Enhanced IIoT Security: A Hybrid Intrusion Detection Framework Using Signature-Based and Deep Learning Approaches

Komal Bansal
Department of Computer Science and Engineering
Deenbandhu Chhotu Ram University of Science and Technology, India
komal.bansal22@gmail.com

Anita Singhrova
Department of Computer Science and Engineering
Deenbandhu Chhotu Ram University of Science and Technology, India
nidhianita@gmail.com

Abstract: *With the rapid adoption of Industrial Internet of Things (IIoT) technologies, ensuring robust security against sophisticated cyberattacks has become a critical challenge. Traditional intrusion detection systems often rely solely on signature-based or anomaly-based methods, which limits their effectiveness in detecting both known and novel attacks. This study proposes a hybrid Industrial Intrusion Detection System (IIDS) that integrates the strengths of signature-based hashing and anomaly-based deep learning techniques. The system begins with signature-based detection using an enhanced Grøstl Hashing Algorithm (GHA) with right-shift rotation to quickly identify known attack patterns. For data that do not match existing signatures, the system employs an anomaly detection module, leveraging the Kullback-Leibler Divergence (KLD)based sailfish optimization algorithm for optimal feature selection. Classification is performed using a SoftSwish Gated Recurrent Unit (SSGRU), which enhances the learning of temporal dependencies and improves detection accuracy. The proposed system is evaluated on five benchmark datasets, and it demonstrated superior performance in terms of intrusion detection accuracy, false positive rates, and computational efficiency compared to standalone approaches. The findings confirm the efficiency of the hybrid IIDS in addressing the evolving security challenges in IIoT environments.*

Keywords: *Deep learning, machine learning, information security, anomaly detection, IIDS, IIoT, GRU, GHA, SOA.*

Received February 17, 2025; accepted October 1, 2025
https://doi.org/10.34028/iajit/23/2/5

1. Introduction

An Industrial Control System (ICS) serves as the digital brain for factories and machines, overseeing and managing operations through automation, sensors, and control mechanisms to ensure efficiency and coordination in industrial processes [13]. These systems regulate and coordinate the actions of various industrial components through digital control [8], often rely on interconnected hardware and software systems to manage complex operations in an increasingly digitized industrial environment. The ICS integrates hardware and software to control, monitor, and safeguard critical information, ensuring operational integrity and security [2]. However, this integration also exposes industrial environments to significant cybersecurity risks [4]. Unauthorized access, malicious attacks, and data breaches pose significant threats to information systems, compromising data integrity and security, which in turn disrupt operations, jeopardize safety, and negatively impact various stakeholders [37]. Therefore, the development of robust and effective Industrial Intrusion Detection Systems (IIDS) is essential for protecting these environments from cyber threats.

IIDS are critical for monitoring and securing networks and machinery in industrial environments, including the Industrial Internet of Things (IIoT). These systems detect suspicious activities or cyber-attacks by continuously analyzing the network traffic and system behavior [29]. To achieve this, IIDS relies on three primary detection methods: signature-based, anomaly-based, and hybrid approaches.

1. Signature based detection: this method identifies attacks by matching incoming data with a database of known attack patterns. It is highly efficient and faster than other methods [32] because of its simple comparison process, making it ideal for detecting previously identified threats.
2. Anomaly based detection: this technique establishes a baseline for normal behavior and monitors deviations [27]. It is effective in detecting new or zero-day attacks [1].
3. Hybrid methods: hybrid methods leverage the strengths of both signature-and anomaly-based techniques. It provides a comprehensive solution capable of detecting known attacks with speed, while also identifying novel threats through anomaly detection [12].

Machine Learning (ML) and Deep Learning (DL) algorithms enhance anomaly detection by providing adaptive learning capabilities that improves the

detection accuracy and reduce false positives [28]. The existing IDS solutions still face several unresolved challenges in IIoT environment. Signature-based systems are ineffective against zero-day attacks, while anomaly-based systems often suffer from high false alarm rates, and hybrid models struggle to maintain both speed and scalability when processing massive volumes of IIoT data. Moreover, the heterogeneity and dynamic nature of IIoT devices makes accurate classification increasingly difficult. These challenges highlight the need for an IDS framework that simultaneously improves detection accuracy, reduces false positives, and ensures efficient real-time processing in large-scale IIoT networks.

The objective of this research is to design and develop a hybrid IDS framework for IIoT environments, so as:

1. To improve detection accuracy by employing an enhanced classification algorithm.
2. To reduce computational overhead by optimizing hash code generation and feature selection processes.
3. To ensure scalability and robustness of the IDS framework for large-scale IIoT environments.

The contributions of this work are summarized as follows:

- A hybrid IIDS model is proposed which integrates, rotation-based Right-Shift Strategy with Grøstl Hashing Algorithm (RSS-GHA) for faster hash code generation, Kullback-Leibler Divergence (KLD) based Sailfish Optimization Algorithm (KLD-SOA) for efficient feature selection, and the SoftSwish Gated Recurrent Unit (SSGRU) for improved classification.
- The proposed method is evaluated on multiple datasets, including benchmark (Newer version of the KDD cup 1999 dataset (NSL-KDD), UNSW-NB15) and IIoT-specific datasets (water tank, gas pipeline, Washington University in St. Louis (WUSTL), IIoT 2021), ensuring robustness and practical applicability.
- The proposed RSS-GHA is benchmarked against prevailing hashing algorithms namely, Grøstl, Swifft, MD5, and SHA-512 in terms of hash code generation time.
- The proposed KLD-SOA algorithm is evaluated in terms of feature selection time and fitness function optimization, and the results are compared with SOA, Honey Badger optimization Algorithm (HBA), Cuckoo Search optimization Algorithm (CSA), and Bacteria Foraging Optimization Algorithm (BFOA), highlighting its efficiency and effectiveness.
- Performance evaluation parameters namely detection accuracy, precision, recall, specificity and F1-score have been considered to quantitatively demonstrate the effectiveness of proposed SSGRU framework, with respect to existing models, namely GRU, Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Deep Neural Network (DNN) for different datasets.

The structure of this paper is as follows: section 2 reviews related works and their limitations, section 3 outlines the proposed methodology, section 4 presents the results and discussion, and section 5 concludes the study while suggesting future directions.

2. Literature Survey

Huong *et al*. [11] presented an approach for cyber-attack detection that utilizes anomaly detection in an ICS. To detect the attacked data, a Variational Auto-Encoder with Long Short-Term Memory techniques (VAE-LSTM) was employed. Anomaly detection was used to identify patterns in data and security threats. Nevertheless, imbalanced data distribution across manufacturing sites caused bias in pattern learning for the federated models, leading to inaccurate anomaly detection that diminished the effectiveness of the model.

The study of Soliman *et al*. [26] accomplished a DL framework for IIDS to secure industrial IoT. To increase the detection rate, a singular value decomposition technique was employed to diminish the data features. GRU and Synthetic Minority Over Sampling Techniques (SMOST) were used in this study to mitigate overfitting and underfitting issues. This methodology had a higher accuracy rate and diminished error rate. Nevertheless, GRU had the longest training time and slower convergence, which decreased the overall efficacy of the system.

Wang *et al*. [34] developed a stacked DL methodology for the detection of malicious cyber-attacks targeting Supervisory Control and Data Acquisition (SCADA) systems. SCADA systems are continuously being exposed to numerous heterogeneous cyber threats. To detect cyber-attacks in industry, this methodology also employed the random forest procedure. Therefore, when compared with other existing studies, this model enhanced SCADA systems by accurately detecting malicious intrusions. However, the system's performance was degraded by an increase in false positive values.

Arshad *et al*. [3] introduced an intrusion detection framework for Industrial IoT devices. With minimal energy consumption, intrusion was effectively detected using this method. This system evaluated the potential for collaboration between IoT devices (hosts) and edge devices to improve the efficiency of intrusion detection. The framework was implemented using the Contiki OS. This framework was found suitable for resource-constrained environments. Therefore, the effectiveness of detecting cyber-attacks was improved. However, the inherent privacy measures for host nodes were inadequate, especially in cross-network collaboration, when collaborative intrusion detection was effective. Therefore, this system posed a significant challenge in

protecting sensitive information.

A Population Extremal Optimization (PEO)-centered Deep Belief Network (DBN) methodology for cyber-attack detection in SCADA-based Industrial Automation and Control Systems (IACS) was presented by Lu *et al.* [18]. PEO was employed to determine the DBN parameters for cyber-attack detection in SCADA-centered IACS. An Ensemble learning (En) scheme named En-PEO-DBN was utilized for effective improvement in detection performance. Despite accurate detection, the PEO-DBN suffered from prolonged fitness evaluation times whenever the DBN parameters were tuned automatically. Therefore, the overall system efficiency was affected.

A stealthy cyber-attack on an ML-IDS in ICS was propounded by Chen *et al*. [5]. This study employed two attack strategies: an optimal solution to find stealthy attacks suitable for smaller and more flexible samples, and a Generative Adversarial Network (GAN) attack, which was efficient for larger and less flexible samples. However, the attacks in optimal solutions lacked flexibility in this system, especially in ICS with unchangeable features and scenarios with small data sizes. Therefore, this affected the overall performance of the system.

Liu *et al*. [17] developed a hierarchically distributed intrusion detection scheme for the safety protection of industrial cyber physical system. In this study, by utilizing the anomaly-based Process Noise and Measurement Noise-Adaptive Kalman Filter (PNMN-AKF), potential and covert attacks were detected. Using the Forgetting Factor-induced Recursive Gaussian Mixture Model (FF-RGMM), cyber-attacks were identified. The sparse-DBN technique characterized misuse behavior for detecting potential attacks. Therefore, both known and unknown attacks were detected in this framework. Nevertheless, owing to the uncertain behavior of the captured attacking nodes, the detection rate was relatively low during an unknown attack. Therefore, it affected the efficacy of the system.

Wang *et al*. [36] introduced K-Density-Based Spatial Clustering of Applications with Noise (K-DBSCAN), an anomaly detection algorithm for seasonal time series data. It combined autocorrelation analysis with K-means clustering for daily-level granularity before DBSCAN application, improving anomaly detection compared to DBSCAN and A-DBSCAN, particularly for local anomalies. However, limitations include manual parameter selection, limited generalizability beyond temperature data, and dependence on K-means clustering. Experiments on Beijing and Sanya temperature data demonstrated its effectiveness, but further research needed for broader applicability and adaptive parameter tuning.

Venkatraman and Surendiran [31] developed an adaptive hybrid IDS tailored for multimedia IoT environments, addressing the limitations of traditional IDS in adapting to dynamic IoT communications. The approach used a timed automata controller to improve the detection accuracy against attacks such as Denial of Service (DoS) and control hijacking, achieving a detection rate of 99.06%. While effective in smart city contexts, the system's focused on specific attack types, limiting its ability to address a broader range of threats in multimedia IoT systems, thereby affecting its overall comprehensiveness.

2.1. Research Gaps

Although various IDS models employing both signature-based and anomaly-based attack detection have been developed, they continue to encounter certain drawbacks. The specific limitations are as follows:

1. Most existing approaches suffer from slow processing speeds because of the large volume of IIoT data.
2. Several methods require an extensive amount of time to select appropriate features, which increases the training duration.
3. Signature-based detection systems struggle to keep up with rapidly evolving attack patterns because they rely on predefined attack signatures and thus are ineffective against zero-day attacks.
4. Anomaly based systems often generate many false positives owing to the variability of normal behavior in IIoT environments, making it difficult to differentiate between benign and malicious activities.
5. Most systems focus predominantly on either signature-based or anomaly-based detection, leading to a reduced overall detection accuracy.
6. Many current systems are limited to classifying only specific attack types, making them ineffective for detecting new or unknown attacks.
7. The dynamic and heterogeneous nature of IIoT devices leads to frequent misclassifications, as most existing systems lack adaptability to new IIoT-specific attack vectors.

To address these issues, a hybrid IIDS framework utilizing Residual Sum of Squares-Generalized Hebbian Algorithm (RSS-GHA), SOA, and SSGRU has been proposed. This approach aims to improve both the speed and accuracy of detection, ensuring robust defense against evolving cyber threats in IIoT systems. The proposed technique aims to build an effective IDS model for IIoT data centered on an enhanced classification algorithm, and reduced features.

3. Proposed Methodology

A hybrid IDS that leverages both signature-based and anomaly-based detection is proposed in this study. First, a signature-based hashing technique is applied to detect known attack patterns. If no match is found, the data are passed to a classifier employing deep learning, which detects anomalies after feature selection using an optimization algorithm. This approach ensures

comprehensive detection of both known and novel attacks. A flow diagram illustrating the proposed method is presented in Figure 1.

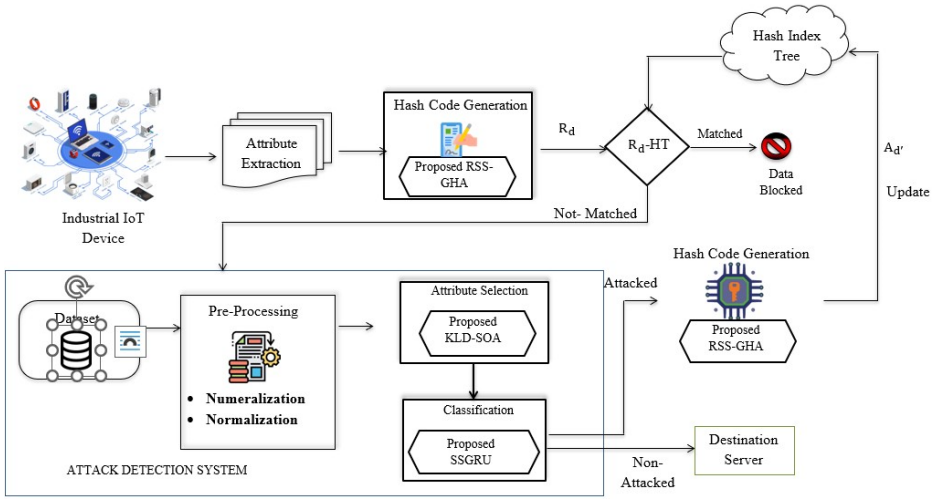


Figure 1. Flow diagram of the proposed work.

3.1. Signature Based Detection

Signature-based detection is a security technique that identifies threats by comparing them to a database of known patterns or signatures, such as malware code or attack behaviors. While effective against known threats, it struggles to detect new, unknown, or evolving threats, making it less adaptable to zero-day attacks.

3.1.1. IIoT Device Initialization

Initially, to monitor the industrial environment, IIoT sensor devices are initialized with the node ID, energy, and location. After initialization, data sensing is performed. In the next step, network attributes are extracted to classify the data.

3.1.2. Attribute Extraction

The Network Traffic Attributes (NTA), namely the Internet Protocol (IP) address, command address, command memory, response address, response memory, and control mode are extracted. Specific characteristics and information derived from the communication patterns of IIoT devices are called NTAs. The extracted attributed data (R_d) are given in Equation (1),

$$R_d = [R_1, R_2, \ldots \ldots R_p] \; where \; d = 1 \; to \; p \qquad (1)$$

p is the total number of extracted NTAs. Subsequently, hash code is generated for (R_d) and matched with an index tree to analyze the attack. (R_d) is transmitted to the attack detection system to detect intrusions if no data are present in the index tree or if (R_d) is not matched with the data in the index tree. If (R_d) matches the data in the index tree, then it is blocked from transmission. The hash code generation process is described in detail below.

3.1.3. Hash Code Generation

By employing RSS-GHA, a hash code is generated. Both SHA-512 [22] and Grøstl [14] are cryptographic hash functions designed to provide strong security; however, SHA-512 typically involves more complex operations and a larger internal state, making it slower in many cases. The vulnerability of MD5 has been exploited in many notable security breaches, such as creating fake authentication systems [35], whereas Grøstl is SSL certificates and malware targeting hash-based specifically designed to resist attacks that are effective against MD5, including collision attacks. The prevailing GHA partitions the input message into fixed size blocks and maintains a hash state, which is a matrix with a size at least twice that of the final output. The substitution and permutation steps are involved in Grøstl to effectively process the input data. However, Grøstl creates identical hash values for the same input data owing to a distinct permutation, thereby degrading the overall process [19]. A rotation-based RSS is added to address this problem. The overall workflow of the algorithm is illustrated in Figure 2.

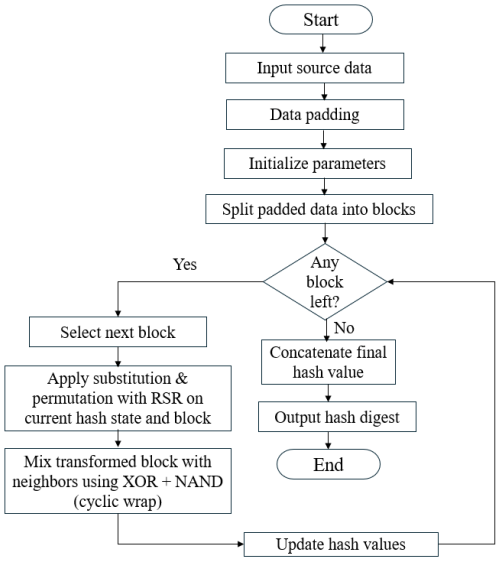


Figure 2. Flowchart of proposed RSS-GHA algorithm.

The RSS involves moving the data to the right and shifting the rightmost bit to the left. More variability is introduced by RSS in hash computation. Additional diffusion by the incorporation of RSS in GHA helps the hash function mix the data even better, adding another layer of security and reducing the chance that an attacker can reverse-engineer the hash to find the original data. The detailed mathematical model of the proposed RSS-GHA is explained below using Equations (2) to (8).

- **Padding**: initially, the input source data (R_d) is partitioned into fixed-size blocks denoted as (M_k). To prevent data loss, the last block is padded if its length is smaller than the required block size.
- **Initialization and block processing**: subsequently, the hash state (h_0) is initialized, and the input data are divided into k' blocks, where k' denotes the total number of fixed-size blocks (including any padded block) obtained from the input data. Each block (M) is represented as,

$$M_k = \{M_1, M_2, \dots \dots M_{k'}\} \; where \; k = 1 \; to \; k' \tag{2}$$

For the k^{th} block, the processing matrix is initialized by eXclusive OR (XOR) with the previous chaining value such that the initial hash state h_0 is the Initialization Vector (IV) used by RSS-GHA: a 512-bit block initialized to zeros with the intended digest length encoded in the last 64 bits.

$$Q^{(0)} = M_k \oplus h_{k-1} \tag{3}$$

To generate hash-code for (R_d), the initialized processing matrix $Q^{(0)}$ is iterated with the (R') number of rounds using the following steps:

- *Step* 1: Substitution. After initialization, the first transformation applied to the processing matrix is the substitution step. This operation introduces non-linearity into the hashing process to strengthen resistance against cryptanalytic attacks. In this phase, each element of Q located at row i and column j is replaced using the round-dependent S-box ($S^{(r)}$), which is a nonlinear mapping defined for the r^{th} round. This ensures that a small change in the input data produces unpredictable differences in the hash state. The substitution is mathematically expressed as Equation (4).

$$\tilde{Q}_{ij}^{(r)} = S^{(r)}\left(Q_{ij}^{(r-1)}\right) \forall \; i,j \tag{4}$$

where ($\tilde{Q}^{(r)}$) is the substituted matrix for round r.

- *Step* 2: Permutation. Subsequently, permutation occurs with the inclusion of RSS. RSS inclusion causes additional diffusion in the hash state to generate distinct hash values for diverse input data. This step involves shifting specific bits in the hash state to the right and rotating the rightmost bits to the left. This is given in Equation (5),

$$Q'^{(r)}_{ij} = \tilde{Q}^{(r)}_{(i-s_j \;\; mod \;\; N_r),j} >>> \rho_i \tag{5}$$

Where, the permuted matrix in round r is notated as $Q'^{(r)}$, row-shift applied to column j is symbolized as s_j; the total number of columns in the matrix is proffered as N_r; the modulus function is signified as *mod*; ρ_i is the right-rotation count for row i, and >>> represents circular right rotation.

- *Step* 3: Matrix mixing. To generate a mixed matrix, the resulting permuted matrix columns are shuffled and computed as

$$Q''^{(r)}_{ij} = Q'^{(r)}_{ij} \oplus \left(Q'^{(r)}_{(i+1) \; mod \; N_r,j} \mid Q'^{(r)}_{(i+2) \; mod \;\; N_r,j}\right) \tag{6}$$

where the mixed matrix is denoted as $Q''^{(r)}$ in round r, the bitwise XOR and NAND operators are denoted as $\oplus$ and $\mid$, and $Q'^{(r)}_{(i+1) \; mod \; N_r,j}$ and $Q'^{(r)}_{(i+2) \; mod \; N_r,j}$ represent the data in the same column (j) of the next row ((i+1) *mod* N_r,j) and the second next row ((i+2) *mod* N_r,j).

- *Step* 4: Final hash state. Finally, the XOR operation is performed using the round constant ($R'_{C_{ij}}$), combined with the permuted matrix output, and integrated with the intermediate hash state ($h_{l'}$). These are formulated as follows.

$$h_{l'} = \left(Q''_{ij} \oplus R'_{C_{ij}}\right) \oplus h_{l'-1} \tag{7}$$

Therefore, the concatenation of the final hash state is given by,

$$\overleftrightarrow{h} = h_1, h_2, \dots \dots h_{n'} \; where \; l' = 1 \; to \; n' \tag{8}$$

Here, the generated hash code is denoted as $\overleftrightarrow{h}$, and the total number of $\overleftrightarrow{h}$ is denoted as n'.

Subsequently, $\overleftrightarrow{h}$ is matched with a hashed index tree. If a signature match is identified, the attack is detected and flagged for reporting purposes. Otherwise, the data is forwarded to the anomaly detection module for further analysis.

Algorithm 1: Pseudo code of proposed RSS-GHA algorithm.

Input: Input source data (R^d)
Output: Generated hash code $\overleftrightarrow{h}$
Begin
  Pad input into blocks, M_1, ... $M_{k'}$
  Initialize hash state, (h_0)
  For each block, M_k
    Initialize the matrix, $Q^0 = M_k \oplus h_{k-1}$
    For r = 1 to R'
      Calculate substitution,
      $\tilde{Q}_{ij}^{(r)} = S^{(r)}\left(Q_{ij}^{(r-1)}\right)$
      Compute permutation,
      $Q'^{(r)}_{ij} = \tilde{Q}^{(r)}_{(i-s_j \;\; mod \;\; N_r),j} >>> \rho_i$
      Evaluate mixed matrix,
      $Q''^{(r)}_{ij} = Q'^{(r)}_{ij} \oplus \left(Q'^{(r)}_{(i+1) \; mod \; N_r,j} \uparrow Q'^{(r)}_{(i+2) \; mod \; N_r,j}\right)$
  End for
  Calculate Final Hash State
  $h_{l'} = \left(Q''_{ij} \oplus R'_{C_{ij}}\right) \oplus h_{l'-1}$
End for

Concatenate all hash values and return hash code
End

To classify the attacked and non-attacked data, the mismatched hash code data is then assigned to an attack detection system. The pseudo code of proposed RSS-GHA algorithm is given in Algorithm (1).

3.2. Anomaly based Attack Detection System

In this methodology, the classifier undergoes a training phase, enabling it to learn the attack patterns and distinguish them from normal behavior. Once trained, the classifier is deployed for attack detection by analyzing the input data to identify potential security threats. The training process consists of several steps, such as data preprocessing, feature extraction, feature selection, and classification.

3.2.1. Pre-Processing

Primarily, data attributes (R_d) undergoes pre-processing to transform the raw attributes into a form that is appropriate for classification. The pre-processing steps, namely numeralization and normalization, are executed.

1. Numeralization: here, (R_d) is numeralized to convert categorical data into a numerical format. It assigns unique numbers to categories, thus facilitating algorithms to process and analyze data effectively. The numeralized data (N_d) is given in Equation (9),

$$N_d = N_1, N_2, \ldots, N_e \quad where\ d = 1\ to\ e \tag{9}$$

The total amount of numeralized data (N_d) is signified as e. Subsequently, normalization of (N_d) occurs, as explained in the following section.

2. Normalization: from (N_d), normalization is performed using the minimum and maximum values to scale and transform the data to a common range or distribution. Normalization ensures fair contributions from all the data features and improves the model convergence, performance, and stability of the algorithms. The normalized data (W_d) is given as,

$$W_d = \frac{N_d - N_{d_{\min}}}{N_{d_{\max}} - N_{d_{\min}}} \tag{10}$$

Here in Equation (10), the minimum and maximum values of (N_d) are represented by $N_{d_{\min}}$ and $N_{d_{\max}}$. Subsequently, the attributes are extracted from the normalized data (W_d).

3.2.2. Attribute Extraction and Selection

Attributes namely IP address, command address, memory address, response address, response memory, and control mode are extracted from this normalized data (W_d), and optimal features are chosen for the effective classification of diverse types of attacks. The s' numbers of the extracted attributes (H) are represented by Equation (11),

$$H = h_1, h_2, \ldots, h_{\acute{s}} \tag{11}$$

Attribute selection then occurs using the KLD-SOA algorithm. The SOA, which is a population-based meta-heuristic algorithm, is inspired by the attack-alternation strategy of hunting sailfish [24]. A group of sailfishes hunting a school of sardines with efficient attack and energy-saving alternation is involved in the SOA strategy. The selection of attributes is optimized using the SOA algorithm based on the defined criteria. However, a random coefficient generation process exists in SOA. As randomness hinders reproducibility, SOA cause inconsistent outcomes. The stability and reliability of the optimization process may be affected by the lack of control over the generated coefficients. Therefore, the KLD is used in the coefficient generation process to address this issue. The KLD adds a probabilistic constraint and augments optimization stability. The KLD-SOA algorithm is described as follows:

- *Step* 1: Initialization. Initially, the primary populace ($H_{s'd}$), with a diverse subset of attributes, is initialized as

$$H_{\acute{s}d} = \begin{bmatrix} h_{1,1}\ h_{1,2} \ldots h_{1,d} \\ h_{2,1}\ h_{2,2} \ldots h_{2,d} \\ \vdots \quad \vdots \quad \cdots \quad \vdots \\ h_{s',1}\ h_{s',2} \cdots h_{s',d} \end{bmatrix} \tag{12}$$

where the s' number of data attributes with dimensions d is notated as ($H_{s'd}$).

- *Step* 2: Fitness calculation. Subsequently, the fittest solution (f^T) is computed. This is utilized to choose the optimal data attributes centered on the highest classification accuracy, (C_a).

$$f^T = max(C_a) \tag{13}$$

- *Step* 3: Elite matrix generation. Later, an elite matrix (E^m) is generated from the fittest solution to obtain the subset of attributes with the best solutions (f'_{best}), and is given as Equation (14),

$$E^m = f'_{best} \tag{14}$$

Here, $H_{s'}$ denotes a subset of attributes that required improvement or adjustment. In addition, the elite matrix (e^m) for attribute improvement (f''_{imp}) to obtain the best solution is given as:

$$e^m = f''_{imp} \tag{15}$$

- *Step* 4: Attribute alternation strategy. After that, in the g^{th} iteration, the newly selected attributes are updated. Here, to enhance the effective optimization, KLD is used rather than the random generation process. This is represented as Equation (16),

$$X^g_{new.H_{s'd}} = X^g_{E^m} - \lambda_g * \left(K^l * \left(\frac{X^g_{E^m} + X^g_{e^m}}{2} \right) - X^g_{old,H_{s'd}} \right) \tag{16}$$

where the newly selected attribute X from ($H_{s'd}$) in the g^{th} iteration is notated as $X^g_{new.H_{s'd}}$, the (e^m) having best

solution is symbolized as $X_{E^m}^{g}$, the coefficient in the g^{th} iteration is signified as λ_g, the subset of attributes that need improvement or adjustment is elucidated as $X_{e^m}^{g}$, the old best solution is interpreted as $X_{old,H_{s'd}}^{g}$, and the KLD coefficient is given as (K^l), which is expressed as,

$$K^l = \Sigma X_{new.H_{s'd}}^{g} \log \frac{X_{new.H_{s'd}}^{g}}{X_{old.H_{s'd}}^{g}} \tag{17}$$

$$\lambda_g = 2 * K^l * A_s - A_s \tag{18}$$

The attribute selection density (A_s) is given by Equation (19),

$$A_s = 1 - \left(\frac{G_{H_{s'd}}}{G_{H_{s'd}} + G_{H_{s'}}} \right) \tag{19}$$

Here, the total number of data attributes and attributes that require improvement in each iteration are referred to as $G_{H_{s'd}}$ $G_{H_{s'}}$.

- *Step* 5: Identifying and updating data attributes. Finally, the best solution in the g^{th} iteration is updated as

$$X_{new,H_{s'}}^{g} = X_{E^m}^{g} - K^l * (X_{E^m}^{g} - X_{old,H_{s'}}^{g}) \tag{20}$$

Exploration-exploitation balance E_b^2 is employed in the optimization for controlling the exploration of attribute subsets and is given as,

$$E^p = B_e * (1 - (2 * g * \varpi)) \tag{21}$$

In Equation (21), the exploration parameter that controls the (E_b^2) is specified as E^p, the initial value to start the E_b^2 is notated as B_e, the coefficient that influences the rate at which the E^p decreases over time is represented as ϖ, and the g^{th} iteration is denoted as g.

- *Step* 6: Attribute and variable updation. Finally, during each iteration, the attributes and variables are updated and are given by Equations (22) and (23) respectively,

$$\alpha^a = G_{H_{s'}} * E^p \tag{22}$$

$$\beta^v = v_g * E^p \tag{23}$$

where the updated attributes and variables are represented as α^a, β^v and the number of variables in the g^{th} iteration is denoted as v_g.

- *Step* 7: Output. After that, the selected attributes are signified as (S_{T_r}), and the u' numbers of S_{T_r} are represented as Equation (24),

$$S_{T_r} = S_1, S_2, \dots, S_{u'} \; where \; T_r = 1 \; to \; u' \tag{24}$$

These selected attributes are assigned to the classifier to determine the attacked and non-attacked data.

3.2.3. Classification and Attack Detection

Finally, using the SSGRU algorithm, attacked and non-attacked data are classified from the selected attributes. By efficiently capturing the temporal dependencies in data attributes, cyber threat classification is enhanced by employing the GRU algorithm [6]. The GRU assists in discerning patterns of attacks and normal behavior, thus contributing to accurate classification. However, GRU has limitations, such as low learning efficiency and slow convergence. Therefore, SoftSwish (SS) activation is employed in the GRU rather than the hyperbolic tangent function to address this issue. Smoother gradients are provided by SS activation, potentially assisting convergence and increasing model performance. Figure 3 shows the SSGRU classifier.

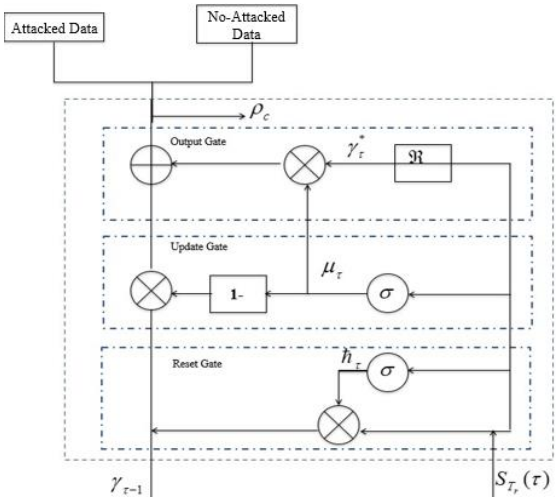


Figure 3. Architecture of the proposed SSGRU classifier.

The SSGRU algorithm is described as follows.

- **Initialization**: initially, the chosen attributes, (S_{T_r}) are rendered as input to the classifier.
- **Reset gate and update gate**: the reset gate ($\hbar_\tau$) controls information retention and facilitates the methodology to forget irrelevant patterns adaptively. Update gate (μ_τ) regulates the blending of new and prior information. This is expressed in Equations (25) and (26) respectively,

$$\hbar_\tau = \sigma'(W_\hbar . [\gamma_{\tau-1}, S_{T_r(\tau)}]) \tag{25}$$

$$\mu_\tau = \sigma'(W_\mu . [\gamma_{\tau-1}, S_{T_r(\tau)}]) \tag{26}$$

Where, the weights of reset and update gates are signified as $W_\hbar$ and W_μ, the hidden state function with time τ is notated as γ, and the sigmoid activation function is symbolized as σ' and is given as,

$$\sigma'(S_{T_r}) = \frac{1}{1 + \vartheta^{-S_{T_R}}} \tag{27}$$

Here in Equation (27), the exponential function is denoted by ϑ.

- **Hidden state**: To overcome the problems of slow convergence and low learning efficiency, the candidate hidden state (γ_τ^*) with weight (W) and SS activation was computed. This is given as,

$$\gamma_\tau^* = \Re(W . [\hbar_\tau * \gamma_{\tau-1}, S_{T_r(\tau)}]) \tag{28}$$

In Equation (28), the SS activation function is represented by the ($\Re$) function. The SS activation

function ($\Re$) with an exponential value (e) and input (S_{T_r}) is expressed as

$$\Re(S_{T_r}) = \frac{S_{T_r} e^{S_{T_r}}}{1 + \sum_{T_r=1}^{\mu'} e^{S_{T_r}}} \quad (29)$$

- **Output gate**: the output gate (ρ_τ) determines which information of the hidden state is shared to generate the output. The SSGRU output is presented as:

$$\rho_\tau(1 - \mu_\tau) * \gamma_{\tau-1} + \mu_\tau * \gamma_\tau^* \quad (30)$$

Therefore, this classifier classifies the data as attacked or non-attacked, which are denoted as ($A_{d'}$) and ($N_{a'}$). The pseudo code of the proposed SSGRU is presented in Algorithm (2) to illustrate its step by step working.

Finally, the destination server is reached by non-attacked data ($N_{a'}$). However, hash code is generated from the attacked data ($A_{d'}$) using the RSS-GHA algorithm, which is discussed in section 3.1.3, and is stored in the index tree.

In real time, the data from IIoT sensor devices are collected, the hash code is generated, and it is verified with an index tree for attack detection. If the hash code matches, the data are blocked from being transmitted. Otherwise, the data are forwarded to the attack detection system to detect intrusions.

Algorithm 2: Pseudo code of proposed SSGRU.

Input: Selected features, (S_{T_r})
Output: Classified result, (Ad') and (Na')
Begin
Initialize (S_{T_r}), iteration(t), and maximum iteration (t^{max})
While t<t^{max}
Calculate reset and update gates,

$$\hbar_\tau = \sigma'(W_{\hbar}.[\gamma_{\tau-1}, S_{T_r(\tau)}])$$
$$\mu_\tau = \sigma'(W_{\mu}.[\gamma_{\tau-1}, S_{T_r(\tau)}])$$

Compute hidden state,

$$\gamma_\tau^* = \Re(W.[\hbar_\tau * \gamma_{\tau-1}, S_{T_r(\tau)}])$$

Evaluate output gate,

$$\rho_\tau(1 - \mu_\tau) * \gamma_{\tau-1} + \mu_\tau * \gamma_\tau^*$$

End while
Return→($A_{d'}$), ($N_{a'}$).
End

4. Results and Discussion

The proposed approach is evaluated on multiple datasets to measure its performance and effectiveness using Python. Key metrics, namely accuracy, detection rate, precision, recall, F1-score, and hash code generation time, are analyzed. The results indicate that the hybrid IDS framework effectively identifies attacks by leveraging the signature based and anomaly-based detection mechanisms.

For each dataset, the system demonstrated consistent performance, with high detection accuracy and minimal false positives. Additionally, the right-shift rotation optimization in the GHA improved the hash code generation time, further enhancing the efficiency of the signature-based module.

4.1. Dataset Description

In this study, five publicly available datasets, namely NSL-KDD [30], water tank [20], gas pipeline [21], WUSTL-IIOT-2021 [38], and UNSW-NB15 [7] are used. The detailed information on the datasets is provided in Table 1. In the proposed method, 80% of the data is utilized for training, while the remaining 20% is designated for testing.

To demonstrate the effectiveness of the proposed method, the performance of both proposed RSS-GHA and proposed SSGRU is evaluated and compared with baseline algorithms.

Table 1. Datasets used in the performance analysis of proposed system.

S. No.	Dataset	No. of features	Normal instances	Attack instances	Total instances
1	NSL-KDD dataset	42	77054	71463	148517
2	Water tank dataset	24	172415	63764	236179
3	Gas pipeline dataset	27	61156	35863	97019
4	WUSTL-IIOT-2021 dataset	41	1107448	87016	1194464
5	UNSW-NB15 dataset	49	93000	164673	257673

4.2. Performance Evaluation of RSS-GHA

RSS-GHA combines bitwise right shift rotation operation with the Grøstl hashing approach for high-speed data processing and to enhance cryptographic security. The proposed RSS-GHA is compared with the prevailing Grøstl, Swifft, Message Digest 5 (MD5), and Secure Hash Algorithm 512 (SHA512) in terms of hash code generation time.

The analysis of hash code generation time in Figure 4 indicates that, for smaller datasets, the performance of the algorithms exhibits variability without a discernible pattern. However, as the size of the sensed data increases, a distinct trend emerges, with RSS-GHA demonstrating significantly better performance than the other algorithms. This highlights the efficiency and scalability of RSS-GHA in handling larger data volumes.

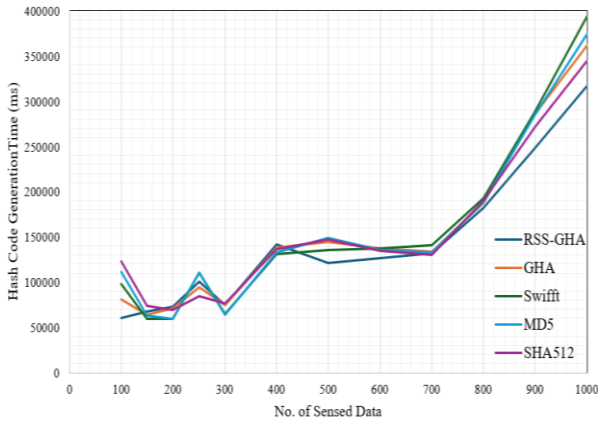


Figure 4. Comparison of hash code generation time.

4.2.1. Cryptanalysis Analysis of RSS-GHA

Along with the performance analysis, the security soundness of the proposed hashing algorithm RSS-GHA

is equally critical. Therefore, determinism and correctness are presented in this subsection along with cryptanalysis of proposed RSS-GHA.

1) Determinism and Correctness

Hashing functions should never give different results when the same input is presented. Formally,

$$\forall M, RSSGHA(M) = H$$

where M is the message being input and H is the hash computed.

As the RSS-GHA is only composed of deterministic operations (XOR, right-shift rotations, and substitution layers), the output is based on the input. This determines correctness of the design.

2) Cryptanalytic Attack Resistance

The proposed RSS-GHA is resistant to attacks and collisions [10, 23].

- Preimage resistance: it is computationally infeasible to find an input *M* that has a given hash *H* because it will take $O(2^{512})$ operations.
- Second preimage resistance: second preimage resistance is also required to find M such that RSSGHA(M)=RSSGHA(M'), and this too requires $O(2^{512})$ operations.
- Collision resistance: the proposed RSS-GHA produces a 512-bit digest. The complexity of finding a collision is at most bounded by:

$$O(2^{n/2})=O(2^{256})$$

where n=512 represents the length of the digest. This is computationally infeasible with the current computing power, which is a core component of a secure hashing algorithms.

- Differential cryptanalysis: non-linearity and diffusion are enhanced by the combination of both substitution and right-shift rotations; hence, it is resistant to both linear and differential cryptanalyses.

4.3. Performance Evaluation of KLD-SOA

The proposed KLD-SOA algorithm for feature selection is analysed based on feature selection time and fitness value. The graph analysing feature selection time as a function of the number of iterations in Figure 5-a) shows a performance comparison between the proposed KLD-SOA, SOA, HBA, CSA, and BFOA. The KLD-SOA demonstrated a consistently lower feature selection time across all iterations, underscoring its efficiency and superior computational performance over the other algorithms. The results in Figure 5-b) indicate that KLD-SOA achieves superior fitness values more efficiently across iterations, highlighting its effectiveness and optimization capability compared with the other algorithms. Here, fitness is the classification accuracy. The proposed method employs KLD as a coefficient in SOA generation to perform a stable and reliable optimization process. Therefore, as per the experimental outcomes, the data attributes were effectively selected by the proposed KLD-SOA within a minimum time.

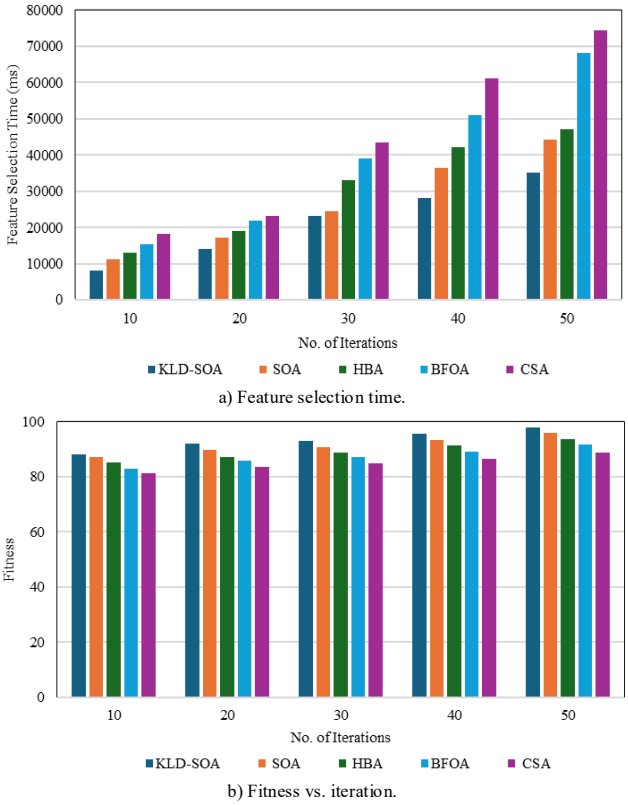


Figure 5. Performance evaluation of the proposed KLD-SOA feature selection method against metaheuristic algorithms.

4.4. Performance Evaluation of SSGRU

To evaluate the effectiveness of the hybrid intrusion detection system in predicting and classifying unseen network attacks, performance metrics accuracy, recall, precision, F1-score, and specificity are used. These metrics are calculated using True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) for both the normal and attack traffic. In this context, *TP* represents correctly identified normal records, *TN* refers to correctly detected attack records, *FP* represents the misclassification of attacks as normal, and *FN* represents the misclassification of normal records as attacks. The performance metrics were computed using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (31)$$

$$Recall = \frac{TP}{TP + FP} \quad (32)$$

$$Precision = \frac{TP}{TP + FN} \quad (33)$$

$$Specificity = \frac{TN}{TN + FP} \quad (34)$$

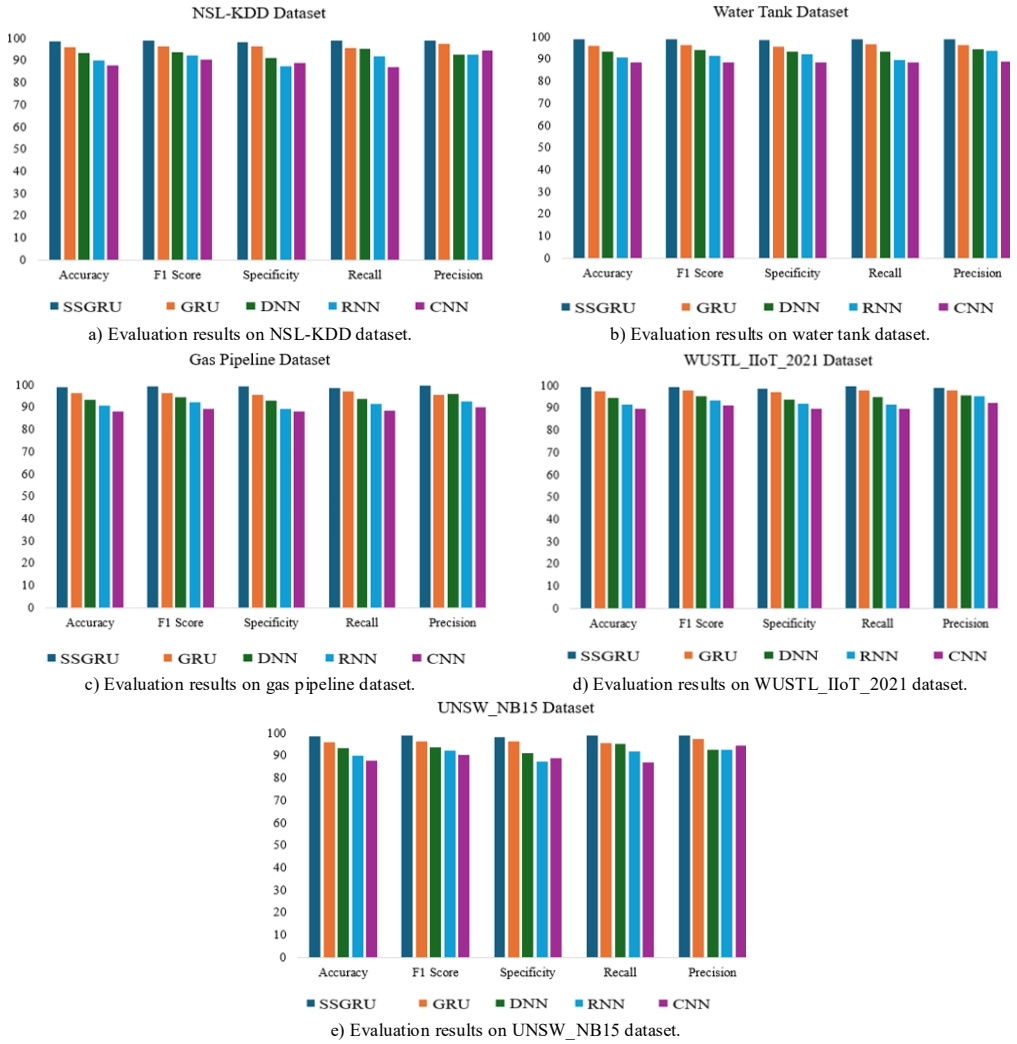
$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (35)$$

Accuracy represents the system's overall classification performance, while the *F1-score* provides a balance between precision and recall, with higher values indicating improved effectiveness. A high F1-score indicates that the model minimizes false positives while effectively detecting the attacks.

The classification performance of the hybrid system is summarized in Table 2.

Table 2. Classification results for the classifier algorithms on datasets.

Dataset	Algorithm	Accuracy	F1-score	Specificity	Recall	Precision
NSL-KDD	SSGRU	99.633	99.708	99.53	99.685	99.662
	GRU	97.732	98.22	96.939	98.183	98.256
	DNN	94.942	96.316	90.613	96.962	95.678
	RNN	92.667	94.761	85.582	95.821	93.724
	CNN	89.867	91.96	83.533	93.785	90.205
Water tank	SSGRU	98.874	98.973	98.727	98.995	98.951
	GRU	96.138	96.506	95.453	96.696	96.316
	DNN	93.555	93.959	93.546	93.564	94.357
	RNN	90.695	91.586	92.091	89.621	93.64
	CNN	88.442	88.685	88.367	88.514	88.856
Gas pipeline	SSGRU	98.871	99.191	99.362	98.662	99.725
	GRU	99.725	96.418	95.616	97.12	95.726
	DNN	93.403	94.695	93.093	93.586	95.831
	RNN	90.682	92.098	89.22	91.689	92.51
	CNN	88.255	89.268	88.069	88.406	90.146
WUSTL_IIOT_2021	SSGRU	99.271	99.398	98.617	99.701	99.098
	GRU	97.49	97.897	96.99	97.827	97.966
	DNN	94.329	95.245	93.639	94.79	95.705
	RNN	91.492	93.347	91.712	91.376	95.406
	CNN	89.687	91.002	89.621	89.734	92.307
UNSW_NB15	SSGRU	98.785	98.967	98.352	99.09	98.845
	GRU	95.883	96.528	96.351	95.569	97.507
	DNN	93.369	93.935	91.146	95.269	92.639
	RNN	90.199	92.218	87.409	91.819	92.62
	CNN	87.606	90.53	88.868	87.014	94.343



a) Evaluation results on NSL-KDD dataset.

b) Evaluation results on water tank dataset.

c) Evaluation results on gas pipeline dataset.

d) Evaluation results on WUSTL_IIoT_2021 dataset.

e) Evaluation results on UNSW_NB15 dataset.

Figure 6. Performance evaluation of the proposed SSGRU classifier against existing deep learning approaches across benchmark datasets.

The performance analysis of the proposed SSGRU and prevailing methods namely GRU, RNN, CNN, and DNN for different datasets are depicted in Figure 6. Compared to other classification algorithms, the proposed model achieved the highest accuracy of 99.63%, outperforming the baseline algorithm GRU by 1.94% only and the CNN algorithm by 10.88% in the NSL-KDD dataset. Therefore, compared to other prevailing studies, the proposed SSGRU exhibit superior results. Therefore, an accurate attack detection is demonstrated using the proposed SSGRU.

4.5. Comparative Analysis

The detection rate measures the proportion of actual attacks correctly identified by an IDS. In the IIoT, determining the detection rate is crucial because a high detection rate ensures that the IDS accurately identifies and responds to intrusions, preventing security breaches that can lead to operational disruptions, data loss, or device manipulation in critical industrial environments.

A comparative study of the proposed technique with the newly developed SMOST [10], Bidirectional Long Short-Term Memory Explainable Artificial Intelligence (BiLSTM-XAI) [23], DNN [33], Multi-Objective BAT (MOBAT) algorithm [9, 25], DNN [15], and Optimized Intra/Inter-Class-Structure-based Variational Few-Shot Learning (OICS-VFSL) techniques [16] is presented in Table 3 to demonstrate its suitability for IIoT intrusion detection. Most studies used the NSL-KDD dataset to evaluate the performance of the system; therefore, a comparison was performed based on this dataset. Higher accuracy, precision, and f-measure values of 98.78%, 98.84%, and 98.96%, respectively, were attained by the proposed SSGRU. However, techniques such as SMOST, BiLSTM-XAI, DNN, and MOBAT obtained lower accuracy of 90.99%, 98.2%, 82.99%, and 96.12%, respectively. The prevailing OICS-VFSL attained a lower F-measure of 98%, which degraded the performance of the system. Therefore, when analogized with other top-notch models, the proposed method performed effectively in detecting attacked and non-attacked data.

Table 3. Comparative analysis with existing works.

Study	Techniques used	NSL-KDD dataset		
		Accuracy (%)	Precision (%)	F-measure (%)
Proposed work	SSGRU	98.78	98.84	98.96
Hash function [10]	SMOST	90.99	91.39	90.89
Rogaway and Shrimpton [23]	BiLSTM-XAI	98.2	95.8	95.1
Wang *et al.* [33]	DNN	82.99	85.15	81.77
Sivamohan and Sridhar [25]	MOBAT	96.12	78.9	-
Liang *et al.* [16]	OICS-VFSL	-	-	98

5. Conclusions

A hybrid intrusion detection framework for Industrial IoT is proposed in this paper, integrating the right RSS-GHA with the SSGRU based anomaly detection model. The proposed methodology begins with attribute extraction, followed by hash code generation using the RSS-GHA. The generated hash code is then compared against a hashed index tree. If a match is found, intrusion is reported; otherwise, the data are forwarded to the anomaly-based detection system for further analysis. RSS-GHA demonstrated a significant reduction in hash code generation time. This improvement is critical in the IIoT, where low-latency processing is essential for ensuring real-time attack detection and system responsiveness. The KLD-SOA algorithm is employed for feature selection, achieving optimal attribute selection while minimizing the time required for the process. The selected attributes have been fed into the SSGRU to classify the attacked and non-attacked data.

The results obtained from the five datasets demonstrated that the proposed model outperformed the baseline algorithms. Notably, the NSL-KDD dataset exhibited the best results, with precision (99.66%), F-measure (99.70%), specificity (99.53%), and accuracy (99.63%) surpassing those of the other models. The comparative analysis further validated the superior attack detection accuracy of the proposed system across all datasets.

The work may be extended in future to use federated learning to enhance system performance in decentralized environments and incorporate cryptographic techniques and load balancing to improve data security and collision avoidance. Additionally, focusing on scalability and real-time performance in large-scale IIoT networks, as well as optimizing the system for energy efficiency in resource-constrained devices may be beneficial.

References

[1] Alaketu M., Oguntimilehin A., Olatunji K., Abiola O., and et al., "Comparative Analysis of Intrusion Detection Models Using Big Data Analytics and Machine Learning Techniques," *The International Arab Journal of Information Technology*, vol. 21, no. 2, pp. 326-337, 2024. https://doi.org/10.34028/iajit/21/2/14

[2] Alem S., Espes D., Nana L., Martin E., and Lamotte F., "A Novel Bi-Anomaly-Based Intrusion Detection System Approach for Industry 4.0," *Future Generation Computer Systems*, vol. 145, pp. 267-283, 2023. https://doi.org/10.1016/j.future.2023.03.024

[3] Arshad J., Azad M., Abdeltaif M., and Salah K., "An Intrusion Detection Framework for Energy Constrained IoT Devices," *Mechanical Systems and Signal Processing*, vol. 136, pp. 1-13, 2020. DOI: 10.1016/j.ymssp.2019.106436

[4] Bansal K. and Singhrova A., "Review on Intrusion Detection System for IoT/IIoT: Brief Study,"

Multimedia Tools and Applications, vol. 83, pp. 23083-23108, 2024. https://doi.org/10.1007/s11042-023-16395-6

[5] Chen J., Gao X., Deng R., He Y., and et al., "Generating Adversarial Examples Against Machine Learning-based Intrusion Detector in Industrial Control Systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1810-1825, 2022. DOI: 10.1109/TDSC.2020.3037500

[6] Cho K., Merrienboer B., Bahdanau D., and Bengio Y., "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," *arXiv Preprint*, vol. arXiv:1409.1259v2, 2014. DOI: 10.48550/arXiv.1409.1259

[7] Divekar A., Parekh M., Savla V., Mishra R., and Shirole M., "Benchmarking Datasets for Anomaly-Based Network Intrusion Detection: KDD CUP 99 Alternatives" *in Proceedings of the IEEE 3rd International Conference on Computing, Communication and Security*, Kathmandu, pp. 1-8, 2018. DOI: 10.1109/CCCS.2018.8586840

[8] Etxezarreta X., Garitano I., Iturbe M., and Zurutuza U., "Software-Defined Networking Approaches for Intrusion Response in Industrial Control Systems: A Survey," *International Journal of Critical Infrastructure Protection*, vol. 42, pp. 1-17, 2023. DOI: 10.1016/j.ijcip.2023.100615

[9] Ghanem W., Ghaleb S., Aman J., Nasser A., and et al., "Cyber Intrusion Detection System Based on a Multiobjective Binary Bat Algorithm for Feature Selection and Enhanced Bat Algorithm for Parameter Optimization in Neural Networks," *IEEE Access*, vol. 10, pp. 76318-76339, 2022. DOI: 10.1109/ACCESS.2022.3192472

[10] Hash Functions, National Institute of Standards and Technology, https://csrc.nist.gov/projects/hash-functions, Last Visited, 2025.

[11] Huong T., Bac T., Long D., Luong T., and et al., "Detecting Cyberattacks Using Anomaly Detection in Industrial Control Systems: A Federated Learning Approach," *Computers in Industry*, vol. 132, pp. 1-16, 2021. https://doi.org/10.1016/j.compind.2021.103509

[12] Kaur S. and Singh M., "Hybrid Intrusion Detection and Signature Generation Using Deep Recurrent Neural Networks," *Neural Computing and Applications*, vol. 32, no. 17, pp. 7859-7877, 2020. https://doi.org/10.1007/s00521-019-04187-9

[13] Kim S., Jo W., and Shon T., "APAD: Autoencoder-based Payload Anomaly Detection for Industrial IoE," *Applied Soft Computing*, vol. 88, pp. 1-9, 2020. DOI: 10.1016/j.asoc.2019.106017

[14] Knudsen L., Gauravaram P., Matusiewicz K., Mendel F., and et al., "Grøstl-a SHA-3 Candidate," *Cryptology and Network Security*, pp. 1-42, 2011. DOI: https://www.groestl.info/Groestl.pdf

[15] Kunang Y., Nurmaini S., Stiawan D., and Suprapto B., "Attack Classification of an Intrusion Detection System Using Deep Learning and Hyperparameter Optimization," *Journal of Information Security and Applications*, vol. 58, pp. 1-15, 2021. DOI: 10.1016/j.jisa.2021.102804

[16] Liang W., Hu Y., Zhou X., Pan Y., and Wang K., "Variational Few-Shot Learning for Microservice-Oriented Intrusion Detection in Distributed Industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5087-5095, 2022. DOI: 10.1109/TII.2021.3116085

[17] Liu J., Zhang W., Ma T., Tang Z., and et al., "Toward Security Monitoring of Industrial Cyber-Physical Systems via Hierarchically Distributed Intrusion Detection," *Expert Systems with Applications*, vol. 158, pp. 1-23, 2020. DOI: 10.1016/j.eswa.2020.113578

[18] Lu K., Zeng G., Luo X., Weng J., and et al., "Evolutionary Deep Belief Network for Cyber-Attack Detection in Industrial Automation and Control System," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7618-7627, 2021. DOI: 10.1109/TII.2021.3053304

[19] Mendel F., Rijmen V., and Schlaffer M., "Collision Attacks on Round-Reduced Grostl," *in Proceedings of the International Workshop on Fast Software Encryption*, Berlin, pp. 509-521, 2014. https://doi.org/10.1007/978-3-662-46706-0_26

[20] Morris T. and Gao W., "Industrial Control System Traffic Data Sets for Intrusion Detection Research," *in Proceedings of the 8th International Conference on Critical Infrastructure Protection*, Arlington, pp. 65-78, 2014. https://inria.hal.science/hal-01386754v1

[21] Morris T., Thornton Z., and Turnipseed I., "Industrial Control System Simulation and Data Logging for Intrusion Detection System Research," *SEMANTIC SCHOLAR*, pp. 1-6, 2015. http://www.ece.uah.edu/~thm0009/icsdatasets/cyberhuntsvillepaper_v4.pdf

[22] National Institute of Standards and Technology, SHA-2 Standard: Secure Hashing Algorithm, NIST Special Publication 800-107 Revision 1, 2015. DOI: 10.6028/NIST.SP.800-107r1

[23] Rogaway P. and Shrimpton T., "Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance," *Fast Software Encryption*, vol. 3017, pp. 371-388, 2004. https://doi.org/10.1007/978-3-540-25937-4_24

[24] Shadravan S. and Asiabar S., "The Sailfish Optimizer: A Novel Nature-Inspired Metaheuristic Algorithm for Solving Constrained Engineering Optimization Problems,"

Engineering Applications of Artificial Intelligence, vol. 80, pp. 20-34, 2019. DOI: 10.1016/j.engappai.2019.01.001

[25] Sivamohan S. and Sridhar S., "An Optimized Model for Network Intrusion Detection Systems in Industry 4.0 Using XAI-based Bi-LSTM Framework," *Neural Computing and Applications*, vol. 35, no. 15, pp. 11459-11475, 2023. DOI: 10.1007/s00521-023-08319-0

[26] Soliman S., Oudah W., and Aljuhani A., "Deep Learning-based Intrusion Detection Approach for Securing Industrial Internet of Things," *Alexandria Engineering Journal*, vol. 81, pp. 371-383, 2023. https://doi.org/10.1016/j.aej.2023.09.023

[27] Soltani M., Ousat B., Siavoshani M., and Jahangir A., "An Adaptable Deep Learning-based Intrusion Detection System to Zero-Day Attacks," *Journal of Information Security and Applications*, vol. 76, pp. 103516, 2023. https://doi.org/10.1016/j.jisa.2023.103516

[28] Talukder A., Hasan K., Islam M., Uddin A., et al., "A Dependable Hybrid Machine Learning Model for Network Intrusion Detection," *Journal of Information Security and Applications*, vol. 72, pp. 103405, 2023. https://doi.org/10.1016/j.jisa.2022.103405

[29] Tama B., Lee S., and Lee S., "A Systematic Mapping Study and Empirical Comparison of Data-Driven Intrusion Detection Techniques in Industrial Control Networks," *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 5353-5380, 2022. https://doi.org/10.1007/s11831-022-09767-y

[30] Tavallaee M., Bagheri E., Lu W., and Ghorbani A., "A Detailed Analysis of the KDD CUP 99 Data Set," *in Proceedings of the IEEE Symposium on Computational Intelligence in Security and Defense Applications*, Ottawa, pp. 1-6, 2009. https://doi.org/10.1109/CISDA.2009.5356528

[31] Venkatraman S. and Surendiran B., "Adaptive Hybrid Intrusion Detection System for Crowd-Sourced Multimedia Internet of Things Systems," *Multimedia Tools and Applications*, vol. 79, pp. 3993-4010, 2020. DOI: 10.1007/s11042-019-7495-6

[32] Verdejo J., Calle J., Alonso A., Alonso R., and Madinabeitia G., "On the Detection Capabilities of Signature-Based Intrusion Detection Systems in the Context of Web Attacks," *Applied Sciences*, vol. 12, no. 2, pp. 1-16, 2022. https://doi.org/10.3390/app12020852

[33] Wang S., Xu W., and Liu Y., "Res-TranBiLSTM: An Intelligent Approach for Intrusion Detection in the Internet of Things," *Computer Networks*, vol. 235, pp. 1-16, 2023. DOI: 10.1016/j.comnet.2023.109982

[34] Wang W., Harrou F., Bouyeddou B., Senouci S., and Sun Y., "A Stacked Deep Learning Approach to Cyber-Attacks Detection in Industrial Systems: Application to Power System and Gas Pipeline Systems," *Cluster Computing*, vol. 25, no. 1, pp. 561-578, 2022. https://doi.org/10.1007/s10586-021-03426-w

[35] Wang X. and Yu H, "How to Break MD5 and other Hash Functions," *in Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Aarhus, pp. 19-35, 2005. https://doi.org/10.1007/11426639_2

[36] Wang X., Yang Y., Ding X., and Zhao Y., "Based on Correlation Analysis and K-Means: An Anomaly Detection Algorithm for Seasonal Time-Series Data," *The International Arab Journal of Information Technology*, vol. 21, no. 6, pp. 987-986, 2024. https://doi.org/10.34028/iajit/21/6/2

[37] Yahuza M., Idris M., Wahab A., Ho A., and Taha A., "Systematic Review on Security and Privacy Requirements in Edge Computing: State of the Art and Future Research Opportunities," *IEEE Access*, vol. 8, pp. 76541-76567, 2020. https://doi.org/10.1109/ACCESS.2020.2989456

[38] Zolanvari M., Teixeira M., Gupta L., Khan K., and Jain R., "WUSTL-IIOT-2021 Dataset for IIoT Cybersecurity Research," *Washington University*, 2021. http://www.cse.wustl.edu/~jain/iiot2/index.html

Komal Bansal is a Research Scholar in Department of Computer Science and Engineering at Deenbandhu Chottu Ram University of Science and Technology, Murthal, Sonepat, India. She holds an M.Tech degree from NIT Kurukshetra, Haryana, India and B.Tech. (Computer Science) from T.I.T. and S Bhiwani, Haryana, India. Her research interests include IoT, Industial IoT, Machine Learning, Deep Learning and Wireless Sensor Networks.

Anita Singhrova is a Professor of Information Technology and Computer Science at Deenbandhu Chottu Ram University of Science and Technology, Murthal, Sonepat, India. She holds a Ph.D. degree from GGS Indraprastha University, Delhi, India. She has completed M.E. (Computer Science and Engg.) from Punjab Engineering College, Chandigarh, India and B.Tech (Computer Science) from T.I.T and S. Her research interests include heterogeneous Networks, Wireless Networks, Machine Learning and Deep Learning.