A Study on Heuristic Algorithms Combined With LR on a DNN-Based IDS Model to Detect IoT Attacks

Thanh - Thuy Tran Thi¹, Duc - Thuan Luong², Hong - Duc Nguyen², and Trong - Minh Hoang²,⁵

¹Faculty of Electronic Engineering 1, Posts and Telecommunication Institute of Technology, Hanoi, Vietnam
²Faculty of Telecommunication 1, Posts and Telecommunication Institute of Technology, Hanoi, Vietnam
hoangtrongminh@ptit.edu.vn⁵, thuytttt@ptit.edu.vn, thuanld.b19vt399@stu.ptit.edu.vn, ducnh.b19vt096@stu.ptit.edu.vn

Abstract
Current security challenges are made more difficult by the complexity and difficulty of spotting cyberattacks due to the Internet of Things explosive growth in connected devices and apps. Therefore, various sophisticated attack detection techniques have been created to address these issues in recent years. Due to their effectiveness and scalability, machine learning-based Intrusion Detection Systems (IDSs) have increased. However, several factors, such as the characteristics of the training dataset and the training model, affect how well these AI-based systems identify attacks. In particular, the heuristic algorithms (Genetic Algorithm-GA, Particle Swarm Optimization-PSO, Cuckoo Search Optimization-CSO, Firefly Algorithm-FA) optimized by the Logistic Regression (LR) approach employ it to pick critical features of a dataset and deal with data imbalance problems. This study offers an Intrusion Detection System (IDS) based on a deep neural network and heuristic algorithms combined with LR to boost the accuracy of attack detections. Our proposed model has a high attack detection rate of up to 99% when testing on the IoT-23 dataset.

Keywords: Intrusion Detection System, Deep Neuron Network, Heuristic Algorithms, IoT-23 dataset.

1 Introduction

The Internet of Things (IoT) technologies have linked billions of items and amassed a vast amount of data, which may be utilized for automated processes and intelligent computing to integrate the physical and digital worlds. These days, IoT applications impact practically every element of society, including industry, healthcare, transportation, and agriculture [11, 26]. With its benefits, new difficulties are presented to network security and performance [24].

An intrusion detection system (IDS) can identify unusual network activity brought on by successful assaults as the initial security measure. Because of the variety of devices, the ease of usage of the protocols, and the constrained resources of the system components, IoT systems might benefit from IDS [8]. IDS systems with machine learning-based active forms use past data from datasets to compare and identify real threats [25]. Accumulated datasets greatly influence the accuracy of predictions and their application in real-life scenarios and the intelligence of machine learning algorithms. Recently proposed neural network-based IDS systems use KDD-99, NSL-KDD, and UNSW-NB15 datasets to assess and identify a variety of threats on computer and IoT networks [24, 30, 6].

Deep learning is a type of machine learning that enables computers to understand the world using a hierarchy of ideas and learn from experience. Deep neural networks (DNN) are networks created using deep learning. Deep learning applications for issues like prediction, identification, and optimization have shown precise results and stimulated a sizable amount of research using this approach [27]. IDS systems utilizing DNN have gotten much interest in recent research [28]. These suggested systems are frequently paired with data preparation techniques to increase performance. However, the accuracy and scope of the application of DNN-based IDS systems have long been a problem for researchers.

This study suggests using an LR-heuristic (GA, PSO, CS, FA) technique and a DNN-based IDS system to improve the attack detection rate in an IoT context. Before DNN training, our proposed model’s a heuristic technique is optimized using Logistic Regression (LR) that chooses the most important characteristics from a dataset. On the IoT-23 [20] datasets, the suggested model has been evaluated, and in particular, our method has solved the major obstacle of the accuracy score of the DNN-based IDS system. When using data balancing techniques, the test accuracy on the IoT-23 dataset is 99%.
2 Related Work

In recent years, IDS-based machine learning systems to prevent DDOS attacks have experienced significant growth due to their effectiveness in facing various attacks from a vast number of IoT devices. The authors of [23] proposed an IDS based on Deep Learning (DL) using Feed Forward Deep Neural Networks (FFDNN) and a filter-based feature selection approach. Using the well-known NSL-KDD dataset, the FFDNN-IDS was evaluated and compared to existing machine learning techniques: Support vector machines (SVM), Decision Tree (DT), K - Nearest Neighbor (KNN), and Naive Bayes (NB). Test results indicate that FFDNN-IDS is more accurate than other techniques. Using the NSL-KDD dataset, an accelerated DNN architecture was developed [21] to detect anomalies in network data. It demonstrates that DNN-based IDS can reliably identify certain attack classes (DoS and Probe assaults) with the required training samples but cannot effectively classify attack types (R2L and U2R) with limited training samples. In [5], a DNN model for intrusion detection is proposed, along with a novel preprocessing technique tested on the KDDCup'99 dataset and designed to improve the performance of detection algorithms. The test results indicate that their pretreatment method outperforms conventional preprocessing techniques regarding precision, recall, and F1-score, resulting in more accurate detection of Advanced Persistent Threats (APTs).

The authors of [7] classify deep learning models used for intrusion detection and summarize the relevant academic literature. Using two historical datasets (KDD 99 and NSL-KDD) and two current datasets, the authors trained and evaluated four primary deep learning models for the classification task type: feedforward neural networks, autoencoders, deep-belief networks, and short-term, long-term memory networks (CIC-IDS2017, CIC-IDS2018). In all four datasets, their findings demonstrated that deep feedforward neural networks generate the appropriate evaluation metrics for accuracy, F1-scores, and training and inference times. The results also demonstrated that the supervised feedforward neural network does not perform better than two well-known semi-supervised learning models, the autoencoder and the deep belief network. By selecting potential features before network input processing, [14] aims to enhance the performance of Deep Neural Networks (DNNs). This work employs the KDD Cup 99 dataset, one of the standard datasets for intrusion detection. Based on their experiments' results, they concluded that feature selection improves IDS compared to the technique without feature selection. This research demonstrated that the DNN for IDS might improve accuracy by 99.4%, precision by 99.7%, recall by 97.9%, and F1-score by 98.8%. However, the diversity and updating of datasets in IoT today require researchers to continue looking for and developing new attack-prevention models.

One of the critical performance objectives of an ML-based IDS system is to improve attack detection performance parameters and reduce computation time complexity. Therefore, machine learning application schemes in IDS systems are often combined with other techniques. Specifically, the swarm approximation optimization techniques. The authors of [9] proposed an IDS model that used an ANN model with LR-GA to achieve a work detection rate of up to 95.26%. The ANN parameters in this model are optimized using the GSPSO technique. The authors in [12] identified the key characteristics of the data and assisted in modeling it by using GA to determine the best parameters for the decision tree (DT) and K-nearest neighbor (KNN) algorithms. They can identify DDOS attacks up to 99% of the time. However, other sorts of attacks are now far less accurately detected. In [2], the GA method is used with a support vector machine (SVM) to choose characteristics from 10 categories with three different priority levels. The authors of [32] described a model that uses a more advanced GA algorithm and deep belief networks (DBN). They assessed the outcomes using the NSL-KDD dataset, demonstrating how their methodology may simplify the network architecture and increase attack detection accuracy. Their study suggested and presented three experiments that combined genetic algorithm and logistic regression. Their suggested model has an F1-score of 93.56% and a classification accuracy of 94.55%.

Synthetic minority sampling technique (SMOTE) and particle swarm optimization (PSO) are combined in the suggested classification algorithms in [29], which also incorporate many well-known classifier techniques, including logistic regression, decision tree model C5 (5), and 1-nearest neighbor. In order to address the randomly occurring problem in a cloud computing context and improve Initial Aerofoil Geometry, the authors in [19, 18] reviewed the available modified PSO scheduling methods. The comparison demonstrates that the new algorithm successfully improves upon the original PSO. The Bayesian information criterion (BIC) was suggested in [22] with the logistic regression model and the particle swarm optimization algorithm. Investigation and comparison of the effectiveness of various fitness functions using BIC. Several different kinds of datasets are used to assess the performance of the suggested technique. Results from experiments utilizing various dataset types show how our suggested strategy can dramatically enhance classification performance with few features. The outcomes demonstrate that the suggested approaches competitively outperform other current fitness functions. [15] studies and develops associated technical levels and realizes early diagnosis and intervention of illness risk factors through applying a unique quantum particle swarm optimization. The findings of the comparative experiments confirm that the proposed strategy is suitable for creating a Logistic Regression Health Assessment Model.

The weighted binary cuckoo search (WBCS) algo-
The Logistic Regression (LR) algorithm constitutes a supervised learning approach that functions to predict the output of a dependent variable using independent variables and their relationships. LR model trains on labeled data and then estimates the coefficients of independent variables to generate a classification model. The LR algorithm owes its roots to the 19th century when statisticians formulated regression methods for data analysis. In the early versions of the LR, categorizing statistical problems was vital; however, in the 1940s and 1950s, the method was further refined by mathematicians and statisticians, including Joseph Berkson, David Cox, and William Cochran. In recent years, LR gained immense popularity in medical applications, predicting the probability of dependent variables such as the possibility of getting an infection or developing heart disease. In business and financial applications, such as predicting the likelihood of a customer buying a product, assessing the risk of investing, and forecasting the cost of developing a new product, LR became a required statistical method. LR is a widely used algorithm in machine learning, mainly in binary classification problems, such as email classification, credit fraud detection, and product categorization prediction, making it one of the foundational algorithms in machine learning. In addition, LR also handles the problem of data imbalance.

The Particle Swarm Optimization (PSO) algorithm is an optimization method inspired by the movement of a flock of birds searching for food in space. The PSO algorithm is a powerful global search method capable of quickly searching in large search spaces. This algorithm uses "particles" to find and optimize solutions. However, it also has limitations, such as the ability to fall into the local optimum and slow convergence for large-sized problems.

The Cuckoo Search (CS) algorithm is a global search algorithm based on the breeding behavior of pigeons. It was inspired by how pigeons put their eggs in other birds' nests to lay and deceive host birds. The Cuckoo Search algorithm can find the best solution in ample search space, especially in intermittent and non-derivative optimization problems. However, it also has some limitations, such as the slow convergence speed for large-sized problems and the need for a long time to find the optimal solution.

The Firefly Algorithm is an optimization method inspired by the way fireflies light up in the dark to find food and communicate with each other. This algorithm uses "fireflies" to find and optimize solutions.

The main components of the proposed model include the K-mean method, Logistic regression (LR) algorithm, heuristic algorithms, and DNN, illustrated in Figure 1. Attack types are separated into high-traffic and low-traffic assaults during the data preparation phase using the provided data set IoT-23. Data set partitioning employs processing methods that reduce computing complexity and data area size. The
data set is split 80/20 into training and test data for assaults with low traffic. For this study, we employ the standard clustering method, reducing the dataset’s dimensionality for high-traffic attacks. When clustering the data, we use the K-means technique to lessen the amount of duplicated data while still covering the complete data domain. We use the LR paired with heuristic algorithms to optimize the heuristic algorithm’s parameters and reduce the data dimension. The target data is then prepared for categorization by the DNN model by combining low-traffic and high-traffic assault data. At this point, data balancing processes are also used on the training set of the proposed model to remove distinctive biases, and once the data has been processed, the suggested DNN model will be evaluated. The in-depth processing methods are presented in detail in this part 4.

4.1 Data Preprocessing

The IoT-23 dataset is a good network traffic dataset from Internet of Things (IoT) devices published in 2020. The collection of datasets excludes local network information and instead focuses on network properties to facilitate generalization to all IoT networks. The IoT-23 database functions as a valuable resource for researchers and practitioners in the field of IoT security by providing a comprehensive dataset for testing and evaluating security solutions, including the detection and prevention of various types of IoT attacks. It has 20 malware captures executed in IoT devices and three captures for benign IoT device traffic. The dataset includes 23 features with 16 attack types described in Table 1, of which 15 are malicious layers and one is safe layer.

When conducting data processing, the missing data is processed by getting the average values of each attack type related to its features. As a result, the mean corresponding to the 16 attack types will be used to fill in the missing values in the feature column. Some features, “unnamed”, “label,” “local resp,” and “local orig” in the IoT-23 dataset, are eliminated from the read records because they are null and have no significance.

Types, connection types, services, and other network properties are analyzed and encoded for translation to numeric data. Categorical data is made up of columns like “service,” “proto,” and “conn state,” each of which transforms the various values it contains into a separate, binary state vector [32]. Therefore, the number of features can be increased by this encoding. The data file presents two columns of the IPv4 address type, “id.orig h” and “id.resp h,” each representing the value of a separate IP address. The IP address encoding standard encodes IPv4 address data into a digital format using the ‘ipaddress’ library. By converting each octet to an 8-bit binary and then concatenating four octets to create a 32-bit binary string, this library will convert IP addresses to a digital representation. Then we will convert back to decimal from 32-bit binary. The missing data is processed by getting the average values of each attack type related to its features. As a result, the mean corresponding to the 16 attack types will be used to fill in the missing values in the feature column. In this paper, we extract four types of attacks from the IoT-23 dataset displayed in Table 2. We have extracted two popular attack types with high traffic and two with low traffic, Okiru, PartOfAHorizontal-PortScan, C&C PartOfAHorizontal-PortScan, C&C-Heart Beat Attack, labeled from 0 to 3 to test with the proposed model.

4.2 Data Clustering

In this scheme, we have clustered the dataset after preprocessing the data. For high-traffic attacks, it is essential to find important data points that have much

<table>
<thead>
<tr>
<th>Number</th>
<th>Type</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Attack</td>
<td>9398</td>
</tr>
<tr>
<td>1</td>
<td>Benign</td>
<td>30854315</td>
</tr>
<tr>
<td>2</td>
<td>C&amp;C</td>
<td>21995</td>
</tr>
<tr>
<td>3</td>
<td>C&amp;C FileDownload</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>C&amp;C HeartBeat</td>
<td>33673</td>
</tr>
<tr>
<td>5</td>
<td>C&amp;C HeartBeat Attack</td>
<td>834</td>
</tr>
<tr>
<td>6</td>
<td>C&amp;C HeartBeat FileDownload</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>C&amp;C Mirai</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>C&amp;C PartOfAHorizontal-PortScan</td>
<td>888</td>
</tr>
<tr>
<td>9</td>
<td>C&amp;C Torri</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>DDoS</td>
<td>19538713</td>
</tr>
<tr>
<td>11</td>
<td>FileDownload</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>Okiru</td>
<td>60990708</td>
</tr>
<tr>
<td>13</td>
<td>Okiru Attack</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>PartOfAHorizontalPortScan</td>
<td>213852924</td>
</tr>
<tr>
<td>15</td>
<td>PartOfAHorizontalPortScan</td>
<td>5</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td>325303570</td>
</tr>
</tbody>
</table>
influence on the model instead of dealing with the entire dataset. Thus, clustering high-traffic datasets reduces the size of the dataset and selects the critical data points. In this study, we use the K-means method for high-traffic attacks. The algorithm will work as follows, initially choosing K random cluster centers. It then computes the data points for those original cluster centers if the data points closest to the cluster center will belong to that cluster. The cluster center is recalculated with the mean of the data points. The algorithm stops when it cannot be improved further. The steps of the K-means algorithm work as algorithm 1.

Algorithm 1 K-means algorithm

1: Initialize k initial center points. These center points are chosen randomly or based on basic knowledge of the data
2: Calculate each data point’s distance to the center point, then assign it to the group with the closest central point
3: Recalculate the position of k-center points based on the data points assigned to the group
4: Repeat steps 2 and 3 until the center points are unchanged or the maximum number of iterations previously given is reached
5: Returns groups of data that have been classified

4.3 Features Selection

In the IoT-23 dataset, the number of features is 33 after data preprocessing. In the fact that the IoT-23 dataset contains small traffic attacks that lead to data imbalances, making the neural network challenging to detect and confusing between types of attacks. Therefore, our study uses LR combined with various heuristic techniques to choose a small practical set of features and reduce dimensionality to enhance attack detection accuracy on the IoT-23. Moreover, it will also deal with the data imbalance problem using class weight methods.

The GA searches for potential solutions and uses crossovers and mutations to generate new generations of solutions, in which better solutions are preferred for the next generation. Solutions are represented as gene sequences, similar to the characters in a string. The LR-GA algorithm is an algorithm that combines the LR algorithm into the fitness function of GA. The LR-GA algorithm has the main steps in algorithm 2.

Algorithm 2 Optimize Genetic Algorithm parameters by LR

1: Initialize a set of initial parameters
2: Initialize the instance for the initial population
3: Calculation of fitness function based on LR algorithm
4: Select highly effective individuals
5: Apply mutation to some solutions in the new generation to create diversity in the solution set
6: Evaluate the quality of each solution in the new generation
7: Select the best solutions to become the solution set for the next generation
8: Repeat from step 3 to step 6 until the best solution or a sufficient number of generations is reached
9: End

The LR-PSO algorithm is an algorithm that combines the LR algorithm into the fitness function of PSO. The LR-PSO algorithm has the main steps of algorithm 3.

Algorithm 3 Optimize Particle Swarm Optimization parameters by LR

1: Initialize a set of initial parameters
2: Initialize the instance for the initial population
3: Calculate the fitness function based on the LR algorithm and choose the best position of the individuals as the next population
4: Update the position of each particle by computing a speed vector and adding it to the particle’s current position. The speed vector is calculated by combining two components: (a) the free component, where the particle is guided by the best position it has found, and (b) the social component, where the particle is guided by the best position the other particles in the set have found
5: Evaluate the quality of each grain after updating the position
6: Update the best position that the set of particles has found
7: Repeat step 3 to step 6 until the best solution is reached or a sufficient number of iterations are reached
8: End

The Cuckoo Search algorithm combined with the Logistic Regression algorithm is presented in algorithm 4.

The Firefly algorithm combined with the Logistic Regression algorithm is implemented as algorithm 5.

4.4 Deep Neural Network-based IDS Model

A DNN model with five layers—one input layer, three hidden layers, and one output layer—is constructed in this study. Initially, the layer will have 128 nodes, which will be compressed to 64, 32, and 16 nodes. The output will then depend on the number of classes in the processing dataset. This neural network’s nonlinear function is called “Relu”. To improve the learning model, we employ the Dropout approach to randomly turn off some nodes during training at a rate of 5% after each lesson. In the network, the output layer will have four nodes corresponding to the problem’s output number. We use the term "softmax" for the activation
function of this last layer. The “Adam” optimizer and
the cross entropy function simulate and put the opti-
mization function into practice. The learning rate is
vital for the learning model; we initially set it to 0.001
to allow it to learn; if the model deteriorates or valida-
tion errors rise for three consecutive periods, we drop
the learning rate by 0.2 times to a minimum of 0.00001.
We employed a method that prompted the suggested
model to stop training early to reduce over-learning.
The training is over when the validation error function
does not decrease after a certain number of epochs.
To verify that the validation error does not increase
over the epochs, we can adjust the number of epochs
to provide the best network accuracy over the learning
interval. Lastly, using the test IoT-23 dataset, we will
assess the proposed model. Loss, precision, recall, and
F1-score are the criteria that are examined.

Algorithm 4 Optimize Cuckoo Search parameters by
LR
1: Initialize a cuckoo-searchable pair of pants that in-
cludes random values for the parameters of the Lo-
gistic Regression model
2: Evaluate each bird’s clothing performance using
the Logistic Regression model to predict and com-
pare with an actual value
3: Use the breeding strategy of the cuckoo search to
create the next generation of populations by com-
bining the values of the birds with the best effect
4: Repeat steps two and step 3 until the best stopping
criterion is reached
5: End

Algorithm 5 Optimize Firefly Algorithm parameters
by LR
1: Initialize Fireflies. Initialize a large number of ini-
itial fireflies. Set parameters, including the num-
ber of fireflies, alpha parameter (similar arithmetic
with coefficients in logistic regression algorithm),
distance r
2: Search and move. Calculate the objective func-
tion value (usually the cost function) for each firefly
based on the parameters of the logistic regression
model
3: Evaluation and selection. Performance evaluation
of logistic regression model with searched param-
eters. Please select the best fireflies based on their
objective function value
4: Update the model. Use the parameters of the best
fireflies to update the logistic regression model
5: Repeat steps 2 to 4 until the performance require-
ments of the model are met
6: End

5 Numerical Results and Discussion

Table 3 shows the parameters that we use for this pro-
posed model. In this study, we use the features se-
lection technique to reduce the dimensionality of the
dataset, which leads to choosing a small practical set
of features, to increase the speed without significantly
reducing the accuracy. Following are the models that
we have tested on our little IoT-23 set. The training
models have been opened with ‘class_weight’ weights,
which helps the models to work effectively with imbal-
canced datasets. We use the numerical parameters to
evaluate the proposed method: precision, recall, and
F1-score.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of remaining features</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR-GA</td>
<td>8</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>LR-PSO</td>
<td>30</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>LR-CSO</td>
<td>26</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>LR-FAO</td>
<td>26</td>
<td>0.99</td>
<td>0.99</td>
<td>0.98</td>
</tr>
</tbody>
</table>

The performance of the proposed model is shown in
Table 4. The results of the four heuristic algorithms
used are generally relatively good. However, of the
four algorithms for feature selection above, the LR-
GA algorithm gives the best results. Because after
features selection, the LR-GA algorithm selects only
eight features from the original 33 features, still gives
high-performance parameters up to 99%, and reduces
computation time. The LR-GA algorithm will reduce
the computation time on the test set because it only
has to decide based on eight features, but the training
process needs more time because it has to choose eight
features from the original 33 features. In other algo-
rithms, the remaining features are 30, 26, and 26 for
LR-PSO, LR-CSO, and LR-FAO algorithms.

The training process of the four models shown in
Figures 2, 3, 4, and 5 shows that the training results
of the algorithms are relatively good.

Finally, Table 5 displays the experimental perfor-
mance outcomes (precision, recall, F1 score) of each
class of each algorithm for the proposed model that
was verified on the IoT-23 dataset. We can see that the
proposed model has outstanding metrics when used on
small samples. As a result, it is preferred to be applied
to devices with low resources at edge networks.

Table 4: The performance of the proposed model on
the IoT-23 dataset.

This paper proposes a DNN-based intrusion detec-
tion system (IDS) model with heuristic methods (GA,
PSO, CSO, and FAO) combined with LR to detect
and evaluate attacks. The proposed model uses the
K-means clustering method to reduce the data size.
Heuristic algorithm’s parameters are optimized by LR
to extract important features while using class-weight
techniques to prevent imbalance attacks. Attacks are
differentiated based on DNN. On the IoT-23 dataset, the proposed model has been evaluated using a variety of sample sizes. The test results demonstrate that, for all heuristic methods, our proposed model’s attack de-

<table>
<thead>
<tr>
<th>Method</th>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR-GA</td>
<td>0</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.99</td>
<td>0.98</td>
<td>0.99</td>
<td>177</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>168</td>
</tr>
<tr>
<td>LR-PSO</td>
<td>0</td>
<td>0.99</td>
<td>1</td>
<td>0.99</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.99</td>
<td>1</td>
<td>1</td>
<td>177</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>168</td>
</tr>
<tr>
<td>LR-CSO</td>
<td>0</td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>0.99</td>
<td>1</td>
<td>177</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.99</td>
<td>1</td>
<td>1</td>
<td>168</td>
</tr>
<tr>
<td>LR-FAO</td>
<td>0</td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.99</td>
<td>0.98</td>
<td>0.99</td>
<td>100000</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>177</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
<td>168</td>
</tr>
</tbody>
</table>
Detection accuracy up to 99.9% with a modest number of samples.

Moreover, the GA algorithm combined with LR on a DNN-based IDS model for attack detection is the most effective of the heuristic algorithms because the number of features selected is relatively small (8 features) but still ensures accuracy, thereby reducing the computation time. Consequently, the proposed model applies to network border devices with limited resource availability, particularly the model employing the GA algorithm. The following steps will entail the deployment of real-time devices in agricultural IoT systems.

Acknowledgement: VINIF partly supports this work. Tran Thi Thanh Thuy was funded by the Master, Ph.D. Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2022.ThS.087.

References


[14] Li, L.-H., Ahmad, R., Tsai, W.-C., and Sharma, A. K. A feature selection based dnn...


[27] SARKER, I. H. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science 2, 6 (2021), 420.


