	92.30%	96.46%	+4.16% better
F1-score	92.20%	96.10%	+3.90% better
Accuracy	91.50%	94.70%	+3.20% better

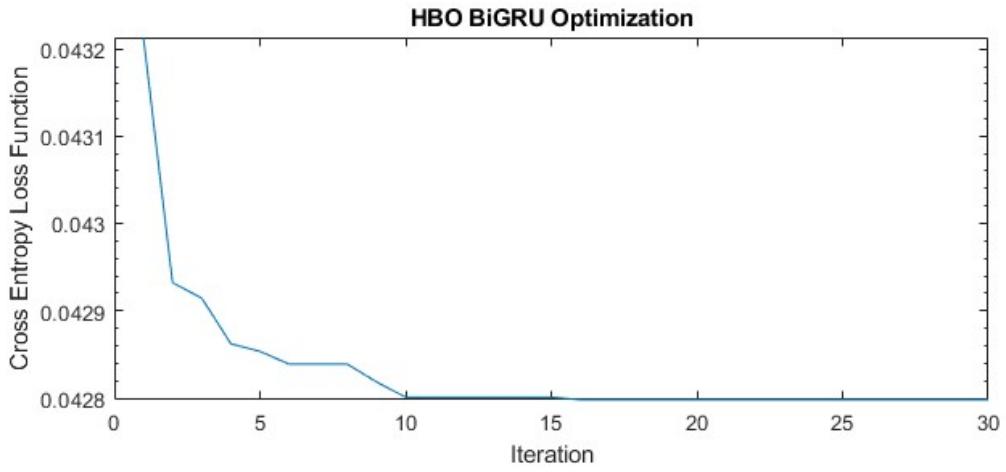

Figure 4 : Convergence curve of the proposed IHBO algorithm during the weight optimization phase.

5. CONCLUSION AND FUTURE WORK

This paper presented an optimized anomaly detection framework for Industry 4.0 IIoT environments by integrating a BiGRU classifier with an enhanced IHBO algorithm. The proposed approach strengthens the optimization process using logistic chaotic mapping to maintain population diversity and chaotic binary hashing to support effective binary feature selection, while also optimizing BiGRU weights within a unified framework [12].

Experimental evaluation on UNSW-NB15 [22] and an industrial IIoT dataset [5] demonstrated that the proposed BiGRU–IHBO model achieves strong performance and improves precision, recall, F1-score, and accuracy compared with baseline methods reported in the same experimental setting [5], [6]. In response to reviewer concerns, it is important to emphasize that very high accuracy can coexist with moderate precision/recall differences in imbalanced datasets; therefore, F1-score and class-sensitive analysis (e.g., confusion matrices) are critical for realistic IDS interpretation [14], [16].

Future work will extend evaluation with deeper class-level reporting and investigate additional imbalance-handling strategies to further improve minority attack detection. Another important direction is to further analyze and reduce the computational overhead introduced by metaheuristic tuning, including efficiency improvements such as reduced evaluation strategies or hybrid training schemes to maintain strong detection performance while improving feasibility for edge/fog IIoT deployments.

REFERENCES

[1] T. Vaiyapuri, Z. Sbai, H. Alaskar, and N. A. Alaseem, "Deep learning approaches for intrusion detection in IIoT networks–opportunities and future directions," International Journal of Advanced Computer Science and Applications, vol. 12, no. 4, 2021.

[2] M. Abdel-Basset, V. Chang, H. Hawash, R. K. Chakrabortty, and M. Ryan, "Deep-IFS: Intrusion detection approach for industrial internet of things traffic in fog environment," IEEE Transactions on Industrial Informatics, vol. 17, no. 11, pp. 7704-7715, 2020.

[3] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, and M. Azrour, "An effective intrusion detection approach based on ensemble learning for IIoT edge computing," Journal of Computer Virology and Hacking Techniques, vol. 19, no. 4, pp. 469-481, 2023.

[4] M. Nuaimi, L. C. Fourati, and B. B. Hamed, "Intelligent approaches toward intrusion detection systems for Industrial Internet of Things: A systematic comprehensive review," Journal of Network and Computer Applications, vol. 215, p. 103637, 2023.

[5] S. Alem, D. Espes, L. Nana, E. Martin, and F. De Lamotte, "A novel bi-anomaly-based intrusion detection system approach for industry 4.0," Future Generation Computer Systems, vol. 145, pp. 267-283, 2023.

[6] X. Li, C. Xie, Z. Zhao, C. Wang, and H. Yu, "Anomaly Detection Algorithm of Industrial Internet of Things Data Platform Based on Deep Learning," IEEE Transactions on Green Communications and Networking, 2024.

[7] N. Tuptuk and S. Hailes, "Security of smart manufacturing systems," Journal of Manufacturing Systems, vol. 47, 05/21 2018, doi: 10.1016/j.jmsy.2018.04.007.

[8] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection," Electronics, vol. 8, no. 3, p. 322, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/3/322.

[9] H. Liu and B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," Applied Sciences, vol. 9, no. 20, p. 4396, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/20/4396.

[10] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," Neural computation, vol. 31, no. 7, pp. 1235-1270, 2019.

[11] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Computers & Electrical Engineering, vol. 40, no. 1, pp. 16-28, 2014.

[12] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems," Mathematics and Computers in Simulation, vol. 192, pp. 84-110, 2022.

[13] B. Lahasan and H. Samma, "Optimized deep autoencoder model for Internet of Things intruder detection," IEEE Access, vol. 10, pp. 8434-8448, 2022.

[14] Y. Zhang and Q. Liu, "On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples," Future Generation Computer Systems, vol. 133, pp. 213-227, 2022/08/01/ 2022, doi: https://doi.org/10.1016/j.future.2022.03.007.

[15] J. A. Faysal et al., "XGB-RF: A Hybrid Machine Learning Approach for IoT Intrusion Detection," Telecom, vol. 3, no. 1, pp. 52-69, 2022. [Online]. Available: https://www.mdpi.com/2673-4001/3/1/3.

[16] B. S. Akash, P. K. R. Yannam, B. V. S. Ruthvik, L. Kumar, L. B. Murthy, and A. Krishna, "Predicting Cyber-Attacks on IoT Networks Using Deep-Learning and Different Variants of SMOTE," Cham, 2022: Springer International Publishing, in Advanced Information Networking and Applications, pp. 243-255.

[17] F. S. Alrayes et al., "Modeling of Botnet Detection Using Barnacles Mating Optimizer with Machine Learning Model for Internet of Things Environment," Electronics, vol. 11, no. 20, p. 3411, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/20/3411.

[18] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," Computer Communications, vol. 199, pp. 113-125, 2023.

[19] H. Nandanwar and R. Katarya, "Deep learning enabled intrusion detection system for Industrial IOT environment," Expert Systems with Applications, vol. 249, p. 123808, 2024.

[20] A. V. Turukmane and R. Devendiran, "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning," Computers & Security, vol. 137, p. 103587, 2024.

[21] M. H. Nadimi-Shahraki, H. Zamani, Z. Asghari Varzaneh, and S. Mirjalili, "A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations," Archives of Computational Methods in Engineering, vol. 30, no. 7, pp. 4113-4159, 2023.

[22] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, pp. 1–6, 2015.