Evolutionary Intelligence (2026) 19:23
https://doi.org/10.1007/s12065-025-01133-8

RESEARCH PAPER

# GA-optimized deep learning intrusion detection framework with LIME explainability for IoT networks

**Elijah M. Maseno**[1] · **Yanxia Sun**[1] · **Zenghui Wang**[2]

Received: 27 July 2025 / Revised: 17 October 2025 / Accepted: 11 December 2025


**Abstract**
The rapid expansion of the Internet of Things (IoT) has generated complex, high-volume network traffic that traditional signature-based Intrusion Detection Systems (IDS) struggle to analyze effectively. Conventional systems often fail to adapt to evolving attack behaviors and high-dimensional data, resulting in reduced accuracy and explainability. This study presents Genetic Algorithm (GA)-optimized deep learning models, namely GA-GRU and GA-LSTM, for efficient and interpretable intrusion detection in IoT ecosystem. GA was employed for dual optimization, simultaneous feature selection and hyperparameter tuning, to achieve a balance between accuracy, computational efficiency, and interpretability, distinguishing this work from prior GA-based IDS approaches. The models were evaluated using two benchmark datasets, IoT-ToN and UNSW-NB15, selected for their diverse traffic patterns, heterogeneous feature spaces, and complementary representation of IoT and enterprise network behavior. This combination supports the cross-domain generalizability of the proposed framework. Experimental findings show that GA significantly reduced the feature space while improving runtime and memory efficiency. The GA-GRU model achieved an accuracy of 95.8%, outperforming state-of-the-art IDS models. Both GA-optimized architectures recorded high precision, recall, and F1-scores, with low False Positive Rates (FPR), confirming their robustness in real-world detection. Although GA identified different optimal feature subsets for GRU and LSTM, the recurring key features indicate selection stability across architectures. Moreover, LIME-based interpretability analysis provided transparency into model decision-making, enhancing trust and explainability. Overall, the proposed framework delivers a novel, generalizable, and resource-efficient IDS solution tailored for next-generation IoT environments.

**Keywords** Intrusion detection system (IDS) · Genetic Algorithm (GA) · Feature selection · IoT security · Explainable AI (LIME)

## 1 Introduction

With The integration of a vast number of devices within the Internet of Things (IoT) ecosystem has made it a critical facilitator of digital transformation. Almost every field has slowly become dependent on the IoT for operation. Organizations such as farming, banking, healthcare, industries, and others depend on IoT for their daily operations. As the adoption of IoT continues to expand, so does their exposure to cyber threats. Due to the huge amount of data produced within the IoT ecosystem, the traditional intrusion detection system lacks the capacity to effectively defend it. Secondly, most of these devices have low computational capacity, in terms of memory and power to handle the huge amount of data produced [1]. Lastly, due to the interconnection of diverse devices within the IoT ecosystem, the data produced

✉ Elijah M. Maseno
masenoelijah@gmail.com

Yanxia Sun
ysun@uj.ac.za

Zenghui Wang
wangzengh@gmail.com

[1] Department of Electrical and Electronic Engineering Science, University of Johannesburg, 5 Kingsway Ave, Rossmore, Johannesburg 2092, South Africa

[2] Department of Electrical Engineering, University of South Africa, Johannesburg, South Africa

Published online: 02 February 2026

is heterogeneous in nature, complicating it to handle. Due to the above-listed issue, IoT ecosystems face a wide range of cyber threats.

Due to the unique characteristics of the IoT ecosystem, researchers have proposed the development of lightweight intrusion detection systems [2]. Towards this goal, researchers have proposed the adoption of intelligible IDS models in the IoT ecosystem [1, 3, 4]. To mitigate the security concerns facing the IoT ecosystem, we need efficient and effective intrusion detection systems with the capacity to operate well in resource constrained environments. [1] the authors pointed out that to come up with powerful and yet lightweight IDS, researchers have to adopt techniques such as feature selection and integration of machine learning algorithms for optimization. Anomaly detection using machine learning has shown significant promise in identifying abnormal patterns that may indicate security breaches. Researchers have adopted the use of deep learning approaches in the field of intrusion detection systems with enormous success. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have been widely used in this area with positive reports. These models have the capacity to model sequential data and uncover temporal dependencies in network traffic [3, 5]. However, the effectiveness of these models depends heavily on the proper configuration of hyperparameters and the selection of relevant features, tasks that are complex, time-consuming, and often performed manually. Inefficient configuration may lead to suboptimal performance, increased training time, and poor generalizability to unseen threats.

Genetic Algorithms (GA), an evolutionary optimization method derived from natural selection, with the ability to effectively navigates the search space and discover superior configurations was proposed in this study, to address the limitation of LSTM and GRU outlined previously in this work. GA, due to its superior capability it has been adopted in many optimization research over the decades [6, 7]. This technique has the potential of improving the efficacy and effectiveness of deep learning-based IDS models, rendering them more appropriate for practical use. GA reduces the input dimensionality, thereby lowering both space and time complexity and improving detection efficiency. In addition, the proposed methodology integrates Local Interpretable Model-Agnostic Explanations (LIME) as a post-hoc explainability layer. This approach facilitates simple and understandable system predictions through discovering and displaying the most significant features influencing each prediction. This upgrade promotes trust and accountability, which are vital elements for cybersecurity systems deployed in critical infrastructures [8]. This research evaluated the proposed optimized models using two publicly available datasets, namely IoT-ToN, and UNSW-NB15, which provide diverse modern attack scenarios. The researchers resolved the rampant issue of class imbalance inherent in most of the existing datasets through the use of the class weights technique during training. The study assesses detection accuracy while also evaluating memory usage and runtime, offering practical insights into the resource efficiency of the optimized models. The proposed frameworks intend to provide a lightweight yet efficient solution that satisfies the unique requirements of the IoT ecosystem.

Key contributions in this paper are:

1. GA-Based Optimization Framework: A unified approach for simultaneously optimizing hyperparameters and performing feature selection for both LSTM and GRU models.
2. Explainable Intrusion Detection: Integration of LIME to provide interpretable, feature-level explanations for model predictions.
3. Benchmark Evaluation on IoT-ToN and UNSW-NB15: Validation of the proposed models on real-world IoT and hybrid attack datasets to ensure broad applicability.
4. Handling Data Imbalance: Use of class weights to improve model robustness against minority class underrepresentation.
5. Resource-Efficient Design: Runtime and memory profiling to assess the feasibility of deploying the models in constrained IoT environments.

The remainder of the paper is organized as follows: Sect. 2 reviews related work; Sect. 3 outlines the proposed methodology; Sect. 4 describes the experimental setup; Sect. 5 presents results and analysis; and Sect. 6 concludes the study with recommendations for future research.

## 2 Related work

Researchers have worked for years to design and develop more efficient intrusion detection systems that enhance security in the rapidly growing IoT environment, which faces ever-increasing cyber threats. This study acknowledges the many successes achieved towards this effort. Nevertheless, this study also points out the many challenges facing the existing intrusion detection systems, particularly the inability to handle the large volumes of data produced in the IoT ecosystems, which are heterogenic in nature, making it difficult for the existing IDS to handle such kinds of data. In addition, most of these IoT devices have limited computational resources. To solve these limitations, machine learning (ML) and deep learning (DL) have attracted the attention of many researchers due to their inherent capability to handle complex tasks.

Recent studies have advanced hybrid deep learning approaches for intrusion detection in IoT environments by combining multiple architectures to jointly capture spatial, temporal, and relational characteristics of network traffic. In [9], a hybrid model integrating Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), and Graph Neural Networks (GNN) was proposed for IoT intrusion detection. The CNN extracted spatial representations of network flows, the GRU modeled temporal dependencies to recognize sequential attack behaviors, and the GNN captured relational dependencies among IoT devices. Evaluated on the ToN-IoT dataset, the model achieved high accuracy and strong generalization capability, demonstrating its potential for real-time IoT security applications. However, the absence of a feature selection mechanism limited its computational efficiency. Building on similar principles, [1], developed a lightweight hybrid intrusion detection system integrating CNN and Bidirectional Long Short-Term Memory (BiLSTM) networks for efficient IoT threat detection. The CNN component extracted spatial features, while BiLSTM captured bidirectional temporal patterns. Feature selection was performed using a chi-square method, identifying 20 key attributes from the UNSW-NB15 dataset as most discriminative. The model achieved an accuracy of 98% in both binary and multiclass classification, indicating strong predictive performance, though the authors noted the need for validation across additional datasets to address potential bias. In another related study, [10] proposed a two-tier intrusion detection framework combining Logistic Regression (LR) at the edge and a one-dimensional CNN in the cloud. The study employed the Bowerbird Courtship-Inspired Feature Selection (BBFS) algorithm to iteratively optimize feature subsets, thereby enhancing model accuracy and interpretability. To mitigate class imbalance, the authors incorporated Synthetic Minority Oversampling Technique (SMOTE), Edited Nearest Neighbor (ENN), and Local Outlier Factor (LOF). The framework demonstrated robust performance across multiple metrics, yet its reliance on a single dataset constrained its generalizability, highlighting the need for further cross-dataset evaluation.

In [6], an intrusion detection system was proposed based on Genetic Algorithm (GA)-driven feature selection and an optimized neural network. The GA was employed to extract the most discriminative and informative features, reducing dimensionality and eliminating redundancy. The selected features were then used to train a back-propagation neural network optimized through a hybrid of Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO). This combined approach enhanced detection accuracy, improved training efficiency, and minimized overfitting. The model, evaluated using the NSL-KDD dataset, achieved high accuracy and reduced detection errors; however, the study did not report computational metrics such as processing time, CPU utilization, or memory usage key factors in assessing IDS practicality for IoT environments. A study by [11], employed Ant Colony Optimization (ACO) to tune hyperparameters in a hybrid model combining Graph Convolutional Networks (GCN) and Gated Recurrent Units (GRU). Principal Component Analysis (PCA) was used for dimensionality reduction by identifying principal components with maximal variance, thereby improving model efficiency. The approach was validated using multiple benchmark datasets, including EDGE-IIoTset, CICAPT-IIoT, and WUSTL-IIoT, achieving an overall accuracy of 97%. Despite its promising results, the study did not evaluate computational resource requirements, which are crucial for deployment in resource-constrained IoT systems. In another related study, [7] proposed an Industrial IoT (IIoT) intrusion detection model integrating Artificial Neural Networks (ANN) with GA-based feature optimization. The GA effectively selected salient features from high-dimensional input data, reducing computational complexity while enhancing detection accuracy. Experiments on IoTID20 and IoT-Botnet datasets confirmed strong performance in detecting diverse intrusion types. Nonetheless, the study lacked quantitative reporting on feature reduction rates and provided limited methodological details on how feature importance was measured.

The authors in [8], the authors proposed a transparent and high-accuracy Intrusion Detection System (IDS) based on Long Short-Term Memory (LSTM) networks integrated with the Firefly Algorithm for feature selection. To enhance interpretability and model accountability, they incorporated two Explainable Artificial Intelligence (XAI) techniques Local Interpretable Model-Agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP). These methods quantify the contribution of each input feature to prediction outcomes, offering cybersecurity analysts granular insight into feature relevance and model decision logic. The model was trained and evaluated using two publicly available datasets, NF-BoT-IoT-v2 and IoTID20, achieving robust detection performance. The authors noted that future work should focus on improving the computational efficiency of the Firefly-based feature selection process. [12] introduced XBiDeep, an explainable deep learning-based IDS designed for the Internet of Medical Things (IoMT). The system combines a bidirectional GRU–LSTM hybrid architecture with XAI techniques such as LIME and SHAP to identify critical features contributing to various attack types. XBiDeep was validated using three IoMT-specific datasets CICIoMT2024, IoMT-TrafficData, and ECU-IoHT achieving high accuracy in both balanced and imbalanced data scenarios. The study emphasized the need for developing lightweight, resource-efficient IDS architectures that maintain strong detection capability while optimizing computational performance.

Ref. [13] proposed a Lion Optimization-based intrusion detection system integrating feature selection with deep learning to enhance cyber threat detection. The study employed Lion Optimization for Feature Selection (LOFS) combined with an attention-based bidirectional long short-term memory (ABiLSTM) model, while hyperparameter tuning was performed using the Gorilla Troops Optimizer (GTO). Experimental results demonstrated the promising performance of the proposed NIDS-LOFSDL system. However, the evaluation focused solely on classification metrics, excluding critical performance indicators such as training time, computational complexity, and resource utilization, which merit future investigation. Similarly, [5] developed hybrid intrusion detection models integrating Grey Wolf Optimization (GWO) with deep learning architectures. The initial configuration combined Gated Recurrent Units with GWO (GRU-GWO), while the subsequent variant employed Long Short-Term Memory with GWO (LSTM-GWO). In both cases, GWO effectively performed feature selection, significantly reducing data dimensionality and enhancing detection accuracy while minimizing computational demands. Nonetheless, the evaluation was limited to a single dataset, potentially constraining the generalizability of the results. In related research, [14] proposed a scalable hybrid intrusion detection framework MapReduce-based Black Widow Optimized Convolutional-LSTM (BWO-CONV-LSTM) to address the limitations of existing systems. The model integrates Artificial Bee Colony (ABC) optimization for feature selection and a hybrid CONV-LSTM classifier whose hyperparameters are tuned via the Black Widow Optimization (BWO) algorithm. Implemented within a MapReduce environment, the framework efficiently processes large-scale network traffic data. Experimental validation across four benchmark datasets NSL-KDD, ISCX-IDS, UNSW-NB15, and CSE-CIC-IDS2018 demonstrated superior performance in terms of accuracy, false positive reduction, computational cost, and classification efficiency compared to existing methods. However, the study did not report quantitative metrics related to memory consumption (e.g., RAM utilization).

Ref. [15] investigated hyperparameter optimization of one-dimensional convolutional neural networks (1D-CNNs) for network intrusion detection using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to tune nine parameters. Both optimization methods, evaluated on the UNSW-NB15, CIC-IDS2017, and NSL-KDD datasets, significantly outperformed the baseline model in terms of accuracy, precision, recall, and F1-score. On the UNSW-NB15 dataset, GA and PSO achieved notably high accuracy; however, the study acknowledged that these techniques introduced substantial computational overhead during hyperparameter tuning, potentially limiting their applicability in resource-constrained environments. In related research, [16] proposed IoT-Defender, an edge-based intrusion detection framework for IoT networks employing a Modified Genetic Algorithm (MGA) for feature selection and an optimized LSTM classifier. The framework was evaluated using BoT-IoT, UNSW-NB15, and N-BaIoT datasets, demonstrating superior detection performance, particularly on BoT-IoT. Nonetheless, the study did not include quantitative measurements of memory utilization, an omission that limits insight into the system's scalability and efficiency. In their work, [17] developed a PSO-GA optimized ResNet-BiGRU model that integrates ResNet for local feature extraction, BiGRU for capturing temporal dependencies, and an attention mechanism for refined feature representation. The hybrid PSO-GA optimization enhanced convergence and generalization, leading to improved detection accuracy across all metrics on the NSL-KDD dataset. The model also achieved robust performance on KDD99, UNSW-NB15, and CIC-IDS2017 datasets, outperforming conventional methods. However, as in prior studies, the authors did not report any evaluation of memory requirements. Research by [18] proposed a hybrid metaheuristic–deep learning architecture for anomaly-based intrusion detection in IoT environments. The model employed the Butterfly Optimization Algorithm (BOA) for feature selection to effectively reduce data dimensionality while preserving critical discriminative features. The selected features were used to train an LSTM network, resulting in high detection accuracy for both known and novel attacks. Experimental evaluation using the IoTID20 and ToN-IoT datasets demonstrated the model's robustness and efficacy in safeguarding IoT systems. Despite its superior accuracy, the approach exhibited relatively high computational cost, suggesting a trade-off between performance and efficiency.

## 2.1 Overview of ML algorithms

In this section, an overview of the machine learning methods that were utilized in this work is presented.

(i) **Genetic algorithm**

Genetic algorithms are a form of computational optimization technique that functions on the concept of natural selection and genetics. They are employed to address complex issues by emulating the evolutionary process to iteratively enhance a population of prospective solutions. Employing evolution principles such as selection, crossover, and mutation, it evolves through generations to improve a selected candidate population. In each generation, the individual's fitness is assessed according to a specified objective function, directing the evolutionary process toward optimal solutions. Due to its strength to navigate such complex space, GA has been adopted by many researchers to solve complex tasks

such as feature selection, hyperparameter tuning, and model architecture refinement [19, 20].

Within the realm of deep learning, the GA has surfaced as an effective instrument for optimizing neural network architectures and training parameters [6, 21]. Figure 1 below illustrates a genetic algorithm flow diagram.

The principles of evolution that govern the functioning of genetic algorithms are as follows:

**Selection:** The process of selecting the most suitable individuals from the existing population to serve as parents for the subsequent generation. In this work we employed the tournament selection method.

**Crossover:** The process for producing offspring by combining the genetic material of two different individuals' parents. This work adopted the two-point crossover method to produce offspring.

**Mutation:** This is the process of introducing random alterations to the genetic material of the offspring in order to preserve the diversity of the population. This study adopted the bit-flip mutation technique.

(ii) **Long Short-Term Memory (LSTM)**

Long Short-Term Memory (LSTM) was proposed by [22] to solve the issue of vanishing gradient in traditional recurrent neural networks. To address the vanishing gradient problem, the authors introduced the concept of a memory cell. The memory cell is made of three components, namely input, forget, and output gates, that manage how information is handled within the network. Three completely connected layers utilizing sigmoid activation functions calculate the values for the input ($I_t$), forget ($F_t$), and output gates ($O_t$) as illustrated in Fig. 2 below. Consequently, the sigmoid activation constrains all values of the three gates into a range of (0,1). Furthermore, an input node is necessary, usually calculated using a tanh activation function. The input gate regulates the extent to which the input node's value is incorporated into the current internal state of the memory cell. The forget gate ascertains whether to retain the existing memory value or discard it. The output gate ascertains if the memory cell should affect the output at the present time step.

Mathematically to compute the three parameters:

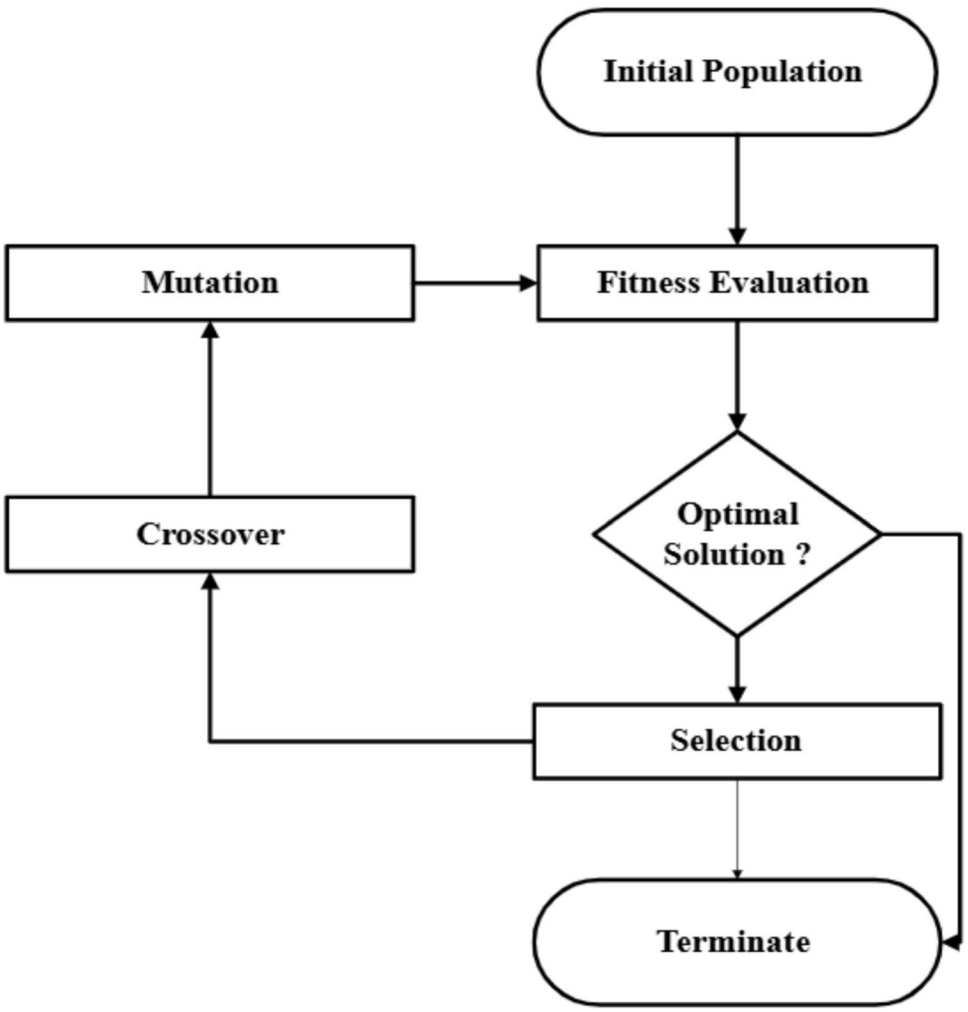


**Fig. 1** Flow diagram of GA

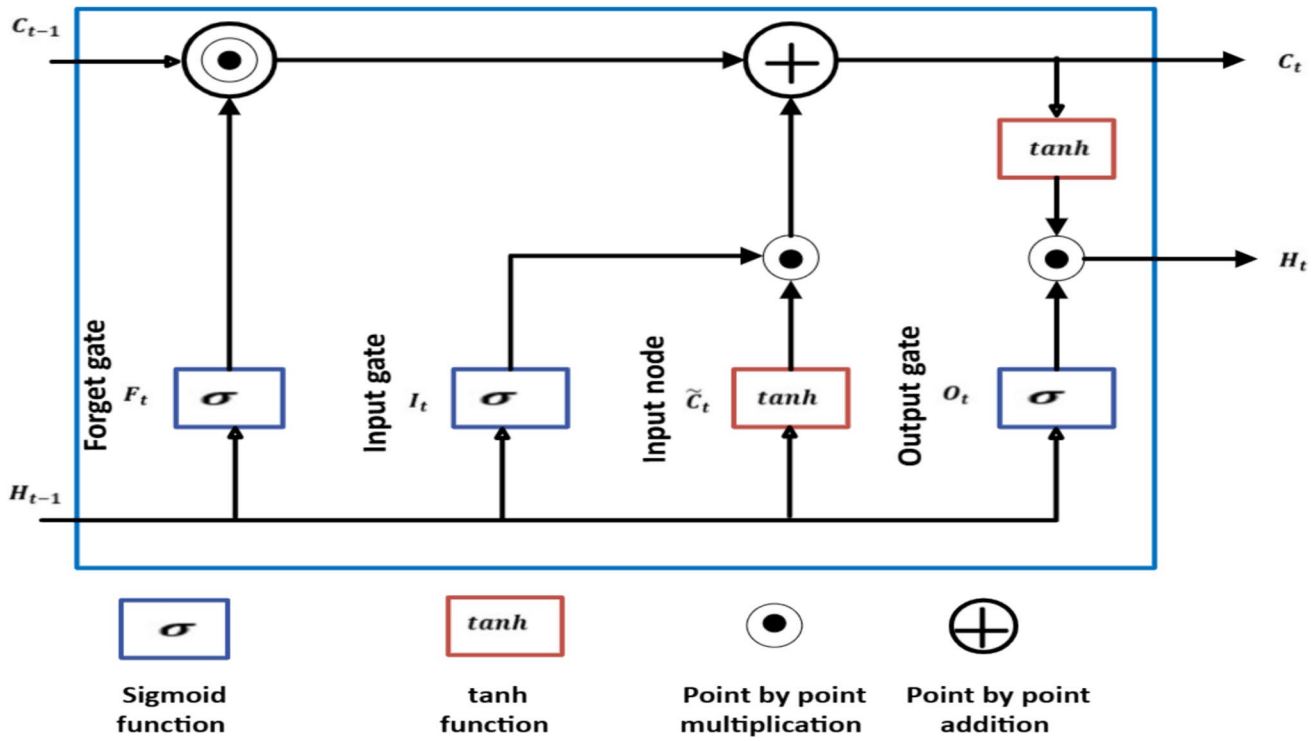


**Fig. 2** LSTM structure

$$\boldsymbol{I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i)} \quad (1)$$

$$\boldsymbol{F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f)} \quad (2)$$

$$\boldsymbol{O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o)} \quad (3)$$

where: $\boldsymbol{W_{xi} W_{xf}}$ $\boldsymbol{W_{xo}}$ and $\boldsymbol{W_{hi}, W_{hf}}$, $\boldsymbol{W_{ho}}$ are all weights. Biases represented by $\boldsymbol{b_i, b_f, b_o}$

For the input node:

$$\boldsymbol{\tilde{C}_t = \sigma(X_t W_{xc} + H_{t-1} W_{hc} + b_c)} \quad (4)$$

where: $\boldsymbol{W_{xc}}$, and $\boldsymbol{W_{hc}}$, are all weights. and $\boldsymbol{b_c}$, bias.

To get the updated memory internal state ($\boldsymbol{C_t}$), we use Eq. 5:

$$\boldsymbol{C_t = F_t . C_{t-1} + I_t . \tilde{C}_t)} \quad (5)$$

where $\boldsymbol{C_{t-1}}$ is the old cell internal state.

Finally, to get the output of the memory cell as seen by other cells (the hidden state: $\boldsymbol{H_t}$), we employ Eq. 6.

$$\boldsymbol{H_t = O_t . tanh(C_t)} \quad (6)$$

Although more computationally intensive than simpler architectures like Gated Recurrent Units (GRUs), LSTMs are proficient at capturing both short- and long-term patterns, making them a valuable in deep learning applications.

(iii) **Gated Recurrent Unit (GRU)**

GRU were designed as to solve the issue of high complexity and computational demands of LSTM [23]. To address this issue GRU collapses the three gates in LSTM to two gates, namely reset gate ($R_t$) and the update gate ($Z_t$). Like LSTM, GRU applies sigmoid activation function to maintain the range of the gates between 0,1. The update gate function is similar forget gate, it regulates the extent of past information retained, whereas the reset gate governs the impact of prior states on the current input. Figure 3 represent the structure of GRU.

Mathematically to compute the two parameters:

$$\boldsymbol{R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r)} \quad (7)$$

$$\boldsymbol{Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z)} \quad (8)$$

where: $\boldsymbol{W_{xr} W_{xz}}$, and $\boldsymbol{W_{hr}, W_{hz}}$, are all weights. Biases represented by $\boldsymbol{b_r, b_z}$.

$$\boldsymbol{H_t^{'} = tanh\,(X_t W_{xh} + (R_t . H_{t-1})\, W_{hh} + b_h} \quad (9)$$

where: $\boldsymbol{W_{xh}}$, and $\boldsymbol{W_{hh}}$, are all weights. Bias represented by $\boldsymbol{b_h}$.

To gate the hidden state, we use Eq. 10 below:

$$\boldsymbol{H_t = (1 - Z_t)\,.H_{t-1} + Z_t . H_t^{'}} \quad (10)$$

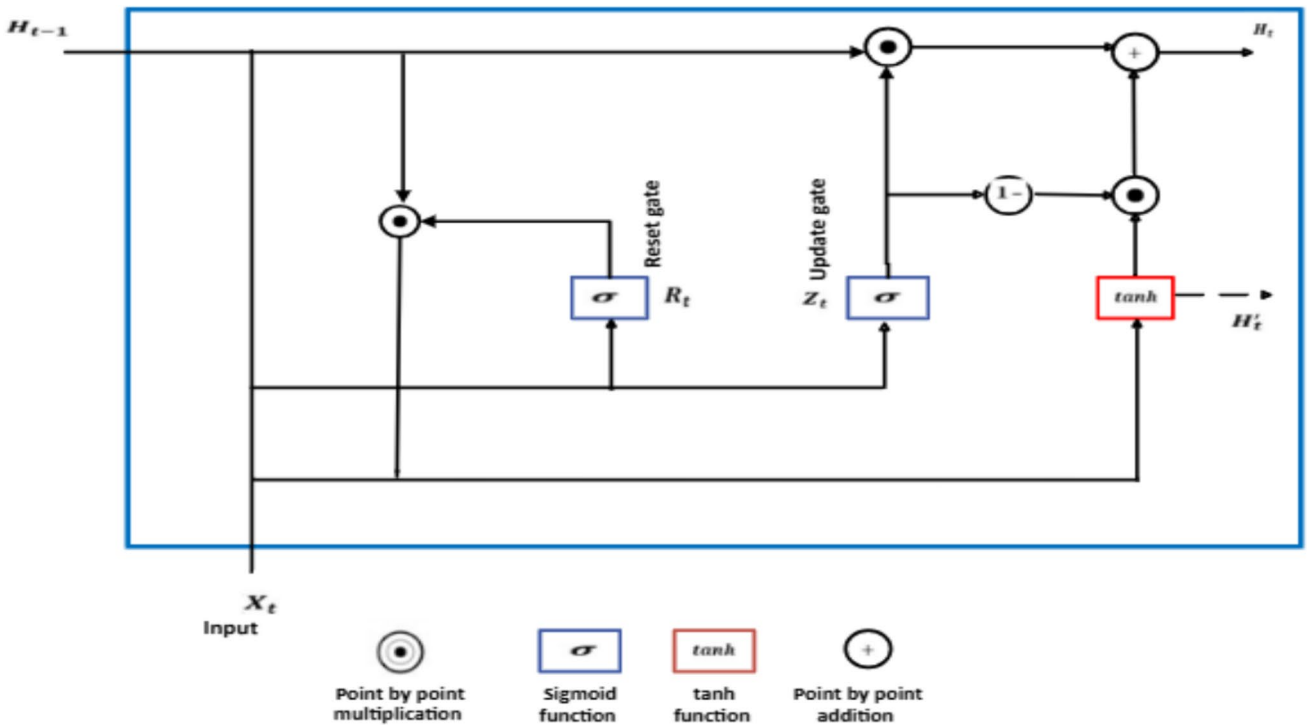


**Fig. 3** GRU structure

## 2.2 Explainable AI (XAI)

Despite the extensive integration of AI across various fields, they have continued to function as black boxes. This problem has diminished the reliability of most the AI models, hence constraining their implementation in crucial sectors such as healthcare. To address this issue, Explainable artificial intelligence (XAI) methodologies have arisen to transform the opaque nature of machine learning (ML) models into a more comprehensible format. These strategies facilitate the explanation of model functionality, aiming to enhance the transparency of machine learning models and boost end-user trust in their outputs [24–26]. Ref. [24] pointed out that SHapley Additive exPlanations (SHAP) and Local Interpretable Model Agnostic Explanation (LIME) are the two prominent methods in explainable artificial intelligence (XAI), especially applicable to tabular data. SHAP was introduced in 2017 by [26]. The operation of SHAP is based on game theory. It tries to explain any model by seeing each feature or combination of features as a participant and the model outcome as the reward. It allocates important value to each feature for a specific prediction. This study focused on the application of LIME due to its low computing time compared to SHAP [24]. LIME was proposed in 2016 by [25]. LIME seeks to clarify the functioning of the model in relation to a particular instance within it. To achieve this, LIME functions by generating surrogate models. This generated model is trained to as an estimate prediction of the entire black box model. The trained model reflects machine learning predictions in a localized manner, rather than a global one.

Mathematically the operation of LIME can be explained as follows:

$$\widehat{\boldsymbol{f}}\,(\boldsymbol{x}) = \boldsymbol{arg}\underset{\boldsymbol{g}\epsilon\boldsymbol{G}}{\boldsymbol{min}}\boldsymbol{L}(\boldsymbol{f},\boldsymbol{g},\boldsymbol{\pi}_{\boldsymbol{x}}) + \boldsymbol{\Omega}(\boldsymbol{g}) \tag{11}$$

where: $\boldsymbol{f}$: the original complex model, $\boldsymbol{x}$: the instance being explained, $\widehat{\boldsymbol{f}}\,(\boldsymbol{x})$: the local surrogate (interpretable) model around $\boldsymbol{x}$, $\boldsymbol{g}\epsilon\boldsymbol{G}$: a set of interpretable models, $\boldsymbol{L}(\boldsymbol{f},\boldsymbol{g},\boldsymbol{\pi}_{\boldsymbol{x}})$: a loss function measuring how close $\boldsymbol{g}$ approximates $\boldsymbol{f}$ around $\boldsymbol{x}$, weighted by proximity $\boldsymbol{\pi}_{\boldsymbol{x}}$, $\boldsymbol{\pi}_{\boldsymbol{x}}$ : a proximity measures that weights instances by their closeness to $\boldsymbol{x}$, $\boldsymbol{\Omega}(\boldsymbol{g})$: a complexity penalty to keep the explanation simple.

In summary, this work acknowledge that a lot of progress has been made in improving the performance of existing intrusion systems. However, there are still gaps which need to be solved. This work proposed a GA pipeline for feature selection and hyperparameter tuning. In addition, this work also incorporated the class balancing, and explainability into a single framework. The main objective of this design is to address the issue of computational constraints and demands of IoT ecosystems. This paper fills in this gap by integrating XAI component on GA-optimized LSTM/GRU IDS model. The model was evaluated using IoT-ToN and UNSW-NB15 datasets.

## 3 Methodology

The proposed optimized intrusion detection system consists of four stages: data preprocessing/cleaning, feature selection/hyperparameter optimization, evaluation, and explainability with LIME as shown in Fig. 4.

**Motivation for Using GA-Optimized GRU/LSTM Approach** The motivation for this work is as follows:

- Deep learning models like GRU and LSTM are highly sensitive to hyperparameter choices Manually tuning these parameters is time-consuming, computationally expensive, and often leads to suboptimal configurations. GA automates this process, efficiently exploring the hyperparameter space.
- Intrusion detection datasets such as IoT-ToN and UNSW-NB15 contain many features, some of which are irrelevant or redundant. Feature selection via GA reduces dimensionality, improving generalization, speeding up training, and lowering memory usage.
- IoT devices operate under tight resource constraints (limited CPU, memory, and power). Optimizing model size and complexity using GA ensures that the IDS remains lightweight and deployable in real-world IoT environments.
- Joint optimization of both feature subsets and model parameters using GA enhances detection accuracy and robustness.
- LIME provides interpretable decision logic, increasing trust in automated IDS systems.

### 3.1 Datasets

This study employed two types of dataset to evaluate the proposed models, namely: IoT-ToN (Internet of Things—Ton_IoT Network) and UNSWNB15 datasets. These are

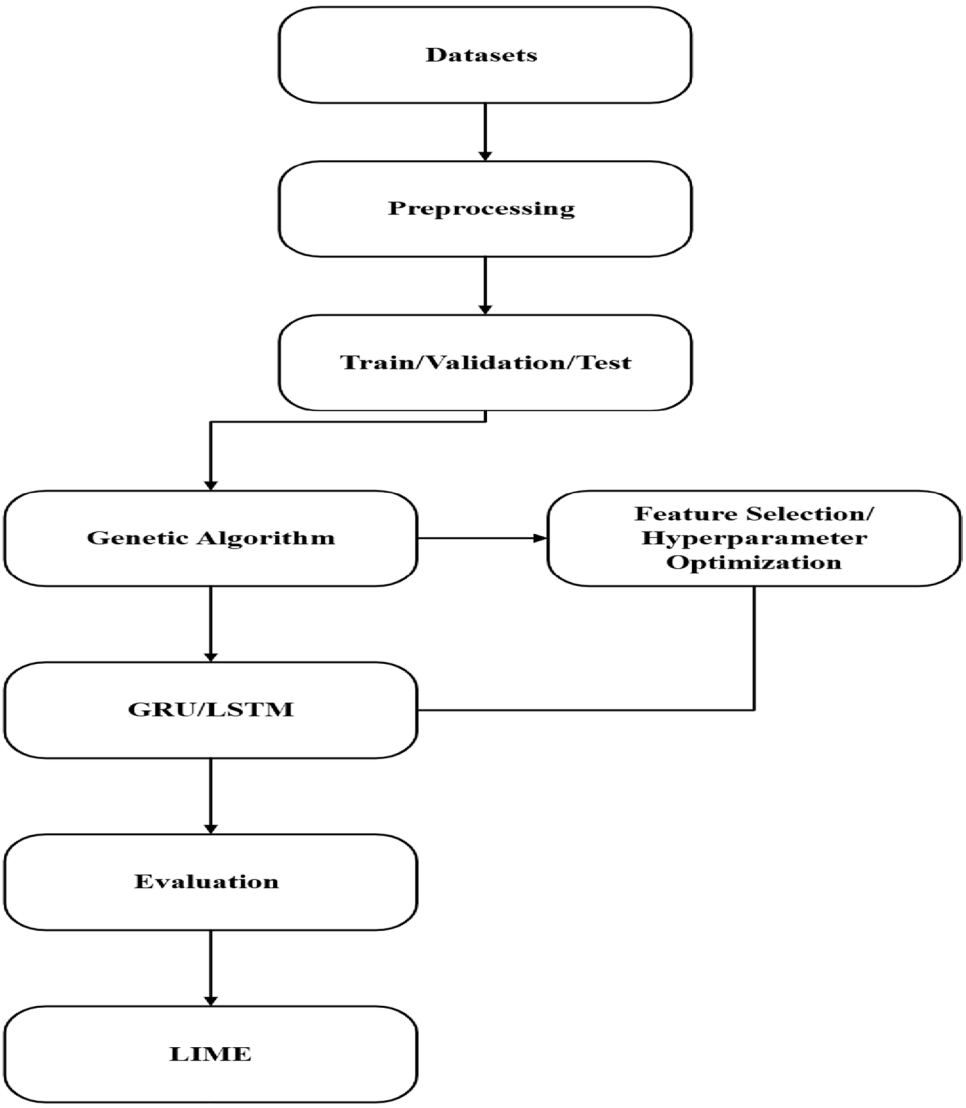


**Fig. 4** The proposed model

publicly available datasets and widely used for training and testing intrusion detection systems.

(i) **IoT_ToN network dataset**

This dataset was introduced by [27] in his published work. The aim of his work was to generate a heterogeneous dataset which could be used to train machine learning algorism and other AI related models. This model captures the complex ecosystem of IoT, which makes it unique compared to other existing datasets. This characteristic makes it suitable for evaluation of IoT related intrusion detection systems. This dataset It emulates genuine network traffic by reproducing both benign and malicious behaviors across a varied spectrum of IoT devices and communication protocols. According to the author the dataset contains nine types of attacks, namely, Scanning attack, Denial of Service (DoS) attack, Distributed Denial of Service (DDoS) attack, Ransomware attack, Backdoor attack, Injection attack, Cross-site Scripting (XSS) attack, Password cracking attack, and Man-In-The-Middle (MITM) attack. In addition, the dataset contains 44 attributes categorized into four classes with information such has connection, statistics, user attributes (i.e., DNS, HTTP and SSL activities), and violation attributes. These attributes make IoT_ToN network dataset suitable for evaluation of IoT related cybersecurity researches.

(ii) **UNSWNB15 dataset**

The UNSW-NB15 dataset was introduced by [28]. This dataset was created to solve the limitation of KDD98, KDDCUP99 and NSLKDD datasets. The authors pointed that most of these datasets did not reflect the current network environment hence limited in the evaluation of developed intrusion detection systems. This dataset has been widely used for research in network intrusion detection. According to the authors, this dataset has 49 features extracted from Argus and Bro. One of the main limitations of this dataset is that it does not capture IoT related cyber-attacks (Moustafa, 2021). However, its widespread use and fair distribution position it as a reliable benchmark for assessing intrusion detection techniques in both traditional and IoT contexts.

### 3.2 Data preprocessing phase

Data preprocessing phase is very crucial phase in the design and development of machine learning models. The raw data must be converted into a usable format before training and validation of models. This process starts with the identification of any inconsistences such as, missing values, columns with constant values, duplicate entries and others within the dataset. This is followed with the application of various techniques to solve the identified issues. After data cleaning has been done what follows is the transformation process. This process may include techniques such as feature scaling, encoding of categorical variables, data discretization. This a one of the most important process in machine learning pipeline which makes sure the original dataset in prepared into a more suitable format before model training starts.

### 3.3 Feature selection

Feature selection is technique applied in machine learning filed to deal with issue of data dimensionality. Though huge datasets are necessary in model training but irrelevant features can affect model performance and lead to high computational demands. This phase is important to reduce the complexity of the dataset by eliminating the unnecessary feature and keeping the most optimal features. Research has shown that this method has the potential of increasing the accuracy of a model, shorten the training time, as well as reducing the computational demands [29–34]. Feature selection techniques can be classified into different categories, namely: Filter Methods, Wrapper Methods, Embedded Methods, Greedy Feature Selection Algorithms, Recursive Feature Elimination with Cross-Validation (RFECV) and Sequential Feature Selection Methods [35]. This study focused on Wrapper Method due to its effectiveness in feature selection [36]. Wrapper approaches train the machine learning algorithm using different subsets of features, incrementally adding or removing features and evaluating the outcomes at each iteration [37]. All wrapper approaches aim to identify the feature set that maximizes model performance. Genetic Algorithm (GA), is a type of wrapper feature selection method. To perform feature selection, GA represents each feature subset as a chromosome, and its efficacy is assessed by a fitness function associated with a machine learning model. Through several iteration of evolution process (selection, crossover, and mutation), GA explores the feature space to discover the optimized feature subset lowering both space and time complexity thereby optimizing the prediction. lowering both space and time complexity and improving detection efficiency. This method has the potential to improve model's accuracy, precision, and reducing the computational demands of intrusion detection models.

### 3.4 Hyperparameter optimization

Hyperparameter optimization it is a process of searching the best combination of the hyperparameters to optimize machine learning algorithms. This process can be done either using manual search or employing automation search. In neural networks the researchers can automate the process to search the optimal combination of hyperparameters such as, learning rate, quantity of hidden units, dropout rate, and batch size. The automation process can be broadly classified into grid search, random search, and metaheuristic methods. Genetic Algorithms (GA), adopted in this work falls under

the category of metaheuristic methods. These methods seek to explore the hyperparameter space effectively to identify the combination that produces optimal performance. The optimal performance of any machine learning algorithm highly depends on choice of hyperparameter. This work explored the optimization GRU and LSTM using GA. GA is employed to automatically search for optimal combinations of hyperparameters to enhance the performance of the models. The hyperparameters are represented as chromosome (solutions), which corresponds to a unique configuration of the model. The initial population of the chromosomes is generated randomly. Using fitness function the chromosomes are evaluated to determine which chromosomes will be selected for reproduction. For reproduction, crossover and mutation techniques are applied. The new generation replaces the old population, and the process is repeated for a predefined number of generations. The chromosome with the best fitness score at the end of the GA process is used to configure the final GRU or LSTM model.

### 3.5 Optimized models

This work proposes a methodology that integrates deep learning with evolutionary optimization to improve intrusion detection in IoT environments. Two recurrent neural network models, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are constructed to discern temporal patterns in network traffic data. The optimization process follows the steps below.

BEGIN

1. Load Dataset

2. Preprocess Dataset

3. Initialize Genetic Algorithm (GA) Parameters: Population size, Number of generations, Crossover and mutation probabilities and Define hyperparameter and feature search spaces

4. FOR each individual in GA population DO:

   a. Extract hyperparameters: [units, dropout, lr, epochs, batch_size]

   b. Extract feature mask: [f1, f2, ..., fn] where fi ∈ {0,1}

   c. IF no features selected THEN assign low fitness and CONTINUE

   d. Select features from training/validation sets using mask

   e. Create and compile either LSTM or GRU model using extracted hyperparameters

   f. Train model on selected features (training set)

   g. Evaluate F1-score on validation set

   h. Assign F1-score as fitness value

5. Apply GA Operators: Selection (e.g., tournament), Crossover (two-point), and Mutation (bit-flip on feature mask and mutation on hyperparameters)

6. After final generation, extract best individual: Best hyperparameters and Best selected features

7. Train Final LSTM and GRU Models

8. Apply LIME Explainability

END

### 3.6 Evaluation

The efficacy of the optimized LSTM and GRU models was evaluated on a holdout test set using standard classification metrics, including accuracy, precision, recall, F1-score, and the confusion matrix. These measures offered a thorough evaluation of each model's detection efficacy. The assessment of computational efficiency involved tracking execution time and memory usage during inference using the psutil package. The assessment compared model performance with and without GA-based feature selection to highlight the impact of dimensionality reduction on detection accuracy and resource efficiency.

### 3.7 Model explainability with LIME

In machine learning, model explainability refers to the ability to understand and evaluate the internal decision-making processes of predictive models. This study employed Local Interpretable Model-Agnostic Explanations (LIME) due to its ability to generate precise and interpretable approximations of model behavior for individual predictions. LIME functions by locally altering the input data and developing an interpretable surrogate model to estimate. This methodology is particularly suitable for thorough analysis of feature contributions, aiding security analysts in understanding the reasoning behind classifying a specific traffic event. These insights support administrators in verifying detection

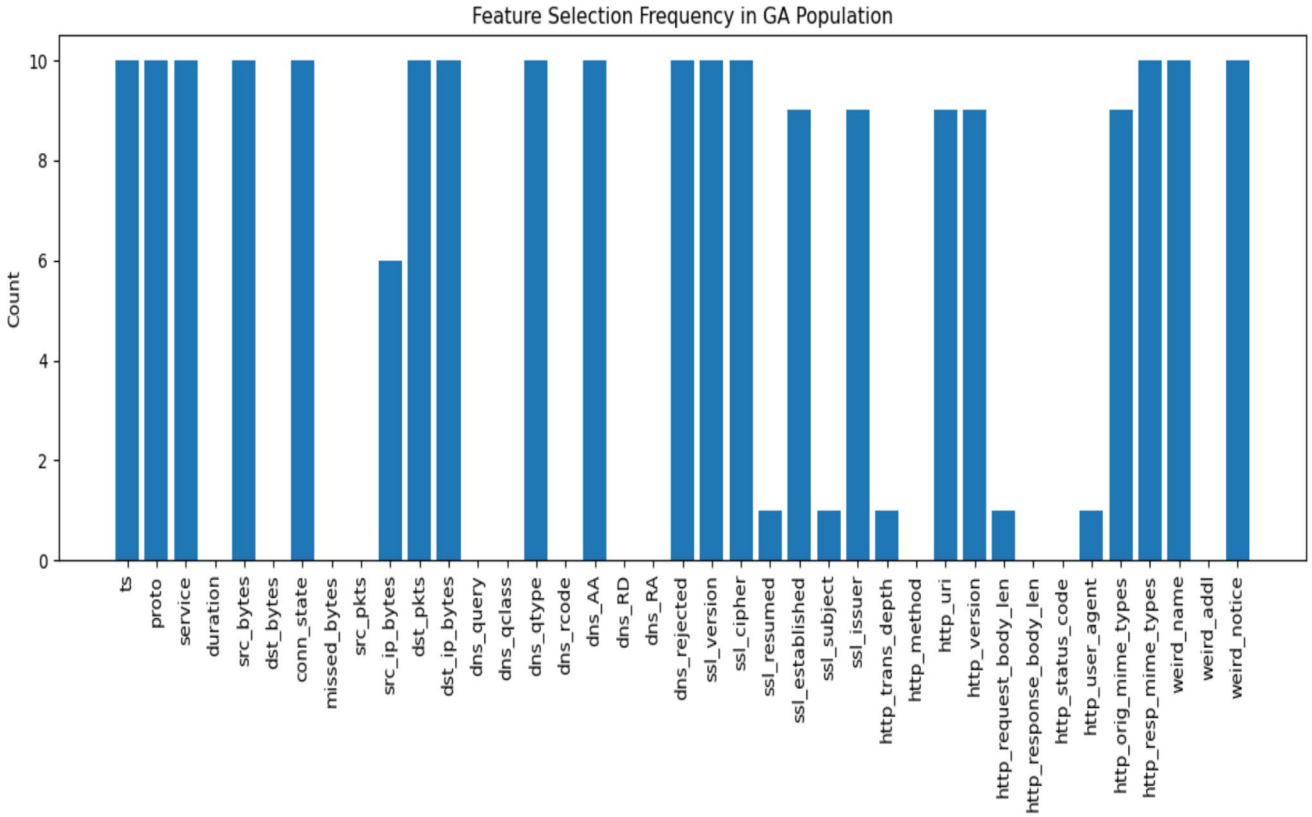


**Fig. 5** GA- LSTM feature selection frequence on the IoT_ToN network dataset

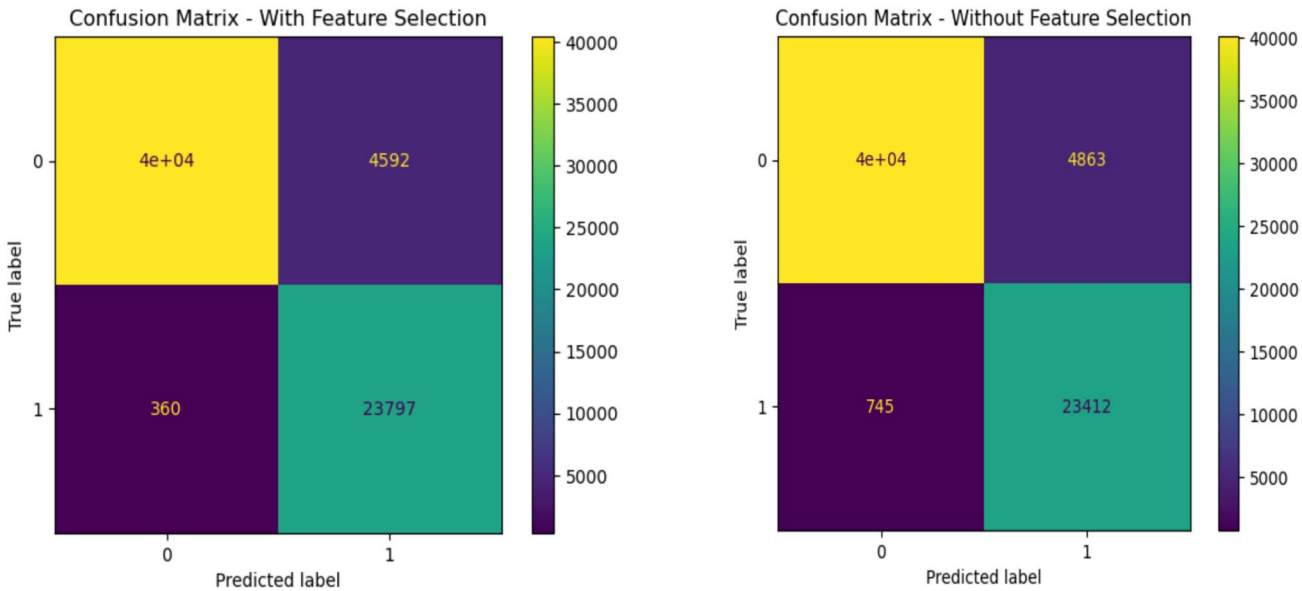


**Fig.6** GA-LSTM confusion matrix on IoT_ToN network dataset

rationale, prioritizing alert triage, and identifying high-risk patterns or misconfigurations.

## 4 Experiments

Two datasets, IoT-ToN and UNSW-NB15, were utilized to assess the efficacy and generalizability of the suggested intrusion detection algorithms. A uniform preprocessing

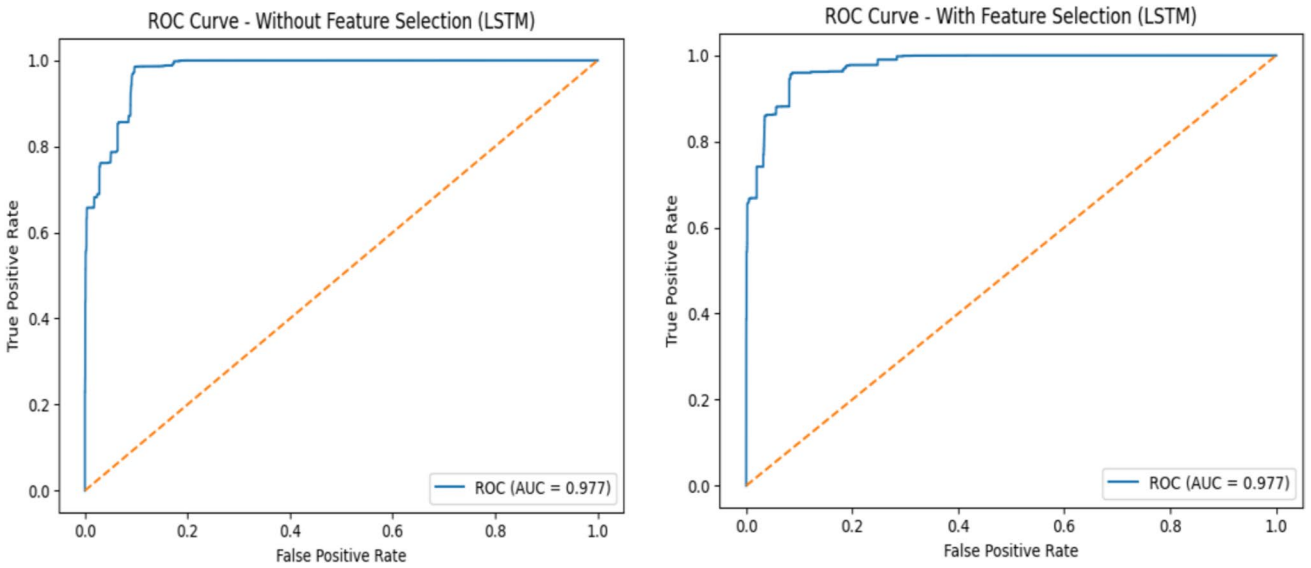


**Fig. 7** GA-LSTM ROC curve on IoT_ToN network dataset

**Table 1** GA-LSTM performance on the IoT_ToN network dataset

| | Precision % | Recall % | F1-score% | Accuracy % |
|---|---|---|---|---|
| Without Feature Selection | 82.8 | 96.9 | 89.3 | 91.88 |
| With Feature Selection | 83.82 | 98.51 | 90.64 | 92.78 |

**Table 2** GA-LSTM computational performance on the IoT_ToN network dataset

| | Time (s) | Memory (MB) |
|---|---|---|
| Without Feature Selection | 57.73 | 41.49 |
| With Feature Selection | 56.53 | 39.05 |

workflow was implemented for both datasets. This involved separating the target label, imputing absent values, encoding categorical variables, implementing Min–Max normalization, and restructuring the data into a 3D structure of (samples, 1, features) appropriate for recurrent neural networks like LSTM and GRU. Numerical variables were imputed utilizing the median technique, and categorical features were populated with the most prevalent category. All categorical variables were encoded with LabelEncoder, and features were standardized to a range of [0, 1]. A stratified sample method was employed to divide the datasets into training (70%), validation (15%), and testing (15%) subsets, maintaining class distribution across the divisions. To rectify the class imbalance present in both datasets, class weights were calculated utilizing the compute_class_weight() method from scikit-learn. Weights were integrated during model training to penalize the misclassification of minority classes, thus enhancing the models' robustness in identifying infrequent attack patterns.

Preprocessing of the IoT-ToN dataset commenced with the removal of whitespace from column headers and the conversion of the type column into binary labels, designating 'Normal' as class 0 and all other attack kinds as class 1. Fields such as IP addresses and ports were excluded, as recommended in [27], to better capture the complexity of security events in the dataset. Following filtering, the dataset preserved 39 features utilized as inputs for the models. Conversely, the UNSW-NB15 dataset was generated by amalgamating the official training and testing subsets. Irrelevant attributes, such as id and attack_cat, were eliminated, and the label column served as the binary classification target. After preprocessing, the refined dataset comprised a total of 47 characteristics. In a manner similar to the IoT-ToN dataset, features were classified, encoded, normalized, and restructured into a 3D tensor with a singular timestep to meet the input specifications of the LSTM and GRU architectures.

## 4.1 Parameter settings

To optimize both hyperparameters and feature selection for the LSTM and GRU models, a Genetic Algorithm (GA) was employed using a unified evolutionary framework. The GA encoded each individual as a vector consisting of five model hyperparameters, number of recurrent units, dropout rate, learning rate, number of training epochs, and batch size, followed by a binary mask representing feature inclusion. The shared hyperparameter search space was defined as follows: the number of units was sampled from 32 to 128; dropout rate ranged from 0.1 to 0.5; learning rate varied between 0.0001 and 0.01; training epochs ranged from 4 to 8; and

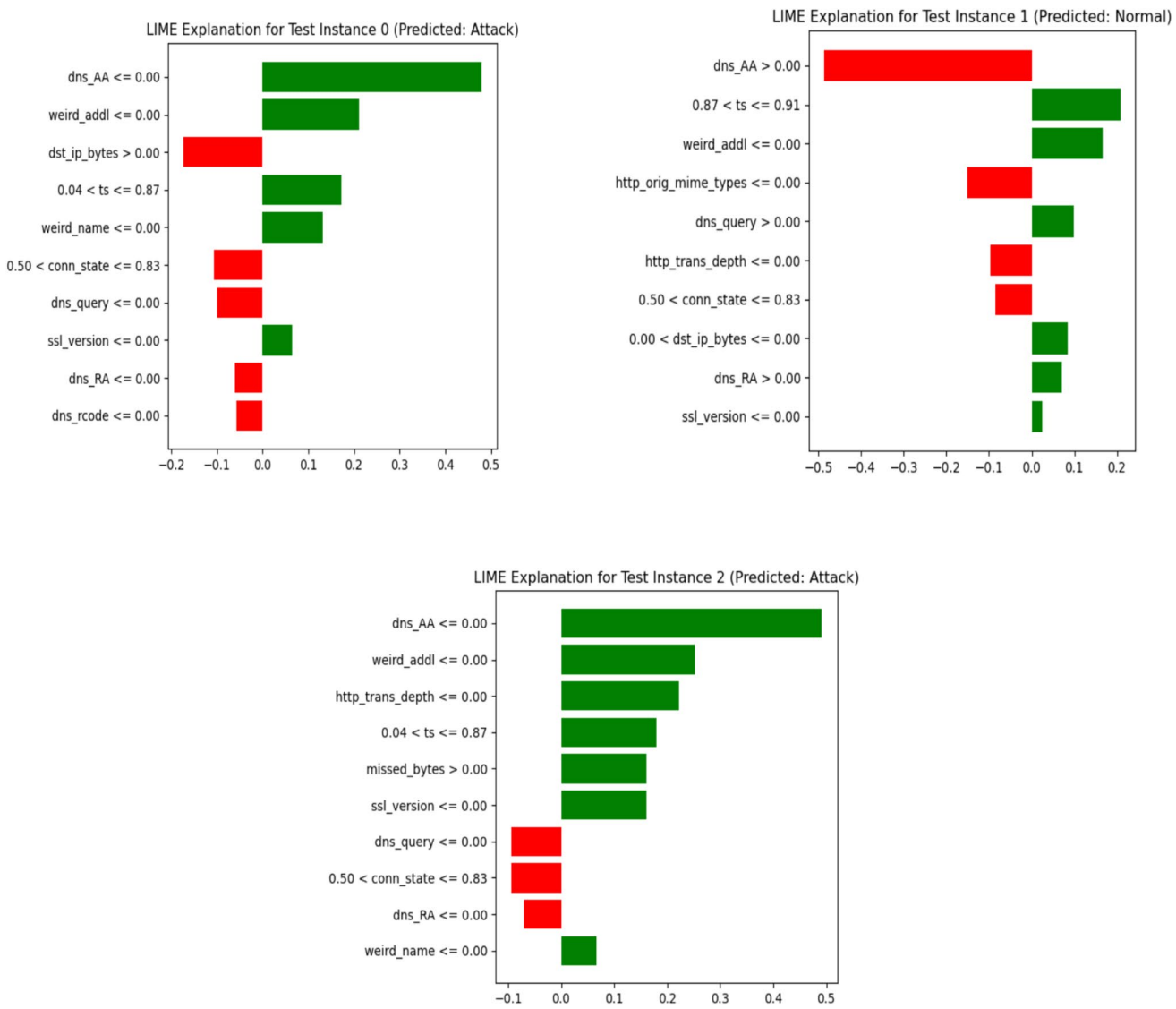


**Fig. 8** LIME outcome using the IoT_ToN network dataset on GA-LSTM

batch size ranged from 32 to 128. The search space for hyperparameters was informed by initial exploratory runs and resource profiling. This bounded range was chosen to ensure convergence while avoiding overfitting and excessive runtime. The length of the binary feature mask matched to the number of features in the dataset, with a value of 1 indicating inclusion and 0 indicating exclusion.

GA parameter setting were as follows, a population size of 10, 5 generations, tournament selection (size = 3), two-point crossover (probability = 0.7), and bit-flip mutation (probability = 0.2). The fitness function was defined as the F1-score computed on the validation set after training the respective model with the candidate configuration.

LIME was initialized using a training dataset limited to the features selected by the Genetic Algorithm, so guaranteeing that the explanations corresponded with the smaller feature space. Essential parameter configurations comprised:

- training_data: A two-dimensional array of test samples containing selected features (dimensions: n_samples × n_selected_features)
- feature_names: List of names corresponding to the selected features
- class_names: ['Normal', 'Attack'], to furnish comprehensible class labels
- mode: 'classification', since the intrusion detection task is binary
- discretize_continuous: True (default), allowing LIME to convert continuous features into discrete bins to enhance local fidelity.

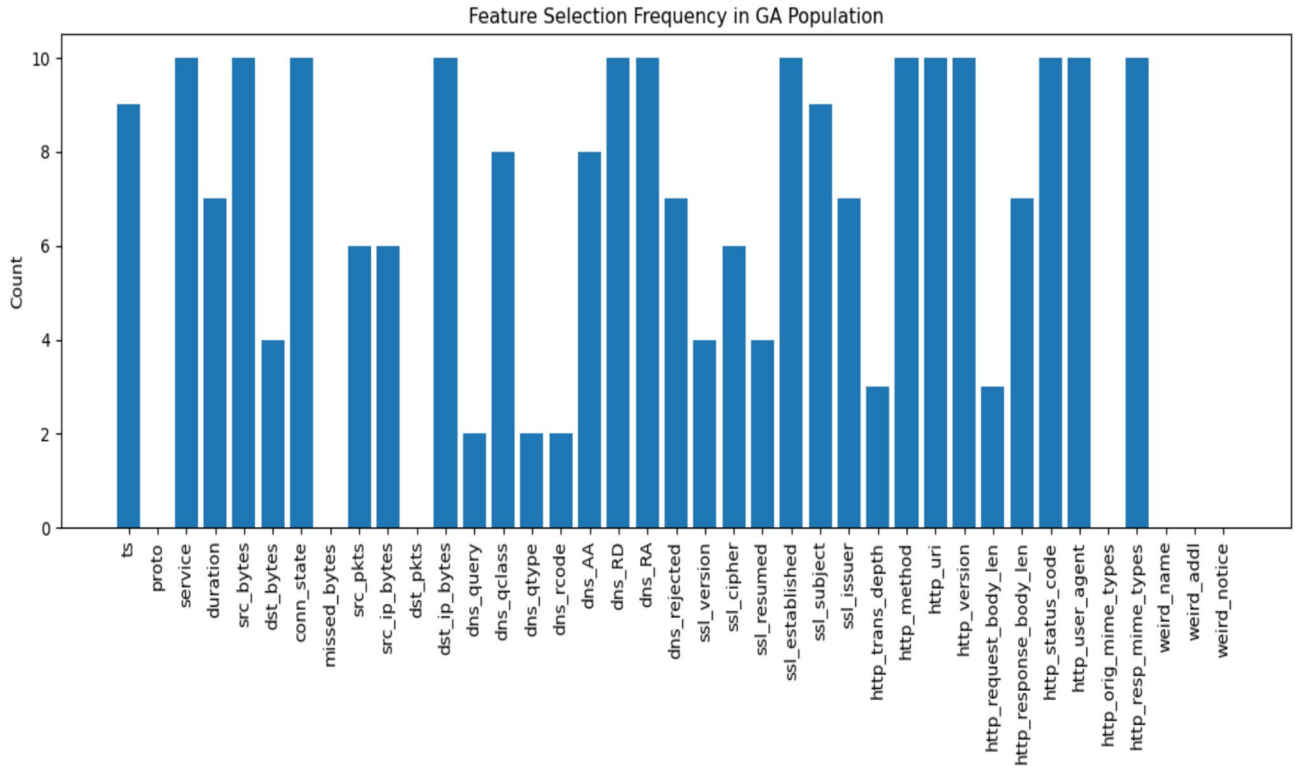


**Fig. 9** GA-GRU feature selection frequence on the IoT_ToN network dataset

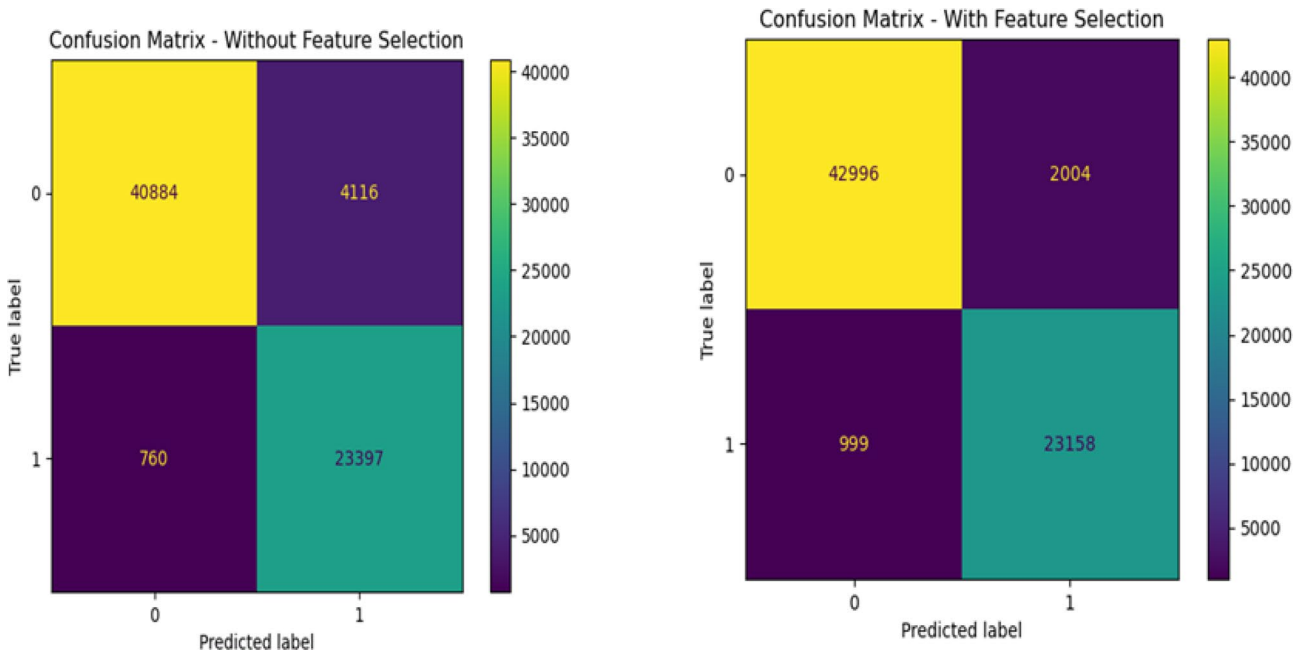


**Fig. 10** GA-GRU confusion matrix on IoT_ToN network dataset

For explanation generation, the following parameters were used:

- num_features: 10, to identify and visualize the top 10 most influential features per instance
- num_samples: 5000 (default), representing the number of perturbed samples generated around the instance for the surrogate model.

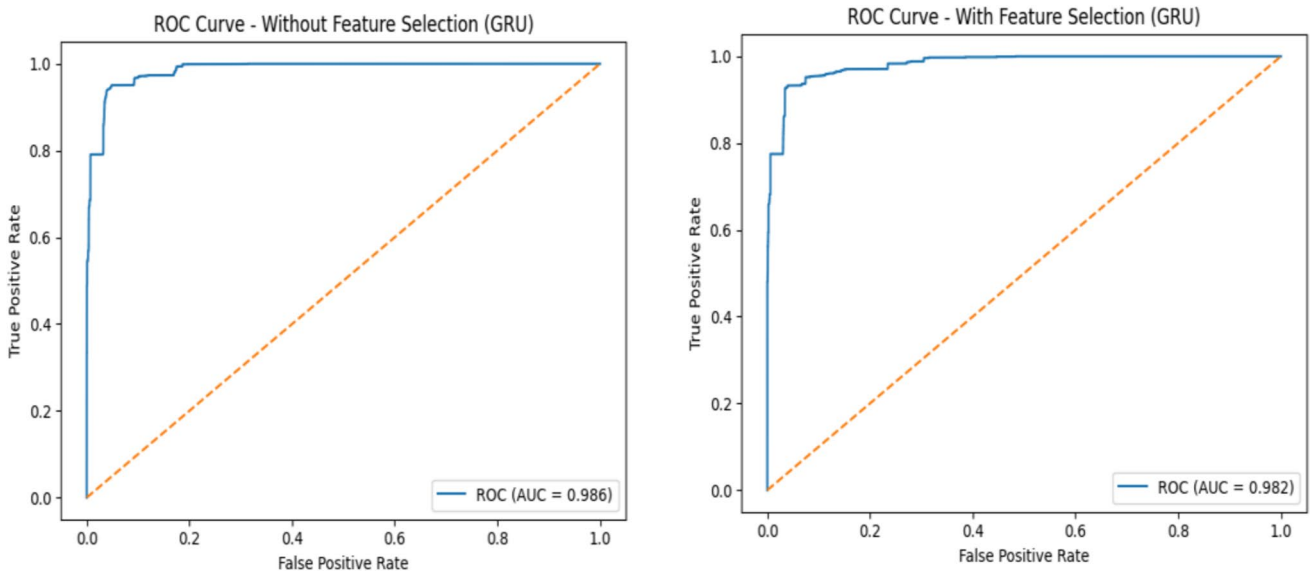


**Fig. 11** GA-GRU ROC curve on the IoT_ToN network dataset

**Table 3** GA-GRU performance on the IoT_ToN network dataset

| | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|
| Without Feature Selection | 85.00 | 96.90 | 93.90 | 93.7 |
| With Feature Selection | 92.00 | 95.90 | 90.80 | 95.8 |

**Table 4** GA-GRU computational performance on the IoT_ToN network dataset

| | Time (s) | Memory (MB) |
|---|---|---|
| Without Feature Selection | 71.19 | 60.34 |
| With Feature Selection | 70.23 | 21.46 |

## 5 Results and discussion

The researchers evaluated the optimized models using classification metrics, such as accuracy, precision, recall, F1-score, and the confusion matrix. In addition, the computational efficiency was evaluated by monitoring execution time and memory utilization. To get a clear picture the evaluation was done before and after feature selection (see Fig. 5).

### 5.1 The IoT_ToN network dataset evaluation

This work first focused on GA-LSTM using the IoT_ToN network dataset. GA reduced the feature set from 39 to 21, lowering both space and time complexity in the proposed approach. Figure 5 describe the frequency in which a particular feature was deemed be important. Figure 6 captures the confusion matrix before and after feature selection. Both models perform relatively well however, without feature selection the model reported a slightly high false positives (4,863) and false negatives (745). This could be an indication of redundancy in the feature subset which can be addressed through feature reduction techniques to improve the precision. The optimized model demonstrated a meaningful balance between detection performance and reliability. Specifically, it achieved a substantial reduction in both false positives (4592) and false negatives (360), leading to corresponding improvements in precision and recall, and consequently, a higher F1-score. This outcome indicates that the GA-based optimization effectively enhanced the model's discriminatory capability, reducing misclassifications of benign traffic as malicious while simultaneously improving its sensitivity to actual attacks. From a cybersecurity perspective, this balance is critical: minimizing false negatives ensures that fewer intrusions go undetected, while controlling false positives reduces alert fatigue among analysts, enhancing the operational usability of the IDS. Therefore, the achieved improvement reflects not only a quantitative gain in detection metrics but also a qualitative enhancement in system trustworthiness and practical deployability within IoT and resource-constrained network environments. Figure 7 below captures the GA-LSTM ROC curve on IoT_ToN network dataset. The ROC curves bend tightly toward the top-left corner, indicating that across a wide range of thresholds the models sustain high true-positive rates at low false-positive rates, far above the random baseline. The indistinguishable AUCs imply that feature selection does not degrade ranking quality; rather, it preserves the model's ability to order attack traffic ahead of normal traffic. Feature selection led to measurable improvements across all metrics, especially in recall and F1-score, suggesting that removing irrelevant or redundant features enhanced the model's detection capability and generalization as shown in Table 1.

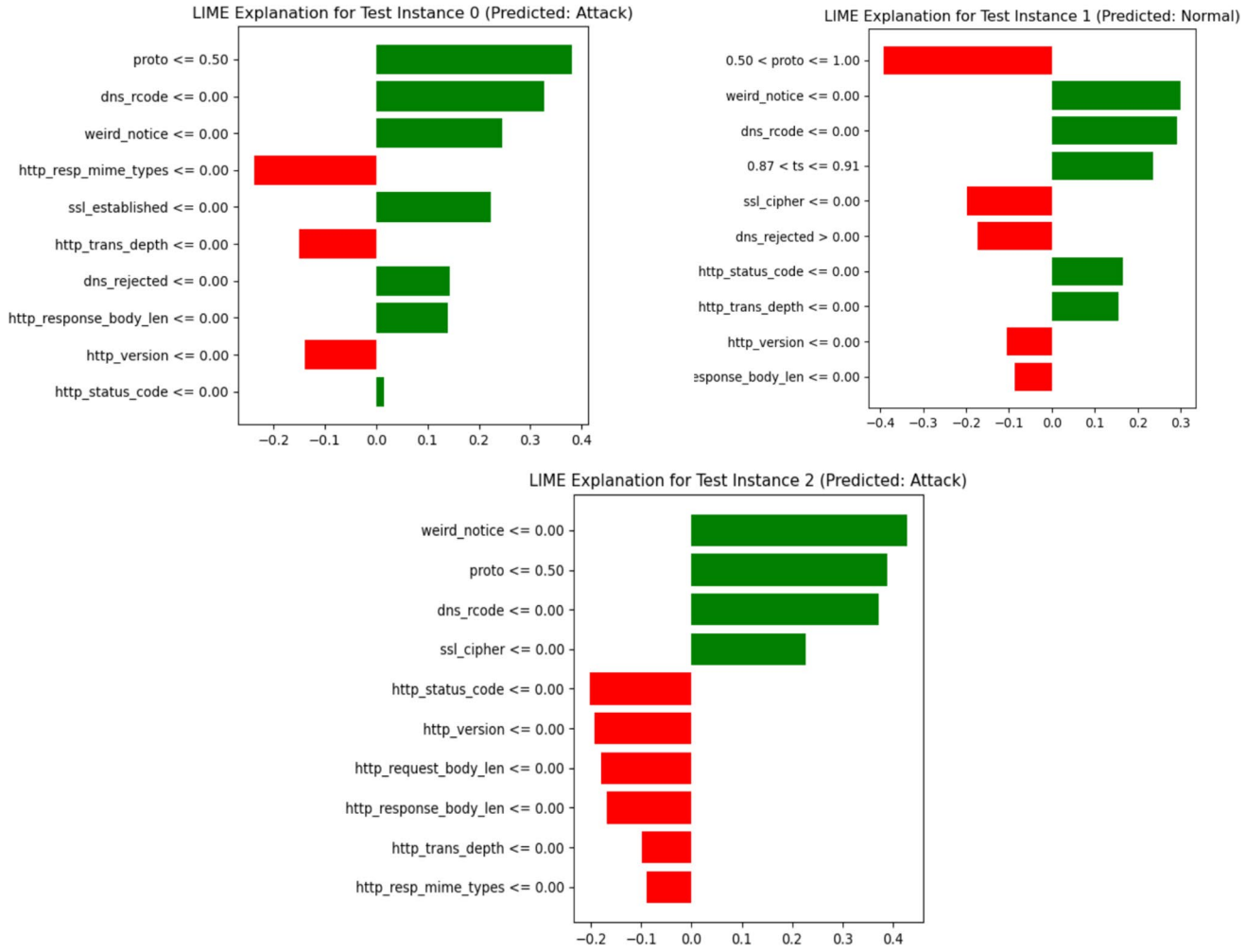


**Fig. 12** LIME outcome using the IoT_ToN network dataset on GA-GRU

Table 2 below shows the execution time and memory utilization of the models before and after feature selection. It is evident from the table that, feature reduction reduced the running time as well as the memory usage. This performance outlines the importance of employing optimization methods in designing efficient and lightweight intrusion detection systems, specifically in a computational constrained environment such as in IoT ecosystems.

Figure 8 below duplicates the LIMA output in three instances in GA-LSTM context. Instances 0 and 2 depicts a likelihood of an attack, according to the analyses seems that the presence of the following features influenced this prediction: dns_AA, weired_name, ssl_version, http_trans_depth and missed_bytes. In instance 1, the negative influence of the following features seems to have influenced a normal prediction of the model: dns_AA, and http_trans_depth.

With GA-GRU optimization using the IoT_ToN network dataset, GA reduced the initial 39 features to 16 features, as shown below in Fig. 9. Figure 10 captures the confusion matrix before and after feature selection. Without feature selection the model reported a high false positive (4,116) but with a law false negative (760). On the other hand, the optimized model performed relatively well, reduced the false positives significantly (2,004) which enhanced the precision. However, the model had a slightly high false negatives (999) affecting the recall. In addition, the model slightly boosts the accuracy confirming the efficiency and effectiveness of the optimized model. Figure 11 below captures the GA-GRU ROC curve on IoT_ToN network dataset. The baseline (all features) attains ROC–AUC = 0.986, while the GA-selected configuration achieves ROC–AUC = 0.982. In both plots, the ROC curves hug the top-left region, showing that across thresholds the classifier sustains high true-positive rates at very low false-positive rates. The small AUC gap (< 0.4 percentage points) indicates that feature selection preserves ranking quality while potentially improving model efficiency (fewer inputs, lower memory/latency) and interpretability.

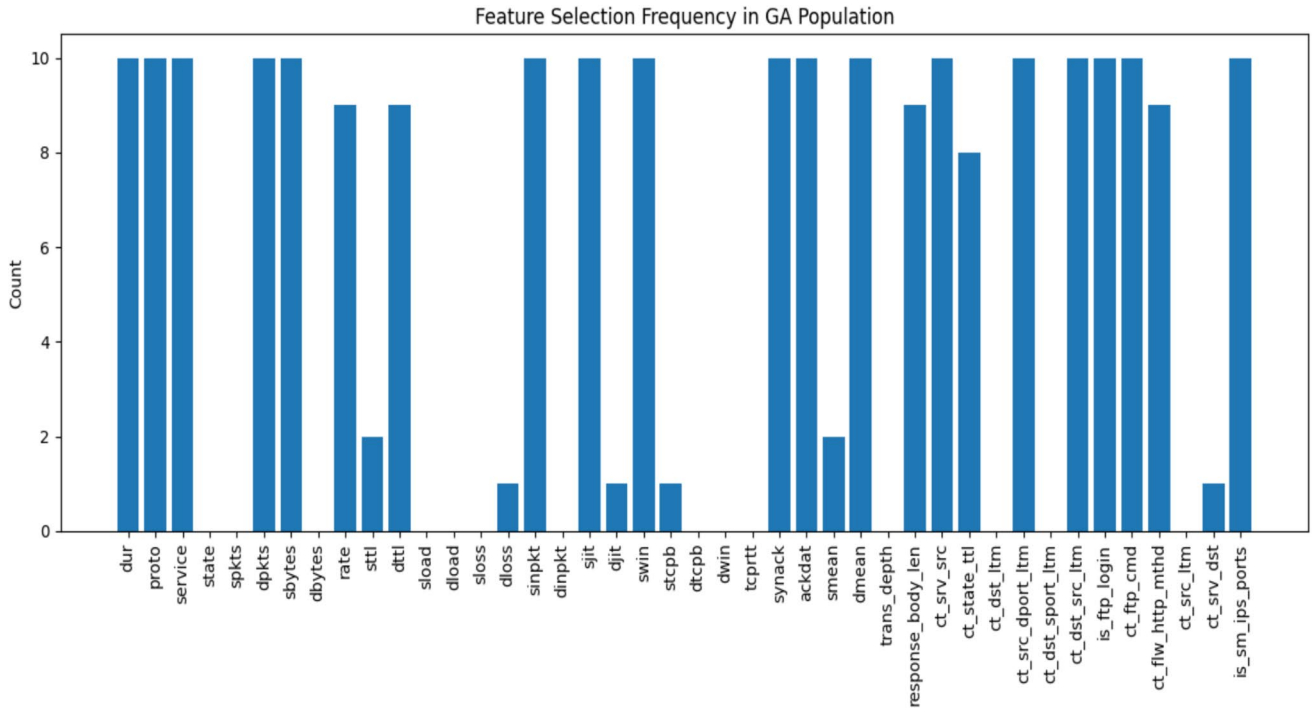


**Fig. 13** GA- LSTM feature selection frequence on the UNSWNB15 dataset

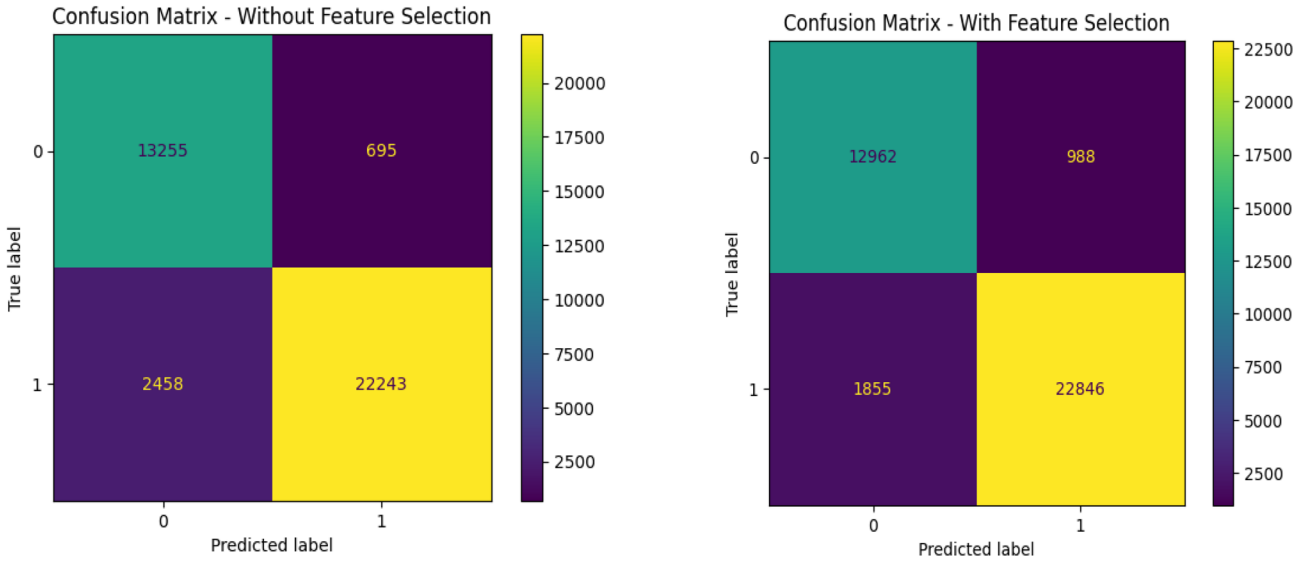


**Fig. 14** GA-LSTM confusion matrix on the UNSWNB15 dataset

In Table 3 below the researchers were able to establish that, feature selection led to a significant improvement in precision and accuracy. Although F1-score and recall decreased slightly, the overall performance gain, due tp reduced false positives. This trade-off is favorable in many real-world intrusion detection scenarios where false alarms are costly.

Table 4 clearly demonstrates that feature removal decreased both operating time and memory use. Feature selection resulted in much more efficient resource usage, especially in terms of memory. While the reduction in execution time was minor, the significant decrease in memory usage highlights the benefit of using feature subset.

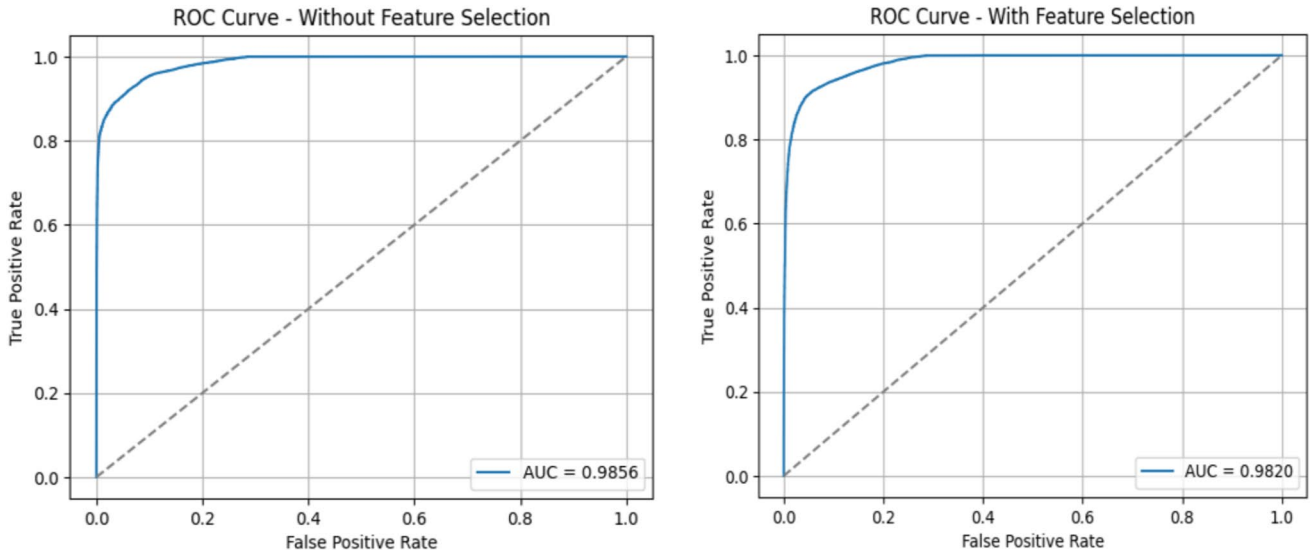


**Fig. 15** GA-LSTM ROC curve on the UNSWNB15 dataset

**Table 5** GA-LSTM performance on the UNSWNB15 dataset

| | Precision % | Recall % | F1-score% | Accuracy% |
|---|---|---|---|---|
| Without Feature Selection | 96.97 | 90.05 | 93.42 | 90.69 |
| With Feature Selection | 95.84 | 92.49 | 94.14 | 90.88 |

**Table 6** GA-LSTM computational performance on UNSWNB15 dataset

| | Time (s) | Memory (MB) |
|---|---|---|
| Without Feature Selection | 44.50 | 22.12 |
| With Feature Selection | 45.80 | 3.62 |

Figure 12 below duplicates the LIMA output in three instances in GA-GRU context. Instances 0 and 2 depicts a likelihood of an attack, according to the analyses seems that the presence of the following features influenced this prediction: proto, ssl_established, dns_rejected, http_responce_body_len and ssl_cipher. In instance 1, the negative influence of the following features seems to have influenced a normal prediction of the model: proto, and dns_rejected.

## 5.2 UNSWNB15 dataset evaluation

This section focused on the evaluation of the optimized models with UNSWNB15 dataset. The first model to be evaluated was GA-LSTM and then GA-GRU followed. GA was able to reduce the initial **42** features to **22** features in the GA-LSTM optimization. Figure 13 describe the frequency in which a particular feature was deemed be important. Figure 14 captures the confusion matrix before and after feature selection. Without feature selection the model reported a low false positive (695) and a high false negative (2458) compared to the model with feature selection. On the other hand, the optimized model had a relatively high false positives (988) but a low false negative (1855). Feature selection led to a more balanced and slightly more effective model. Although precision dropped slightly, the gains in recall, F1-score, and accuracy suggest that the model became better at detecting threats while maintaining high overall correctness. This trade-off is beneficial, especially in intrusion detection, where higher recall and F1-score are critical to minimize missed attacks. Figure 15 illustrates the ROC curves of the GA-GRU model on the UNSW-NB15 dataset with and without feature selection, showing consistently high discriminative performance, with a marginal AUC improvement when feature selection is applied. Table 5 below is a summary of the performance of the model.

The Table 6 clearly demonstrates that feature removal decreased memory use but the model with feature selection had a slightly high running time. While feature selection slightly increased processing time, it resulted in a significant reduction in memory consumption, making the model far more suitable for low-memory environments.

Figure 16 above duplicates the LIMA output in three instances in GA-LSTM context. The first two instances depict a likelihood of a normal flow of traffic, but according to the analyses it is not clear which features influenced this prediction, because most of the features flipped between positive and negative and some features were dropped. In the 0 instance, dttl seems to have a positive influence but in the 1 instance it disappears. On the other hand, swin flipped between positive and negative influence in the two instances. With the 2 instance it clear that the presences of

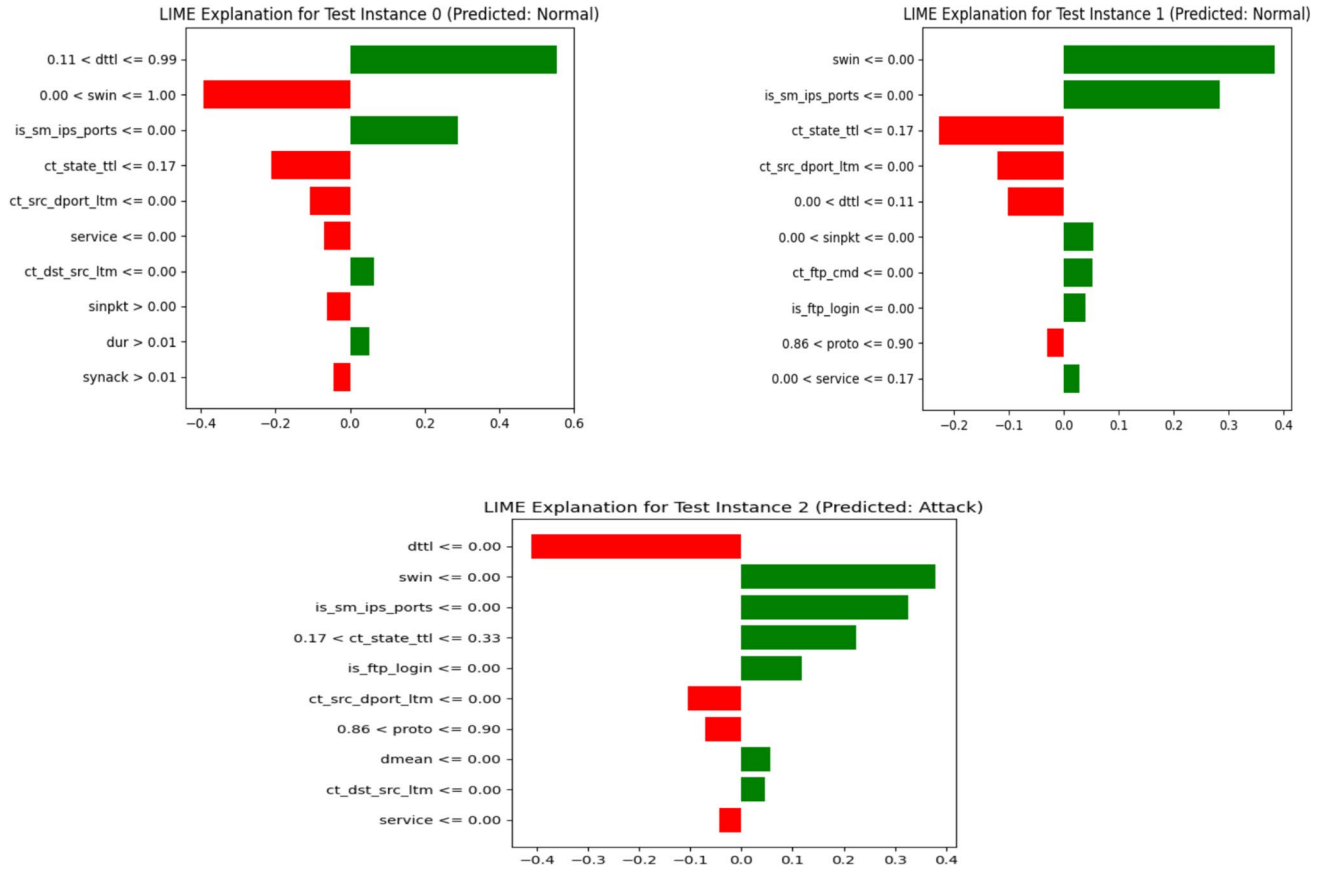


**Fig. 16** LIME outcome using the UNSWNB15 dataset on GA-LSTM

ct_state_ttl and dmean could have influenced the model to an attack prediction.

GA was able to reduce the initial **42** features to **23** features in UNSWNB15 dataset in the GA-GRU optimization. Figure 17 describe the frequency in which a particular feature was deemed be important. Figure 18 captures the confusion matrix before and after feature selection. Without feature selection the model reported a low false positive (485) and a high false negative (2,652) compared to the model with feature selection. On the other hand, the optimized model had a relatively high false positives (1,252) but a law false negative (1,649). Feature selection led to a more balanced and effective model. While there was a minor drop in precision, the gains in recall, F1-score, and accuracy suggest enhanced detection capabilities and improved generalization. This is particularly valuable in security applications, where higher recall reduces the risk of missed attacks. Figure 19 presents the ROC curves of the GA-GRU model on the UNSW-NB15 dataset with and without feature selection, demonstrating strong classification capability in both cases, with feature selection maintaining a high AUC and confirming the robustness of the model's detection performance. Table 7 is a summary of the performance of the model.

The Table 8 clearly demonstrates that feature selection led to a substantial reduction in memory usage with virtually no effect on execution time. This makes the model much more memory-efficient and better suited for deployment in resource-constrained environments, such as IoT devices, without compromising processing performance.

Figure 20 below duplicates the LIMA output in three instances in GA-GRU context. In the first instance depicts a likelihood of an attack, the features that seem to influence this prediction are dttl, sttl, dload, and ct_dst_src_ltm, The other features had a negative influence towards the model. On the second instance, it was a normal prediction. Most of the features had negative influence to the model. Only two features had a positive influence to the model. Last prediction was an attack as shown below. The main features that influenced the model positively were swin, sttl, dload, ct_state_ttl, dwin, and ct_dst_src_ltm. This outcome also affirms the idea of context, in that the mode is influenced by the combination of the features. In this analysis it seems the presence of ct_dst_src_ltm features in fluences the mode towards attack prediction.

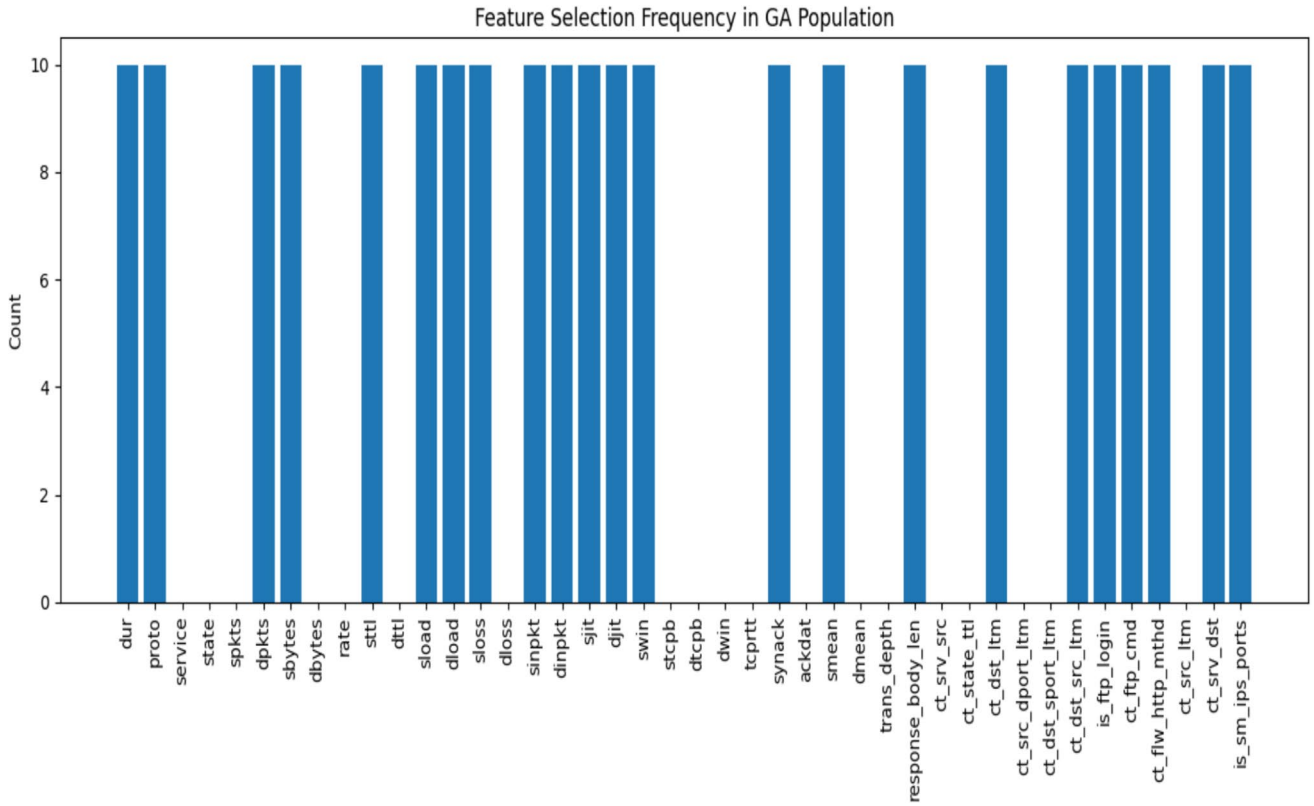


**Fig. 17** GA- GRU feature selection frequency on the UNSWNB15 dataset

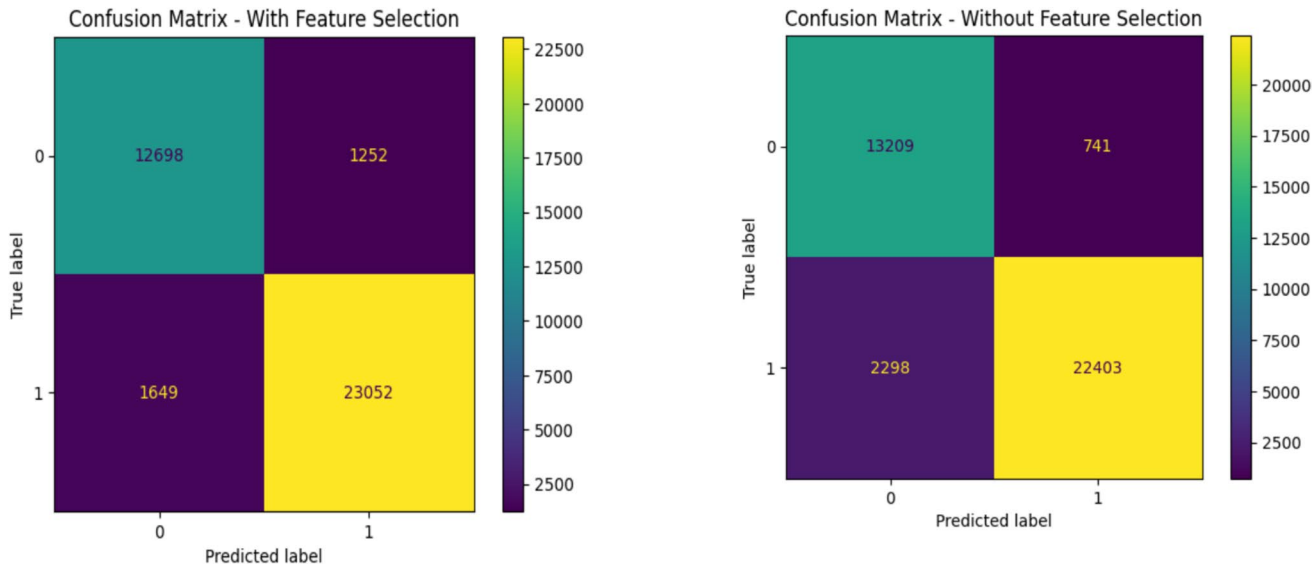


**Fig. 18** GA-GRU confusion matrix on the UNSWNB15 dataset

## 5.3 Qualitative interpretation of feature selection

GA shows a stable preference for features encoding (i) flow magnitude and directionality (sbytes/dbytes, spkts/dpkts, rate, dur), (ii) transport/state integrity (proto, sttl/dttl, ct_state_ttl, synack, ackdat), and (iii) contextual aggregation over recent traffic (ct_* family: per-host/service/port contact counts). These signals align with canonical intrusion behaviors: scans/lateral movement elevate per-destination/service counts; spoofing and man-in-the-middle alter TTL/state patterns; and exfiltration/DoS shift byte/packet balance and rates. Application-layer cues (e.g., ct_flw_http_mthd,

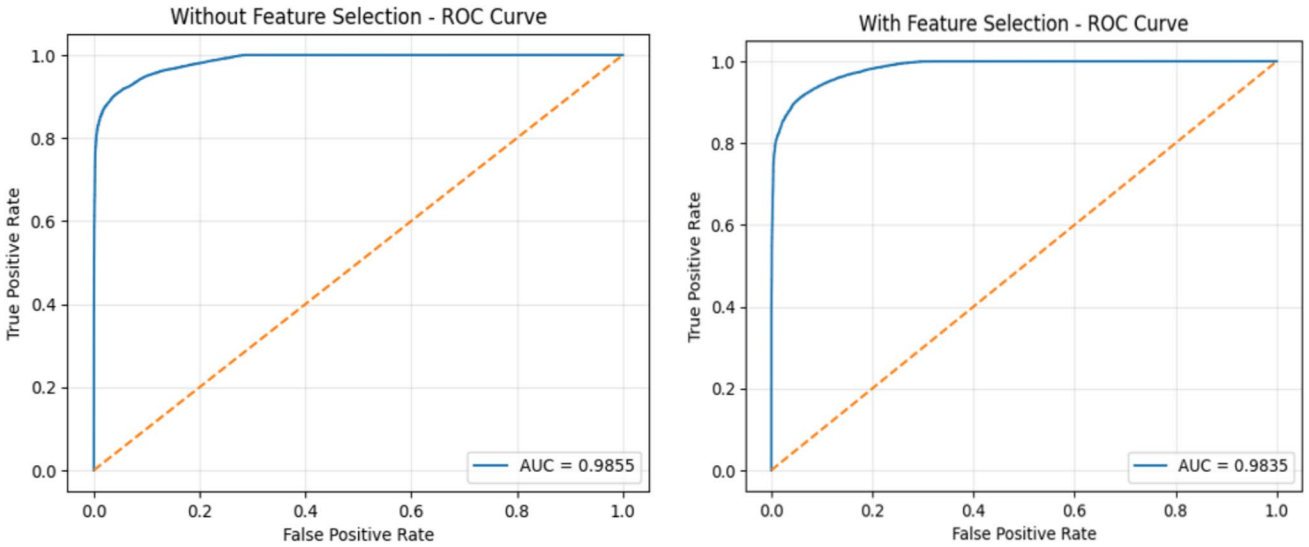


**Fig. 19** GA-GRU ROC curve on the UNSWNB15 dataset

**Table 7** GA-GRU performance on the UNSWNB15 dataset

| | Precision % | Recall % | F1-score% | Accuracy% |
|---|---|---|---|---|
| Without Feature Selection | 97.8 | 89.3 | 93.3 | 92.1 |
| With Feature Selection | 94.8 | 93.3 | 94.0 | 92.5 |

**Table 8** GA-GRU computational performance on UNSWNB15 dataset

| | Time (s) | Memory (MB) |
|---|---|---|
| Without Feature Selection | 31.71 | 54.16 |
| With Feature Selection | 31.98 | 9.6 |

is_ftp_login, ct_ftp_cmd) are selected more situationally, improving detection for protocol-specific attacks. Conversely, jitter, TCP window/base-pointer fields show lower stability, indicating redundancy or sensitivity to noise once the core set is included. This pattern explains the GA's consistent reduction to a compact, high-signal subset and supports the observed performance gains.

To ensure transparency, LIME was used to interpret the GA-optimized models predictions by identifying key features influencing intrusion classification. The analysis showed that attributes such as flow duration, packet size, port entropy, and protocol type had the highest impact in distinguishing benign from malicious traffic consistent with established cybersecurity knowledge. These insights confirm that the model's decisions are logically aligned with real network behaviors, enhancing analyst trust and interpretability. By revealing how individual features contribute to predictions, LIME transforms the IDS into a transparent, human-explainable system, supporting more reliable and accountable intrusion detection.

## 5.4 Comparative evaluation

Table 9 presents a comparative performance analysis between the proposed GA-optimized models (GA-GRU and GA-LSTM) and several existing state-of-the-art intrusion detection systems evaluated on multiple benchmark datasets. The results reveal that the GA-based models consistently outperform prior approaches across key performance metrics, including accuracy, precision, recall, F1-score, and ROC.

For the UNSW-NB15 dataset, traditional models achieved accuracies between 87.53 and 91.05%, with corresponding F1-scores ranging from 87.00 to 90.17%. In contrast, the proposed GA-GRU attained an accuracy of 92.5% and an F1-score of 94.0%, while GA-LSTM achieved 90.88% accuracy and 94.14% F1-score. Similarly, for the IoT-ToN dataset, baseline models reached a maximum accuracy of 88.22%, whereas GA-GRU and GA-LSTM achieved 95.8% and 92.78%, respectively, with ROC values exceeding 97%, signifying strong discriminatory capability.

The improvement observed across diverse datasets demonstrates the effectiveness of Genetic Algorithm optimization in fine-tuning feature subsets and hyperparameters, thereby enhancing model generalization and computational efficiency. Overall, the proposed GA-optimized architectures exhibit robust, scalable, and explainable intrusion detection performance, surpassing existing deep learning-based IDS frameworks in both predictive accuracy and reliability across heterogeneous IoT traffic environments.

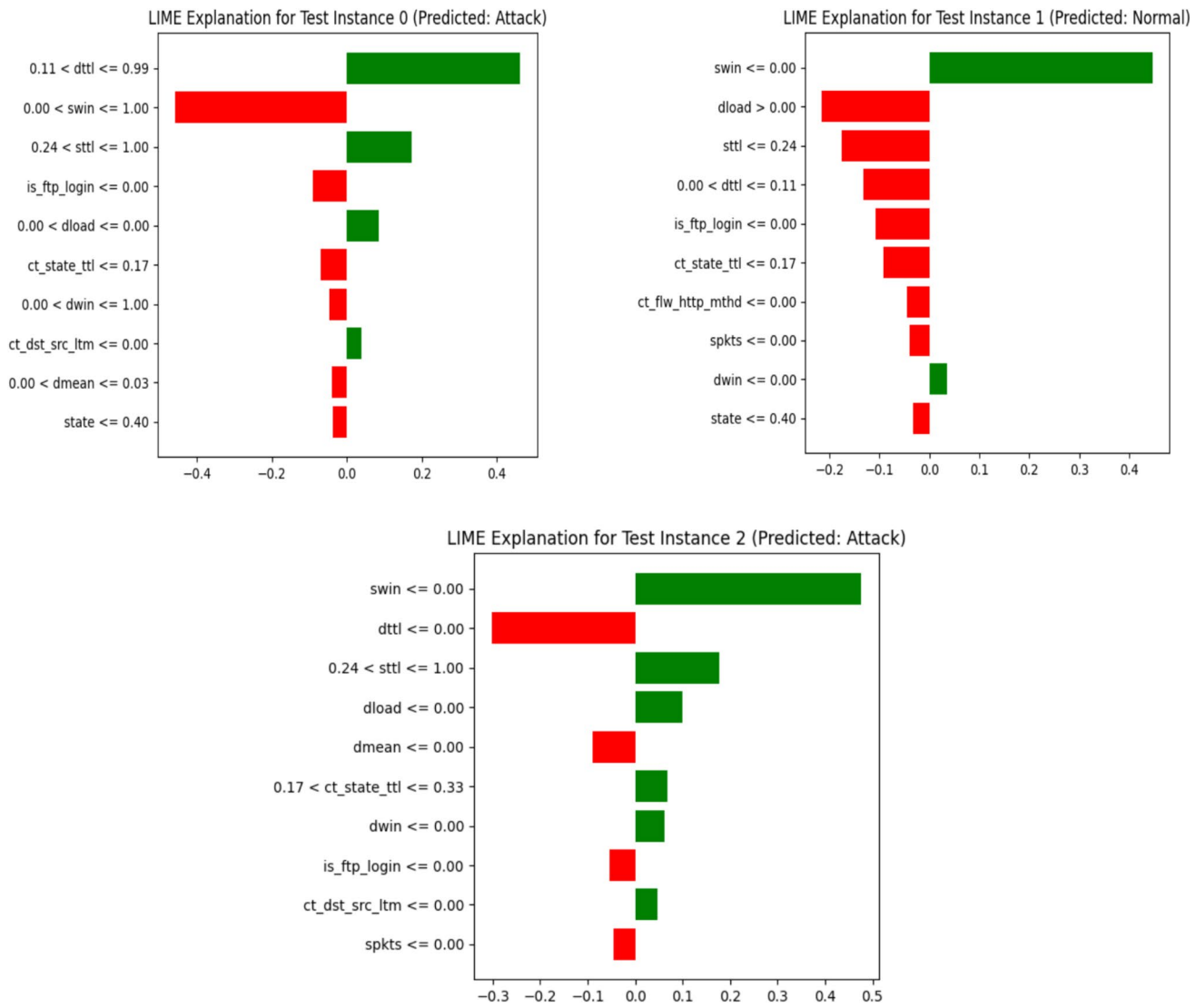


**Fig. 20** LIME outcome using the UNSWNB15 dataset on GA-GRU

**Table 9** Comparative performance of the proposed GA-optimized models against existing state-of-the-art IDS

| Author | Dataset | Accuracy% | Precision% | Recall% | F1-score% | ROC % |
|---|---|---|---|---|---|---|
| [38] | UNSW-NB15 | 87.527 | 89.00 | 88.00 | 87.00 | – |
| [3] | NSL KDD | 91.05 | 85.33 | 93.33 | 90.17 | – |
| [3] | BOT-IOT | 83.89 | 76.97 | 84.63 | 80.00 | – |
| [39] | ToN-IoT | 88.22 | 86.99 | 89.6 | 87.69 | – |
| GA-GRU | UNSW-NB15 | 92.5 | 94.8 | 93.3 | 94.0 | 98.35 |
| GA-LSTM | UNSW-NB15 | 90.88 | 95.84 | 92.49 | 94.14 | 98.2 |
| GA-GRU | IoT_ToN | 95.8 | 92 | 95.90 | 90.8 | 98.2 |
| GA-LSTM | IoT_ToN | 92.78 | 83.82 | 98.51 | 90.64 | 97.7 |

## 6 Conclusion

The exponential growth of the Internet of Things (IoT) has exposed the limitations of conventional signature- and protocol-based intrusion detection systems, which struggle to manage the increasing scale, heterogeneity, and complexity of IoT traffic. To address these challenges, this study proposed Genetic Algorithm (GA)-optimized deep learning models specifically GA-GRU and GA-LSTM architectures that integrate feature selection and hyperparameter tuning to enhance detection accuracy while minimizing computational cost. The models were trained and evaluated using

two benchmark datasets, IoT-ToN and UNSW-NB15, to assess generalization and efficiency across different network environments.

Experimental results demonstrated that GA effectively reduced the dimensionality of both datasets, lowering the number of features from 39 to 21 in GA-LSTM and from 39 to 16 in GA-GRU (IoT-ToN), and from 42 to 22 and 23, respectively, in UNSW-NB15. This reduction yielded significant improvements in space and time complexity, leading to faster training, lower memory usage, and enhanced classification performance. The GA-GRU model achieved the best results, attaining 95.8% accuracy on IoT-ToN and 92.5% on UNSW-NB15, outperforming existing state-of-the-art IDS models reported in recent literature across comparable metrics.

Furthermore, LIME-based interpretability analysis confirmed that the proposed models rely on semantically meaningful network features, enhancing trustworthiness and transparency in intrusion detection. Overall, the GA-optimized deep learning framework demonstrates robust, explainable, and computationally efficient intrusion detection, paving the way for scalable and intelligent security solutions in next-generation IoT ecosystems.

Future work will extend the GA-optimized GRU framework to real-world IoT intrusion detection scenarios using live traffic to evaluate scalability and latency under dynamic conditions. The model will be further enhanced through federated learning for privacy-preserving collaboration and energy-efficient optimizations such as pruning and quantization for edge deployment. Additionally, future studies will explore other metaheuristics (PSO, Firefly, Grey Wolf) to complement GA for improved feature selection and hyperparameter tuning. A key limitation of this study was the absence of SHAP integration, which will be addressed by adopting hybrid XAI approaches combining LIME, SHAP, and attention-based methods to strengthen interpretability and trust. These advancements will promote a scalable, explainable, and efficient IDS for IoT.

**Acknowledgements** This work was partly supported by the South African National Research Foundation under Grant nos. CPRR240327211161, CHN231229201835, RA22112976288, KIC250321305600, and AJCR230704126719.

**Author contributions** EMM: Writing Original draft, Formal Analysis, Methodology, simulation, Validation YS: Supervision, Fund Acquisition, Validation, Review and Editing, ZW: Supervision, Validation, Review and Editing.

**Funding** Open access funding provided by University of Johannesburg.

**Data availability** The datasets used in this research can be accessed through the following links:—[https://unswmy.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fz5025758%5Fad%5Funsw%5Fedu%5Fau%2FDocuments%2FUNSW%2DNB15%20dataset&ga=1\] (https:/unswmy.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fz5025758%5Fad%5Funsw%5Fedu%5Fau%2FDocuments%2FUNSW%2DNB15%20dataset&ga=1)—[https://unswmy.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fz5025758%5Fad%5Funsw%5Fedu%5Fau%2FDocuments%2FTON%5FIoT%20datasets&ga=1] (https:/unswmy.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fz5025758%5Fad%5Funsw%5Fedu%5Fau%2FDocuments%2FTON%5FIoT%20datasets&ga=1).s

## Declarations

**Conflict of interest** The authors declare no competing interests.



## References

1. Benaddi H, Jouhari M, Elharrouss O (2025) A lightweight hybrid approach for intrusion detection systems using a chi-square feature selection approach in IoT. Internet of Things (The Netherlands) 32(March):101624. https://doi.org/10.1016/j.iot.2025.101624
2. Ieropoulos V, Anthi E, Spyridopoulos T, Burnap P, Mavromatis I, Khan A, Carnelli P (2025) Collaborative intrusion detection in resource-constrained IoT environments: challenges, methods, and future directions a review. J Inf Secur Appl 93(June):104127. https://doi.org/10.1016/j.jisa.2025.104127
3. Dash N, Chakravarty S, Rath AK, Giri NC, AboRas KM, Gowtham N (2025) An optimized LSTM-based deep learning model for anomaly network intrusion detection. Sci Rep 15(1):1554. https://doi.org/10.1038/s41598-025-85248-z
4. Kasongo SM, Sun Y (2020) Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. J Big Data 7(1):105. https://doi.org/10.1186/s40537-020-00379-6
5. Elsaid SA, Shehab E, Mattar AM, Azar AT, Hameed IA (2024) Hybrid intrusion detection models based on GWO optimized deep learning. Discov Appl Sci. https://doi.org/10.1007/s42452-024-06209-1
6. Sheikhi S, Kostakos P (2022) A novel anomaly-based intrusion detection model using PSOGWO-optimized BP neural network and GA-based feature selection. Sensors 22(23):9318. https://doi.org/10.3390/s22239318
7. Khan MZ, Reshi AA, Shafi S, Aljubayri I (2025) An adaptive hybrid framework for IIoT intrusion detection using neural networks and feature optimization using genetic algorithms.

Discover Sustain 6(1):382. https://doi.org/10.1007/s43621-025-01141-9
8. Ogunseyi TB, Thiyagarajan G (2025) An explainable LSTM-based intrusion detection system optimized by Firefly algorithm for IoT networks. Sensors 25(7):2288. https://doi.org/10.3390/s25072288
9. Chinnasamy R, Subramanian M, Sengupta N, MP, R. (2025) Contextual Internet of Things intrusion detection: a sliding window convolutional neural network-gated recurrent unit model enhanced by graph neural networks. Cureus J Comput Sci. https://doi.org/10.7759/s44389-025-06001-1
10. Mallidi SKR, Ramisetty RR (2025) A multi-level intrusion detection system for industrial IoT using bowerbird courtship-inspired feature selection and hybrid data balancing. Discover Comput 28(1):109. https://doi.org/10.1007/s10791-025-09632-z
11. Mir M, Trik M (2025) A novel intrusion detection framework for industrial IoT : GCN-GRU architecture optimized with ant colony optimization. Comput Electr Eng 126(June):110541. https://doi.org/10.1016/j.compeleceng.2025.110541
12. Turgut Z, Başarslan MS (2025) XBiDeep: a novel explainable artificial intelligence based intrusion detection system for internet of medical things environment. Internet Things 33:101675. https://doi.org/10.1016/j.iot.2025.101675
13. AlGhamdi R (2023) Design of network intrusion detection system using Lion optimization-based feature selection with deep learning model. Mathematics 11(22):4607. https://doi.org/10.3390/math11224607
14. Kanna PR, Santhi P (2022) Hybrid intrusion detection using MapReduce based black widow optimized convolutional long short-term memory neural networks. Expert Syst Appl 194(May 2021):116545. https://doi.org/10.1016/j.eswa.2022.116545
15. Kilichev D, Kim W (2023) Hyperparameter optimization for 1D-CNN-based network intrusion detection using GA and PSO. Mathematics 11(17):1–31. https://doi.org/10.3390/math11173724
16. Saheed YK, Abdulganiyu OH, Tchakoucht TA (2024) Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with edge capabilities. Appl Soft Comput 155(February):111434. https://doi.org/10.1016/j.asoc.2024.111434
17. Xia Z, He S, Liu C, Liu Y, Yang X, Bu H (2024) PSO-GA hyperparameter optimized ResNet-BiGRU based intrusion detection method. IEEE Access 12(August):135535–135550. https://doi.org/10.1109/ACCESS.2024.3464529
18. Boubertakh O, Sahnoun A, Zitouni A, Harous S (2025) HyMD2I : hybrid metaheuristic - deep learning approach for intrusion detection in IoT. J Netw Syst Manage 9:74. https://doi.org/10.1007/s10922-025-09946-9
19. Vetrimani E, Arulselvi M, Ramesh G (2023) Building convolutional neural network parameters using genetic algorithm for the croup cough classification problem. Meas Sensors 27(March):100717. https://doi.org/10.1016/j.measen.2023.100717
20. Singh U, Saurabh K, Trehan N, Vyas R, Vyas OP (2024) GA-LSTM: performance optimization of LSTM driven time series forecasting: GA-LSTM: Performance Optimization..: U. Singh et al. In Computational Economics (Issue 0123456789). Springer US. https://doi.org/10.1007/s10614-024-10769-0
21. Jhong YD, Chen CS, Jhong BC, Tsai CH, Yang SY (2024) Optimization of LSTM parameters for flash flood forecasting using genetic algorithm. Water Resour Manage 38(3):1141–1164. https://doi.org/10.1007/s11269-023-03713-8
22. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
23. Chao K, Merri¨enboer van, B, Dzmitry B, Yoshua, B (2014) On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In D. and Wu, M. and Carpuat, X. and Carreras, & E. M. Vecchi (Eds.), *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103–111, October 25, 2014, Doha, Qatar. c 2014 Association for Computational Linguistics* (Vol. 1, pp. 103–111). Association for Computational Linguistics. https://doi.org/10.3115/v1/W14-4012
24. Salih AM, Raisi-Estabragh Z, Galazzo IB, Radeva P, Petersen SE, Lekadir K, Menegaz G (2025) A perspective on explainable artificial intelligence methods: SHAP and LIME. Adv Intell Syst. https://doi.org/10.1002/aisy.202400304
25. Ribeiro MT, Singh S, Guestrin C (2016) "Why Should I Trust You?" Explaining the predictions of any classifier. In: NAACL-HLT 2016-2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session, pp 97–101. https://doi.org/10.18653/v1/n16-3020
26. Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: Advances in neural information processing systems, 2017-Decem(Section 2), 4766–4775
27. Moustafa N (2021) A new distributed architecture for evaluating AI-based security systems at the edge: network TON_IoT datasets. Sustain Cities Soc 72(February):102994. https://doi.org/10.1016/j.scs.2021.102994
28. Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings, pp 1–6. https://doi.org/10.1109/MilCIS.2015.7348942
29. Zouhri H, Idri A, Ratnani A (2024) Evaluating the impact of filter-based feature selection in intrusion detection systems. Int J Inf Secur 23(2):759–785. https://doi.org/10.1007/s10207-023-00767-y
30. Ngo VD, Vuong TC, Van Luong T, Tran H (2024) Machine learning-based intrusion detection: feature selection versus feature extraction. Cluster Comput 27(3):2365–2379. https://doi.org/10.1007/s10586-023-04089-5
31. Ji R, Kumar N, Padha D (2024) CNN-GWO-voting & hybrid: ensemble learning inspired intrusion detection approaches for cyber-physical systems. Proc Indian Natl Sci Acad 91(3):848–862. https://doi.org/10.1007/s43538-024-00372-0
32. Ji R, Selwal A, Kumar N, Padha D (2025) Cascading bagging and boosting ensemble methods for intrusion detection in cyber-physical systems. Secur Priv 8(1):1–11. https://doi.org/10.1002/spy2.497
33. Ji R, Kumar N, Padha D (2025) Optimized intrusion detection approach for cyber-physical system using meta-learning with stacked generalization: an ensemble learning inspired approach. Secur Privacy 8(3):1–18. https://doi.org/10.1002/spy2.70031
34. Ji R, Kumar N, Padha D (2024) Hybrid enhanced intrusion detection frameworks for cyber-physical systems via optimal features selection. Indian J Sci Technol 17(30):3069–3079
35. Liu J, Li D, Shan W, Liu S (2024) A feature selection method based on multiple feature subsets extraction and result fusion for improving classification performance. Appl Soft Comput 150:111018. https://doi.org/10.1016/j.asoc.2023.111018
36. Zorarpaci E (2024) A fast intrusion detection system based on swift wrapper feature selection and speedy ensemble classifier. Eng Appl Artif Intell 133:108162. https://doi.org/10.1016/j.engappai.2024.108162
37. Chaudhuri A (2024) Search space division method for wrapper feature selection on high-dimensional data classification. Knowl Based Syst 291:111578. https://doi.org/10.1016/j.knosys.2024.111578

38. Pal KK, Eriksen AV, Dinh N (2025) XGBoost feature selection for multi-class and binary classification on UNSW-NB15 dataset. In: Digest of Technical Papers-IEEE International Conference on Consumer Electronics, pp 1–6. https://doi.org/10.1109/ICCE63647.2025.10930023
39. Kumari TA, Mishra S (2024) Tachyon: enhancing stacked models using Bayesian optimization for intrusion detection using different sampling approaches. Egypt Inform J 27:100520. https://doi.org/10.1016/j.eij.2024.100520