A Hybrid Levy Arithmetic and Machine Learning-Based Intrusion Detection System for Software-Defined Internet of Things Environments

Wenpan SHI[1], Ning ZHANG[2]*
Department of Computer and Digital Law, Hebei Professional College of Political Science and Law, Shijiazhuang 050000, China[1]
Baoding Open University, Baoding 071000, China[2]

***Abstract*—The convergence of Software-Defined Networking (SDN) and the Internet of Things (IoT) has enabled a more adaptable framework for managing SDN-enabled IoT (SD-IoT) applications, but it also introduces significant cyber security risks. This study proposes a lightweight and explainable intrusion detection system (IDS) based on a hybrid Levy Arithmetic Algorithm (LAA) for SD-IoT environments. By integrating Levy randomization with the Arithmetic Optimization Algorithm (AOA), the LAA enhances feature selection efficiency while minimizing computational overhead. The model was evaluated using the NSL-KDD and UNSW-NB15 datasets. Experimental results demonstrate that the LAA outperformed baseline models, achieving up to 89.2% F1-score and 95.4% precision, while maintaining 100% detection of normal behaviors. These outcomes highlight the proposed system's potential for accurate and efficient detection of cyber-attacks in resource-constrained SD-IoT environments.**

***Keywords*—Intrusion detection; internet of things; software-defined; feature selection; levy arithmetic**

I. Introduction

A. Background

The Internet of Things (IoT) emerged from the rapid development of intelligent sensors recently and the need for device connectivity [1]. IoT presents broad opportunities in the healthcare, industrial, and supply chain industries, requiring robust stability, resilience, scalability, versatility, and control [2]. Moreover, IoT components have limits to their capabilities and contain embedded chips with various configurations. Traditional networks have become increasingly complex due to IoT-specific demands. Software-Defined IoT (SD-IoT) seeks to bring Software-Defined Networking (SDN) to the IoT by providing resource flexibility and network management for existing networks. SDN is considered a key technology to develop next-generation networks [3].

SDN transforms conventional Internet architecture via the separation of management and data layers [4]. Therefore, the management layer features more intelligence, programming, and innovation and accesses all SD-IoT components where resources and traffic can be efficiently managed. The security challenges associated with SD-IoT prevent its applications from being realized sooner. First of all, security concerns stem from the shared decision-making power of SD-IoT [5]. Attackers can quickly initiate and take over central controllers by conducting and applying malicious techniques and tactics like Denial of Service (DoS), Distributed DoS (DDoS), and malware, implement erroneous policies, and degrade network performance. Consequently, a security strategy must be a core part of the SD-IoT design to protect against cyber-attacks and maintain functionality [6].

B. Problem Statement

A Network Intrusion Detection System (NIDS) is a tool developed to track and examine traffic on a network to identify threats, breaches, or illegal activities [7]. Signature-driven (also called misuse-focused or knowledge-based) and anomaly-based (also called behavior-based) methodologies are two primary methods for detecting intrusions into IoT systems [8]. It is possible to create a hybrid detection mechanism by combining both. However, this would require a lot of energy and resources to implement. Unlike anomaly-based systems that detect attacks based on traffic patterns, signature-driven systems classify traffic based on known threats. Existing and well-known attacks are well-protected by signature-based systems [9].

The rise of SD-IoT networks demands adaptive security mechanisms to address diverse cyber threats. While, commonly used, traditional methods like user authentication and encryption lack the flexibility to detect a wide range of evolving attacks in dynamic environments [10]. Intrusion Detection Systems (IDS), particularly those powered by machine learning, have proven effective in analyzing network traffic to identify attack patterns more accurately [11]. However, in resource-constrained SD-IoT systems, IDS models must be lightweight, analyzing only critical traffic features to maintain efficiency. Identifying these key attributes is a challenge. Moreover, ML-based IDS predictions are often tricky for cyber security experts to interpret, creating a need for Explainable Artificial Intelligence (XAI). XAI enhances transparency, allowing experts to understand and trust the decisions made by these models in cyber defense [12].

Applications such as geothermal energy extraction and underground mining increasingly rely on interconnected sensors and automated control systems for real-time monitoring and safety management. These cyber-physical environments, which include complex geological modeling and simulation efforts [13], are inherently vulnerable to cyber threats, thereby reinforcing the need for robust and lightweight intrusion detection systems in SD-IoT settings.

C. Research Objectives

This study aims to address critical challenges in SD-IoT security by proposing an innovative solution for building a lightweight machine learning-based IDS. The main objective is to develop an efficient IDS by selecting an optimal subset of features, minimizing computational complexity, and conserving computing resources. The research introduces a novel feature selection method using the Levy-Arithmetic Algorithm (LAA), which applies the Levy flight random step theory to the Arithmetic Optimization Algorithm (AOA) to enhance search efficiency. Key contributions to this research include:

- Defining a security model for SD-IoT applications;
- Introducing a lightweight IDS using minimal features optimized by LAA;
- Training machine learning models such as Multi-Layer Perceptron, XGBoost, Random Forest, and Decision Tree on the selected features.

The remainder of the paper is structured as follows: Section II comprehensively reviews scholarly sources. Section III discusses the proposed method. Section IV presents the experimental setup and results. Section V outlines the primary outcomes, highlights the contributions, and suggests future research.

II. Related Work

This section reviews the existing research on IDS for IoT deployments, listed in Table I. It highlights machine learning-based algorithms, feature selection techniques, and bio-inspired optimization algorithms for coping with resource constraints.

Rahman, et al. [14], proposed two approaches to overcome the limitations of centralized IDS for resource-limited endpoints: semi-distributed and decentralized. They used concurrent machine-learning algorithms to distribute the computing workload, with the semi-distributed scenario involving simultaneous modeling on the edge for feature selection and multiple classification layers. The decentralized scenario involved independent processes for feature selection and multi-layer perceptron classification, then amalgamated by a coordinated edge or fog for decision-making. The proposed approaches shows potential for detection accuracy equivalent to centralized IDS.

Forestiero [15], devised a technique for identifying irregularities in IoT using activity footprints. IoT2Vec, an embedding methodology, is used to depict devices and services using dense vectors. These vectors are allocated to mobile agents that adhere to an adapted bio-inspired paradigm. This approach facilitates intelligent global behavior derived from local movement rules recognized by all agents. A similarity rule facilitates each agent's selective application of movement rules, promoting automatic proximity among similar agents. The approach may detect solitary agents exhibiting anomalous behaviors, perhaps revealing intruders or malevolent users.

Li, et al. [16], used an Artificial Neural Network (ANN) to identify anomalous activity in a healthcare IoT system. The precision of recognition is significantly influenced by the characteristics inputted into the artificial neural network. Identifying relevant and distinctive aspects of network traffic is a critical and complex challenge due to its substantial influence on learning. The suggested approach utilizes the butterfly optimization algorithm to determine the ideal features for learning in an artificial neural network. The findings, achieving an accuracy of 92%, confirm the algorithm's efficacy in detecting discriminative aspects of traffic patterns. The suggested technique surpassed the performance of decision trees, support vector machines, and ant colony optimization used in prior research for the same objective.

TABLE I. Comparative Analysis of IDS Approaches for IoT Deployments

Reference	Approach	Feature selection	Algorithm/model	Key findings
[14]	Semi-distributed and decentralized IDS approaches	Concurrent machine learning algorithms for workload distribution	Multi-layer Perceptron classification	Distributed IDS models show detection accuracy equivalent to centralized IDS.
[15]	Activity footprint analysis using IoT2Vec embedding	Bio-inspired selective movement rules	Mobile agents following adapted bio-inspired paradigms	The method detects anomalous behaviors by identifying isolated agents.
[16]	Anomaly detection in healthcare IoT	Butterfly optimization algorithm	Artificial neural network	Achieved 93.2% accuracy, outperforming decision trees, SVM, and ant colony optimization in feature selection.
[17]	Hybrid metaheuristic-deep learning for IoT intrusion detection	Harris hawk optimization and fractional derivative mutation	LSTM and GRU models	Outperformed other approaches in accuracy and efficiency on public datasets.
[18]	Detection of botnets using hybrid metaheuristics and machine learning	Modified firefly optimization	Hybrid CNN and quasi-recurrent neural network	Superior performance for botnet detection in cloud-based IoT systems.
[19]	IDS using variable searching pattern optimization for feature selection	Variable searching pattern optimization	Deep recurrent neural network	Achieved 96.1% accuracy, effectively identifying intrusions.
[20]	Hybrid IDS using grey wolf optimization and support vector machine	GWO for kernel function and parameter optimization	SVM with GWO	Outperformed other models in F-score, recall, precision, and accuracy on TON_IoT and NSL-KDD datasets.

Sanju [17], presented a hybrid metaheuristic-deep learning methodology to improve the detection of intrusions in IoT systems. An enhanced metaheuristic approach using an ensemble of Recurrent Neural Networks (RNNs) is used to improve intrusion detection in IoT. Various attack kinds in IoT systems are discerned by using LSTM and GRU models, which are forms of RNNs. Feature selection is conducted using Harris Hawk optimization and fractional derivative mutation. The evaluation of the suggested methodology used publicly accessible datasets, and the empirical study indicated that it outperforms other comparable approaches for accuracy and efficiency. It offers a viable technique for improving intrusion detection in IoT systems and may serve as a basis for future research in this domain.

Almuqren, et al. [18] introduced a Hybrid Metaheuristics with a Machine Learning-based Botnet Detection (HMMLB-BND) approach inside a cloud-assisted IoT system. HMMLB-BND concentrates on identifying and categorizing Botnet assaults inside the cloud-based IoT ecosystem. The Modified Firefly Optimization (MFO) algorithm is used for feature selection. HMMLB-BND employs a hybrid convolutional neural network and quasi-recurrent neural network module to identify botnets. The chaotic butterfly optimization approach is used for optimum hyperparameter tuning. A series of simulations were conducted on the N-BaIoT dataset, and the experimental results indicated the superiority of HMMLB-BND compared to other current methodologies.

Jayasankar, et al. [19] suggested an IDS using variable searching pattern optimization for feature selection with an optimum deep recurrent neural network model in an IoT context. It consists of a two-phase procedure: feature selection and incursion classification. In the first step, an ideal set of features is identified with variable searching pattern optimization. Subsequently, in the second phase, breaches are recognized and classified using the DRNN model. The hyperparameters of the DRNN are optimally selected using the Nadam optimizer. A comprehensive simulation study of the model is verified using a benchmark IDS dataset, and the results demonstrate the effectiveness of intrusion detection. The suggested model effectively identifies intrusions with an accuracy of 96.1%.

Ghasemi and Babaie [20] developed a hybrid intrusion detection technique using Grey Wolf Optimization (GWO) and Support Vector Machine (SVM). The SVM distinguishes between anomalous and normal records, while the GWO identifies the kernel function, selects features, and optimizes parameters for the SVM to enhance classification accuracy. The simulations demonstrate that the proposed method surpasses others in detection accuracy, precision, recall, and F-score on the NSL-KDD and TON_IoT datasets.

III. Proposed Method

This section summarizes the graphical abstract used for the proposed model (LAA), which differs from conventional methods of selecting features. Detailed architectural pipelines for the model are described below. A methodology is described in the next section to elucidate the output (prediction) of the model. Fig. 1 illustrates a lightweight, explainable IDS.

This section presents a detailed method for determining appropriate features for devices with storage limitations. As a result, model performance improves owing to decreased calculation time and resource use. Machine learning relies on selecting a subset of optimum features from the available feature dimensions [21]. In this way, the dimensions of the feature vector become smaller, computation time decreases, and machine learning performance improves. A variety of feature selection techniques are used to reduce dimension. Some techniques for selecting features include LAA, information gain, and correlation coefficients.

AOA applies a metaheuristic strategy for analyzing exploration and exploitation balances based on math operations, like Addition, Subtraction, Multiplication, and Division. AOA is primarily inspired by the application of Arithmetic operators to resolve Arithmetic problems. According to Fig. 2, Arithmetic operators are arranged in a hierarchy according to their ascending dominance. The optimization procedure starts by randomly generating candidate solutions (*X*) given by Eq. (1). Best candidate solutions are considered the best-obtained or nearly optimum solutions in each iteration.

$$X = \begin{pmatrix} x_{1,1} & \cdots & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ x_{n-1,1} & \cdots & \cdots & x_{n-1,j} & \cdots & x_{n-1,n} \\ x_{n,1} & \cdots & \cdots & x_{n,j} & x_{n,n-1} & x_{n,n} \end{pmatrix} \tag{1}$$

Prior to the AOA commencement, it must choose the search strategy (i.e., exploration or exploitation). The Math Optimizer Accelerated (MOA) function is a coefficient derived from Eq. (2) used in subsequent search stages.

$$MOA(C_{iter}) = Min + C_{iter} \times \left(\frac{Max - Min}{M_{iter}}\right) \tag{2}$$

where, $MOA(C_{iter})$ represents the value of the function at iteration *t*, M_{iter} indicates the maximum number of iterations, C_{iter} indicates the current iteration and *Min* and *Max* are the lowest and highest accelerated values.

In the AOA, the Division and Multiplication operators generate highly dispersed values during mathematical computations, enhancing the search process's exploration phase. However, due to their high dispersion, these operators struggle to converge on the target as efficiently as the Addition and Subtraction operators. The exploration phase focuses on identifying near-optimal solutions, often requiring several iterations. At this stage, Division and Multiplication operators play a key role in enhancing communication between the exploration and exploitation phases, ultimately supporting the search process.

AOA's exploration operators randomly scan the search space across different regions, aiming to identify better solutions using two primary strategies: the Division and Multiplication strategies, as represented in Eq. (3). This search phase is governed by the MOA function, with the condition r1 > MOA. As illustrated in Fig. 3, the Division operator is activated when r2 < 0.5, while the Multiplication operator remains inactive until the Division task is completed. If the condition is not met, the Multiplication operator takes over. A stochastic scaling coefficient is also applied to introduce more diversity into the exploration, ensuring that a broader range of regions within the search space is evaluated.

$$x_{i,j}(C_{iter} + 1) = \begin{cases} best(x_j) \div (MOP + \varepsilon) \times \\ \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & r2 < 0.5 \\ best(x_j) \div MOP \times \\ \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & oherwise \end{cases} \tag{3}$$

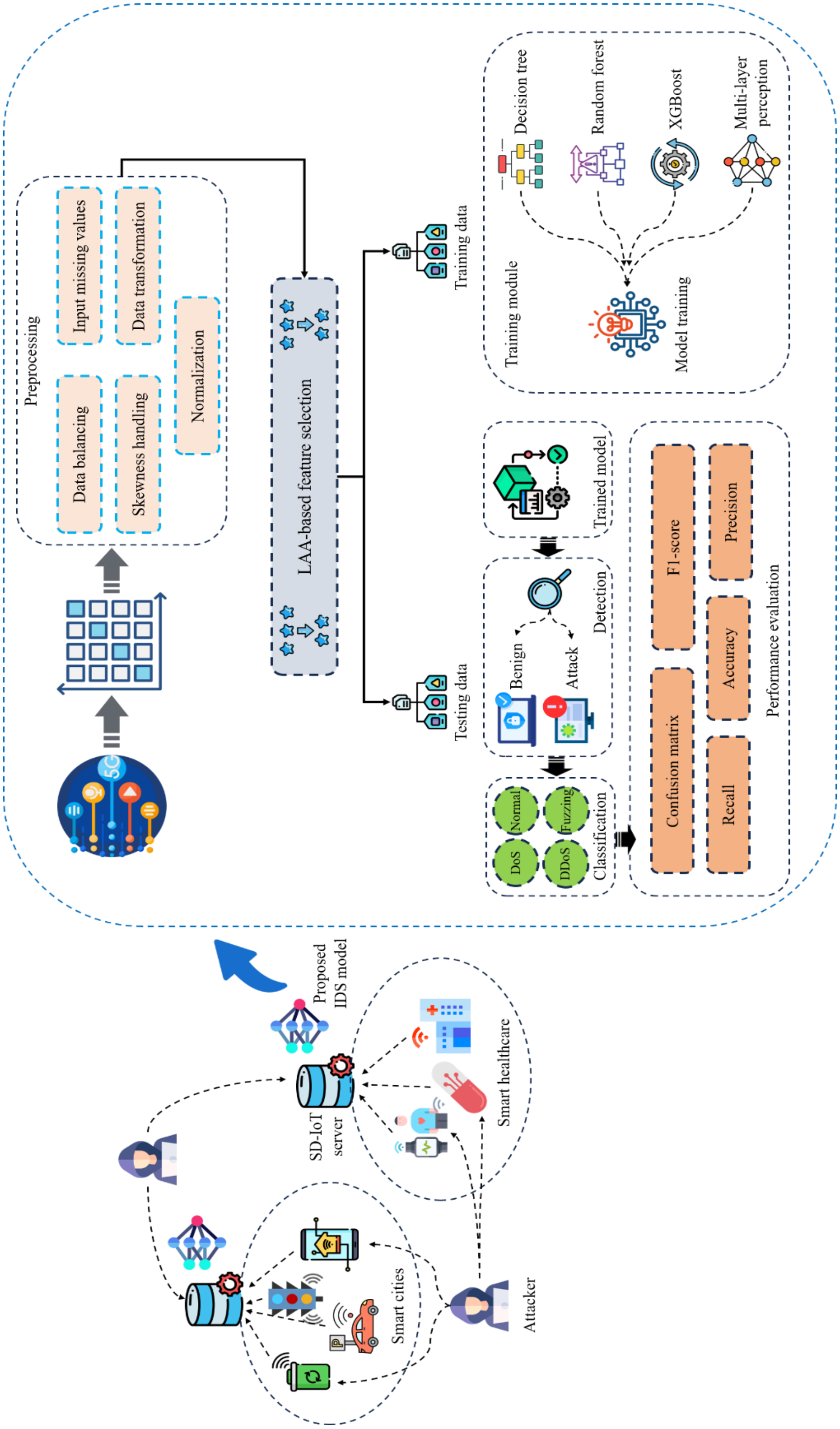

Fig. 1. System model.

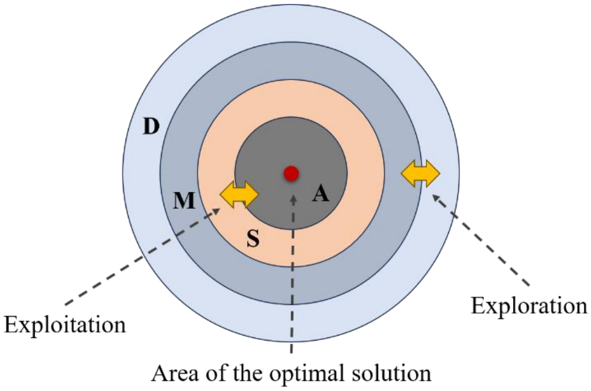

Fig. 2. Arithmetic operators.

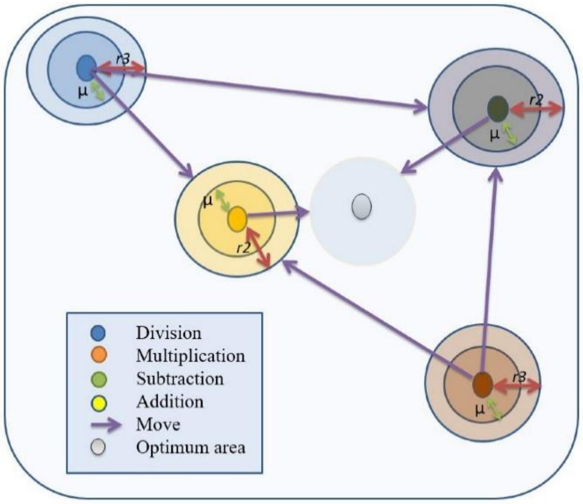

Fig. 3. Search space of AOA.

where, *r2* represents a random number ranging from 0 to 1, μ regulates search operations as a control parameter, ε is a constant parameter for avoiding zero, and MOP gives a probability function-based math optimizer. Multiplication and subtraction operators have low precedence, allowing local searches to find the optimum outcome. The candidate solution is updated iteratively based on Eq. (4) to reach the optimal result.

$$x_{i,j}(C_{iter}+1) = \begin{cases} best(x_j) - MOP \times \\ \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & r3 < 0.5 \\ best(x_j) + MOP \times \\ \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & oherwise \end{cases} \quad (4)$$

In Eq. (3) and Eq. (4), Math Optimizer Probability (MOP) is the key function for finding the optimal value and determining its capability, determined by Eq. (5).

$$MOP(C_{iter}) = 1 - \frac{(C_{iter})^{1/a}}{(M_{iter})^{1/a}} \quad (5)$$

The Levy Random Step (LRS) denotes random walking in which the size of the steps follows a statistical distribution termed the Levy pattern. Its thick tails characterize this distribution, indicating that extreme or unusual events happen more often than normal occurrences. In a Lévy random walk, the step dimensions and orientations are sampled from this distribution, yielding many little steps interspersed with occasional significant leaps in random directions. This unpredictability must be regulated to successfully direct efforts toward optimum solutions while reducing departures from the best potential outcomes. Random steps are stochastic processes that involve taking a series of unpredictable steps, as expressed mathematically in Eq. (6).

$$S_n = \sum_{i=1}^{n} x_i = x_1 + x_2 + \cdots + x_n \quad (6)$$

where, S_n is the sum of consecutive steps and x_i represents each random step. The notion of Lévy random steps and flights, given by the French mathematician Paul Lévy, stems from the first investigations of stochastic processes in physics, including particle motion in fluids and gases. Levy walks and steps have been extensively studied and used across several disciplines, demonstrating their efficacy in optimization issues and other applications. The essence of a Lévy random step is rooted in the Lévy distribution, distinguished by its thick tails, which markedly enhance the probability of extreme step sizes relative to a normal distribution. The probability of a Lévy step may be mathematically estimated as shown in Eq. (7).

$$L(x) \approx |x|^{1-a} \quad (7)$$

where, α controls the tail heaviness, typically in the range $0<\alpha\leq 2$. The Levy distribution probability density function is expressed as in Eq. (8).

$$L(x) = \frac{1}{\pi}\int_0^{\infty} e^{-\gamma\tau^{a}} cos(\tau x) d\tau \quad (8)$$

where, γ is a scaling parameter, usually set to 1, and τ is a small-time interval. As α increases, the distribution shifts closer to the mean, while lower values of α correspond to a distribution further from the mean. Mantegna's approach produces random numbers according to the Lévy distribution. This algorithm effectively generates Levy steps by sampling two random variables from normal distributions. The equation for producing Lévy steps is given in Eq. (9).

$$S = \frac{\nu}{|\nu|^{1/a}} \quad (9)$$

where, ν is a normally distributed variable, standard deviations $\sigma\nu$ determined by the Levy distribution's characteristics.

The LRS is advantageous in optimization algorithms as it facilitates rapid search space exploration via integrating both little, frequent steps and substantial, infrequent leaps. This dual nature enables the algorithm to evade local optima and progress toward more advantageous areas of the search space, hence increasing the probability of identifying a global optimum.

The fundamental components of metaheuristic algorithms, specifically the search space, assessment mechanism, position modification technique, new solution acceptance criteria, and stopping conditions, are crucial in determining their effectiveness. AOA is known for its simplicity and broad applicability to various optimization problems. However, it may struggle with more complex problems, often getting trapped in local optima or requiring numerous iterations to reach the optimal solution. The AOA employs a math-optimized likelihood, as shown in Eq. (5), which adjusts iteratively and determines possible solutions within the search area, as described in Eq. (3) and Eq. (4).

To overcome these limitations, the proposed Levy Arithmetic Algorithm (LAA) introduces LRS into the AOA framework. By incorporating these steps, potential solutions can occasionally make broad, random shifts, allowing the algorithm to discover unexplored areas of the search domain and increasing the likelihood of discovering better optimal solutions.

In the LAA, candidate solutions are updated using the arithmetic operators from AOA and enhanced by the LRS, generating stochastic jumps based on the Levy pattern. This enables the solutions to shift randomly to new positions, promoting a more extensive search for optimal solutions in each cycle. The direction and scale of these jumps are influenced by decision variables (Dim) and population size (N), determined by the dimensionality of the problem and the characteristics of the LRS. While incorporating LRS allows for a broader search space exploration, it may be slower than the standard AOA to reach the optimum solution.

In LAA, the exploration phase, governed by Eq. (3), and the exploitation phase, defined by Eq. (4), are altered by LRS (S), as described in Eq. (9), and are expressed mathematically in Eq. (10) and Eq. (11).

$$x_{i,j}(C_{iter}+1)=\begin{cases} best(x_j)\div S\times(MOP+\varepsilon)\times \\ \left(\left(UB_j-LB_j\right)\times\mu+LB_j\right), & r2<0.5 \\ best(x_j)\times MOP\times S\times \\ \left(\left(UB_j-LB_j\right)\times\mu+LB_j\right), & oherwise \end{cases} \tag{10}$$

$$x_{i,j}(C_{iter}+1)=\begin{cases} best(x_j)-S\times MOP\times \\ \left(\left(UB_j-LB_j\right)\times\mu+LB_j\right), & r3<0.5 \\ best(x_j)\times MOP\times S\times \\ \left(\left(UB_j-LB_j\right)\times\mu+LB_j\right), & oherwise \end{cases} \tag{11}$$

using Eq. (10) and Eq. (11) enhances candidate solution diversity and prevents the algorithm from becoming stuck in local optima. By incorporating the LRS into the LA, the algorithm explores larger areas more effectively, thereby continuously discovering better solutions that traditional arithmetic optimization methods might miss. At first, fitness functions and best solutions are determined from objective functions, decision variables, and conditions, and they are dynamically modified based on the algorithm parameters and the potential solutions.

Three key variables (*r1*, *r2*, *r3*) are critical in narrowing the searching space based on four different operations: addition, subtraction, multiplication, and division, as defined by the Levy flight formulation, enabling the algorithm to approach the optimal solution. Integrating LRS enhances the algorithm's ability to search globally, making it more robust and efficient, thereby increasing the chances of identifying the global optimum. This method proves particularly useful in resolving optimization issues in which a number of local optima must be investigated before reaching the global best solution.

IV. Results and Discussion

A. Datasets

Two widely used datasets are used to analyze the proposed LAA for intrusion detection in SD-IoT environments: UNSW-NB15 and NSL-KDD. UNSW-NB15 is commonly used in intrusion detection research and contains 43 features, including primary network attributes and security features. It includes a class feature with several categories: Exploits, DoS, Fuzzers, Normal, and Backdoors. The dataset comprises 257,673 instances, 70% used for training and 30% for testing. NSL- KDD, as an improved version of the KDD Cup 1999 dataset, contains 42 attributes and focuses on four major types of attacks: Remote to Local (R2L), User to Root (U2R), DoS, and Probe. The dataset comprises 148,517 records, with 85% assigned to training and 15% for testing. Both datasets are significant for testing the efficiency of IDS models, as they cover a broad range of network traffic and cyber-attack scenarios, providing comprehensive benchmarks for assessing performance.

B. Evaluation Metrics

The proposed LAA was assessed using a number of standard metrics, such as accuracy, precision, recall (true positive rate), and F1-score, commonly used in IDS research.

- Accuracy: This parameter determines the total correctness of the algorithm by evaluating the proportion of correctly classified instances relative to the total number of instances, calculated by Eq. (12).

$$A=\frac{\mathrm{TN}+\mathrm{TP}}{\mathrm{FN}+\mathrm{FP}+\mathrm{TN}+\mathrm{TP}} \tag{12}$$

where, FN is a false negative, FP is a false positive, TN is a true negative, and TP is a true positive.

- Precision: It determines the quality of positive predictions by multiplying the number of true positives by the number of positive predictions, calculated using Eq. (13).

$$P=\frac{\mathrm{TP}}{\mathrm{FP}+\mathrm{TP}} \tag{13}$$

- Recall: This metric represents the model's capacity to detect all positive instances correctly and is calculated using Eq. (14).

$$R=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{14}$$

- F1-score: Precision and recall can be balanced by a harmonic mean when data is imbalanced, as defined in Eq. (15).

$$F=2\times\frac{Recall\times Precision}{Recall+Precision} \tag{15}$$

C. *Environment*

The experimental setup for evaluating LAA was performed on a system equipped with an Intel Core i5-8250U CPU running at 3.4 GHz with a Quad-Core configuration supported by 16 GB of DDR4 RAM. The operating system used was Windows 10 (64-bit), ensuring compatibility with the tools employed in the experiment. MATLAB was utilized to implement the LAA, while Python was used for data pre-processing and additional statistical analysis. This computational environment provided adequate resources to efficiently run the experiments, ensuring that the LAA's performance could be fairly compared with the baseline models in terms of speed and accuracy.

D. *Baseline Models*

The effectiveness of LAA was evaluated by comparing it with well-established machine learning models of IDS. These included the SVM, which identifies the optimal hyper plane for classification tasks; Decision Tree (DT), which splits data based on feature values for decision-making; and Random Forest (RF), a method of constructing multiple decision trees to maximize accuracy and minimize over fitting. Additionally, models like ANN, which captures complex patterns in data, and Logistic Regression (LR), a simpler model estimating binary outcomes, were used. Other baseline models included K-Nearest Neighbors (KNN), a distance-based classifier; Naive Bayes (NB), a probabilistic model based on Bayes' theorem; and AdaBoost, a boosting technique that combines weak classifiers to form a more robust classifier. These models served as benchmarks to demonstrate how LAA compares accuracy, computational efficiency, and detection capabilities.

E. *Performance Evaluation*

Fig. 4 shows the precision of the LAA and other algorithms for detecting intrusions in the UNSW-NB15 dataset. The LAA achieved the highest precision for several attack types, including Fuzzers (83.2%), Reconnaissance (85.5%), and Exploits (79.8%), while maintaining 100% precision for detecting normal behaviors. This high precision enhances the real-time performance of IoT systems, especially in hyper-automation processes. Although KNN and AdaBoost performed well in detecting Generic attacks with 100% precision, LAA's overall performance outpaced all other models. Fig. 5 demonstrates that in the NSL-KDD dataset, LAA delivered 95.4% precision, excelling at detecting anomaly attacks (Probe, DoS, U2R, and R2L), while DT achieved 93.1% and KNN reached 91.7%. Other models, such as LR and SVM, achieved precision scores of 90.2% and 89.6%, respectively.

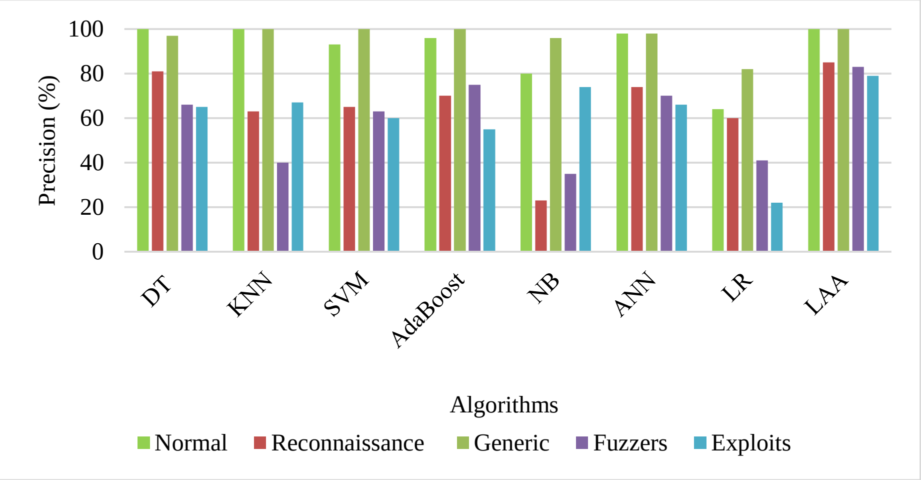

Fig. 4. Precision comparison under UNSW-NB15.

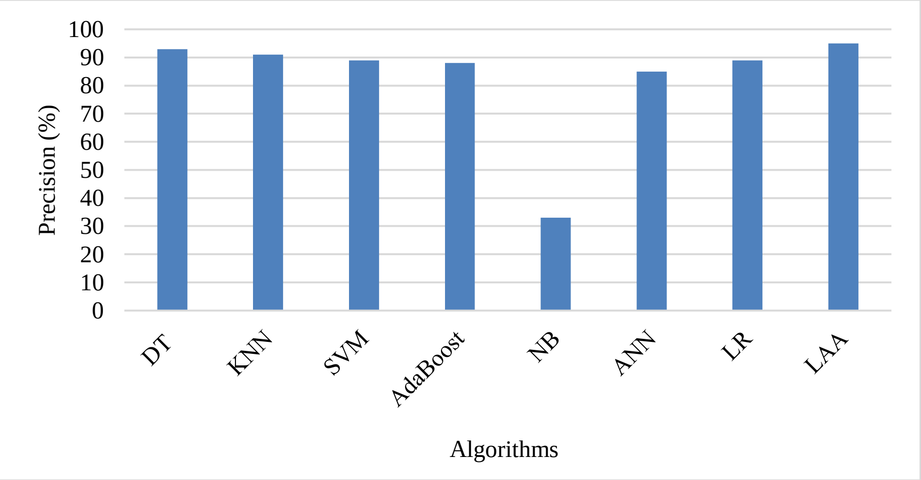

Fig. 5. Precision comparison under NSL-KDD.

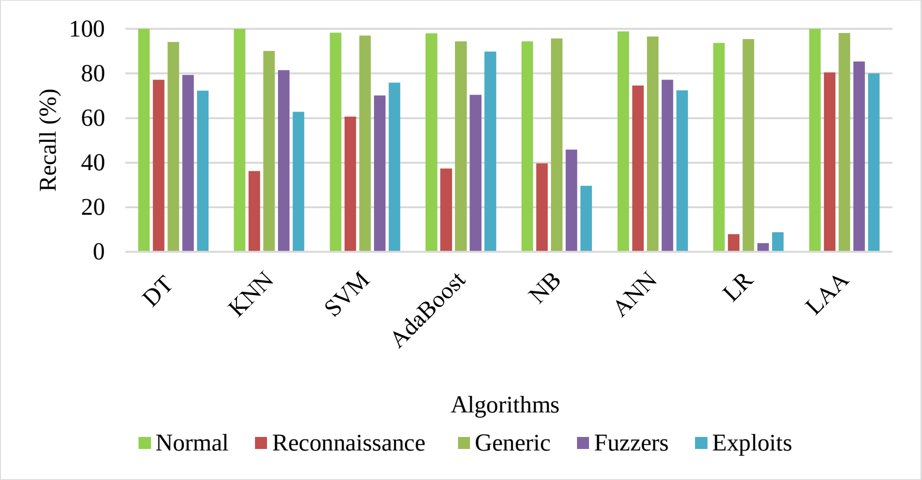

Fig. 6. Recall comparison under UNSW-NB15.

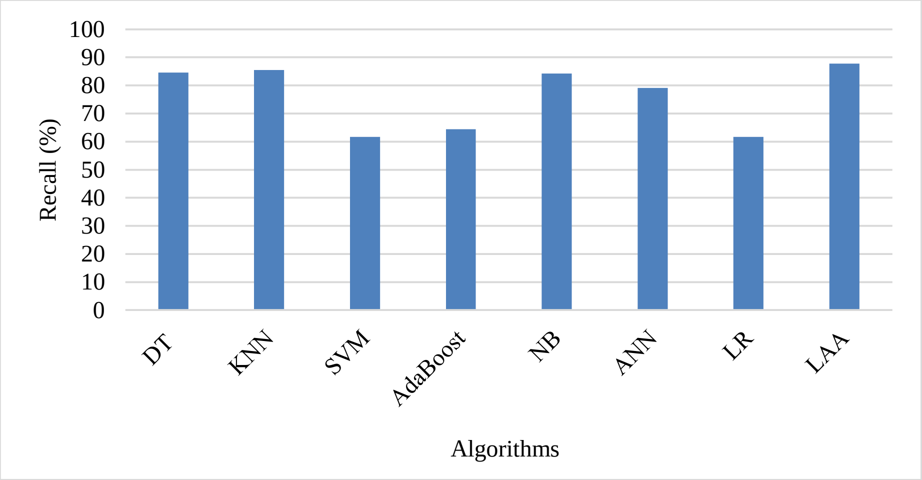

Fig. 7. Recall comparison under NSL-KDD.

Regarding recall, as depicted in Fig. 6, LAA outperformed other machine learning models in detecting cyber-attacks in the UNSW-NB15 dataset. Specifically, LAA achieved 85.4% recall for Fuzzers, 98.2% for Generic, and 80.5% for Reconnaissance attacks. The LAA also demonstrated 100% recall for detecting normal behaviors, ensuring high sensitivity in recognizing benign activities. In Fig. 7, LAA's recall in the NSL-KDD dataset stood at 87.8%, efficiently detecting anomaly categories such as Probe, DoS, U2R, and R2L. Comparatively, KNN achieved a recall of 85.5%, DT reached 84.6%, and NB provided 84.3%. Lower recall values were recorded for models like AdaBoost (64.4%) and SVM (61.7%).

Fig. 8 and Fig. 9 compares the F1-score across different algorithms for both datasets. In the UNSW-NB15 dataset, the LAA attained the highest F1-score of 87.4%, surpassing models such as ANN (83.2%), AdaBoost (82.2%), and SVM (78.9%). Other algorithms like DT (75.8%), KNN (74.1%), and NB (60.7%) recorded lower F1-scores, highlighting the superior performance of the LAA. For the NSL-KDD dataset, LAA again led with an F1-score of 89.2%, while DT and KNN followed with 88.5% and 87.3%, respectively. Models like ANN, LR, and SVM lagged with F1-scores of 82.1%, 73.2%, and 73.1%, respectively. NB exhibited the lowest F1-score at 46.2%.

These findings are consistent with recent studies that emphasize the importance of hybrid optimization in IDS performance. For example, the model proposed by Sanju [17], which integrates metaheuristics with deep learning, also demonstrates competitive F1-scores; however, our method achieved higher precision and recall across both datasets. Similarly, the approach by Almuqren, et al. [18] using Modified Firefly Optimization for botnet detection shows high performance, yet lacks the lightweight and explainable characteristics emphasized in our LAA framework. Compared to the SVM–GWO hybrid by Ghasemi and Babaie [20], our model achieved superior accuracy and a more balanced F1- score, particularly in detecting diverse attack classes under constrained environments.

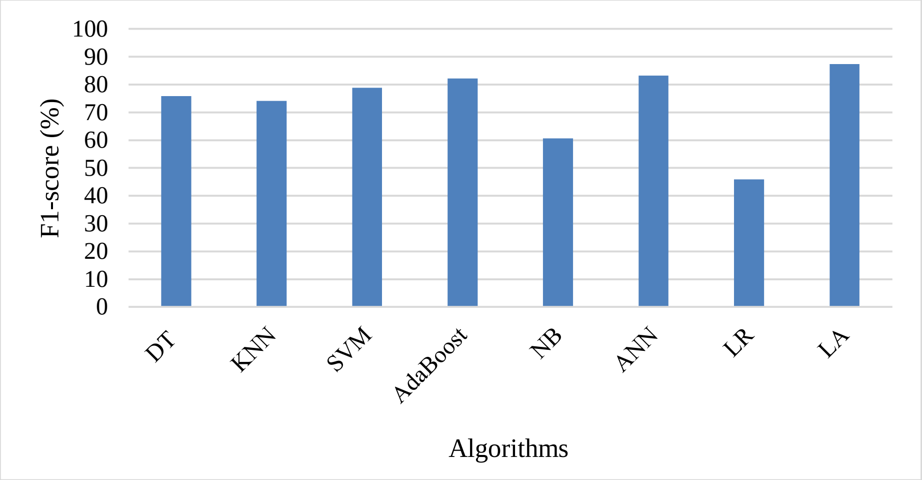

Fig. 8. F1-score comparison under UNSW-NB15.

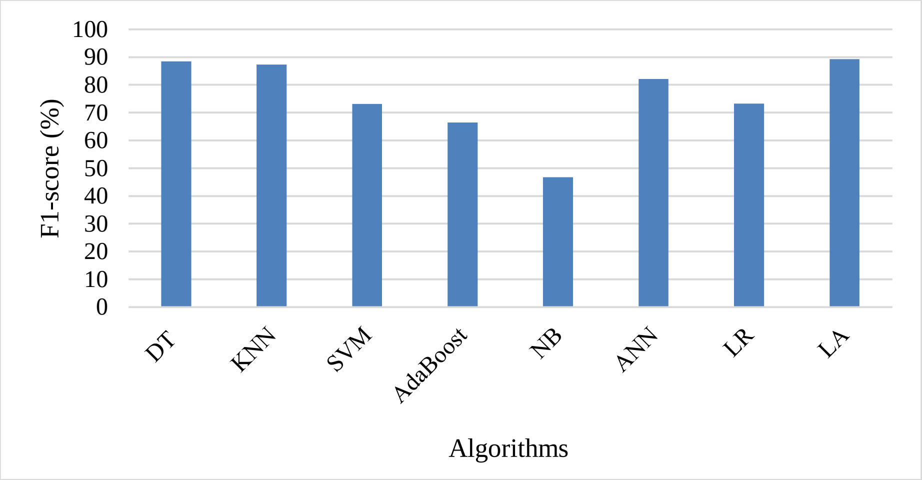

Fig. 9. F1-score comparison under NSL-KDD.

V. Conclusion

In this study, we introduced LAA as a novel and efficient method for enhancing intrusion detection in SD-IoT environments. By integrating LRS with the AOA, the LAA achieved a more dynamic balance between exploration and exploitation, efficiently navigating complex search spaces and identifying optimal solutions for feature selection. The experimental results on the NSL-KDD and UNSW-NB15 datasets demonstrated that the suggested LAA-based IDS model outperformed conventional machine learning algorithms in key performance indicators such as F1-score, recall, precision, and accuracy. The LAA's ability to achieve high detection rates while maintaining computational efficiency makes it particularly well-suited to resource-constrained SD-IoT systems. The proposed LAA presents a significant advancement in intrusion detection, providing a robust, lightweight, and explainable solution for detecting cyber-attacks in SD-IoT environments.

Despite these promising results, the study has certain limitations. First, the performance of the LAA is somewhat dependent on the fine-tuning of algorithmic parameters, which may require domain-specific expertise. Second, although benchmark datasets such as NSL-KDD and UNSW-NB15 ensure comparability, the model's effectiveness on real-world, heterogeneous SD-IoT traffic remains to be assessed. Finally, while the method is computationally efficient in experimental settings, its scalability and real-time performance under production-scale environments require further validation. Future work can explore the extension of LAA to other IoT applications and networks, as well as the development of more advanced hybrid models to further improve detection rates and reduce computational costs.

References

[1] O. Aouedi et al., "A survey on intelligent Internet of Things: Applications, security, privacy, and future directions," IEEE communications surveys & tutorials, 2024, doi: https://doi.org/10.1109/COMST.2024.3430368

[2] B. Pourghebleh, N. Hekmati, Z. Davoudnia, and M. Sadeghi, "A roadmap towards energy - efficient data fusion methods in the Internet of Things," Concurrency and Computation: Practice and Experience, vol. 34, no. 15, p. e6959, 2022, doi: https://doi.org/10.1002/cpe.6959.

[3] A. Rahman et al., "Impacts of blockchain in software - defined Internet of Things ecosystem with Network Function Virtualization for smart applications: Present perspectives and future directions," International Journal of Communication Systems, vol. 38, no. 1, p. e5429, 2023, doi: https://doi.org/10.1002/dac.5429.

[4] R. Huang, X. Yang, and P. Ajay, "Consensus mechanism for software-defined blockchain in internet of things," Internet of Things and Cyber-Physical Systems, vol. 3, pp. 52-60, 2023, doi: https://doi.org/10.1016/j.iotcps.2022.12.004.

[5] P. Kumar, A. Jolfaei, and A. N. Islam, "An enhanced Deep-Learning empowered Threat-Hunting Framework for software-defined Internet of Things," Computers & Security, vol. 148, p. 104109, 2025, doi: https://doi.org/10.1016/j.cose.2024.104109.

[6] B. Bala and S. Behal, "AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges," Computer science review, vol. 52, p. 100631, 2024, doi: https://doi.org/10.1016/j.cosrev.2024.100631.

[7] E. Rivandi and R. Jamili Oskouie, "A Novel Approach for Developing Intrusion Detection Systems in Mobile Social Networks," Available at SSRN 5174811, 2024, doi: https://dx.doi.org/10.2139/ssrn.5174811.

[8] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, "A systematic literature review for network intrusion detection system (IDS)," International journal of information security, vol. 22, no. 5, pp. 1125-1162, 2023, doi: https://doi.org/10.1007/s10207-023-00682-2.

[9] S. Bacha, A. Aljuhani, K. B. Abdellafou, O. Taouali, N. Liouane, and M. Alazab, "Anomaly-based intrusion detection system in IoT using kernel extreme learning machine," Journal of Ambient Intelligence and Humanized Computing, vol. 15, no. 1, pp. 231-242, 2024, doi: https://doi.org/10.1007/s12652-022-03887-w.

[10] N. S. Shaji, R. Muthalagu, and P. M. Pawar, "SD-IIDS: intelligent intrusion detection system for software-defined networks," Multimedia Tools and Applications, vol. 83, no. 4, pp. 11077-11109, 2024, doi: https://doi.org/10.1007/s11042-023-15725-y.

[11] A. Singh, P. K. Chouhan, and G. S. Aujla, "SecureFlow: Knowledge and data-driven ensemble for intrusion detection and dynamic rule configuration in software-defined IoT environment," Ad Hoc Networks, vol. 156, p. 103404, 2024, doi: https://doi.org/10.1016/j.adhoc.2024.103404.

[12] Y. A. Abid, J. Wu, G. Xu, S. Fu, and M. Waqas, "Multi-Level Deep Neural Network for Distributed Denial-of-Service Attack Detection and Classification in Software-Defined Networking Supported Internet of Things Networks," IEEE Internet of Things Journal, vol. 11, no. 14, pp. 24715-24725, 2024, doi: https://doi.org/10.1109/JIOT.2024.3376578.

[13] A. Azadi and M. Momayez, "Simulating a Weak Rock Mass by a Constitutive Model," Mining, vol. 5, no. 2, p. 23, 2025, doi: https://doi.org/10.3390/mining5020023.

[14] M. A. Rahman, A. T. Asyhari, L. Leong, G. Satrya, M. H. Tao, and M. Zolkipli, "Scalable machine learning-based intrusion detection system for IoT-enabled smart cities," Sustainable Cities and Society, vol. 61, p. 102324, 2020, doi: https://doi.org/10.1016/j.scs.2020.102324.

[15] A. Forestiero, "Metaheuristic algorithm for anomaly detection in Internet of Things leveraging on a neural-driven multiagent system," Knowledge-Based Systems, vol. 228, p. 107241, 2021, doi: https://doi.org/10.1016/j.knosys.2021.107241.

[16] Y. Li, S.-m. Ghoreishi, and A. Issakhov, "Improving the accuracy of network intrusion detection system in medical IoT systems through butterfly optimization algorithm," Wireless Personal Communications, vol. 126, no. 3, pp. 1999-2017, 2022, doi: https://doi.org/10.1007/s11277-021-08756-x.

[17] P. Sanju, "Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks," Journal of Engineering Research, vol. 11, no. 4, pp. 356-361, 2023, doi: https://doi.org/10.1016/j.jer.2023.100122.

[18] L. Almuqren, H. Alqahtani, S. S. Aljameel, A. S. Salama, I. Yaseen, and A. A. Alneil, "Hybrid metaheuristics with machine learning based botnet detection in cloud assisted internet of things environment," IEEE Access, vol. 11, pp. 115668-115676, 2023, doi: https://doi.org/10.1109/ACCESS.2023.3322369.

[19] T. Jayasankar, R. Kiruba Buri, and P. Maheswaravenkatesh, "Intrusion detection system using metaheuristic fireworks optimization based feature selection with deep learning on Internet of Things environment," Journal of Forecasting, vol. 43, no. 2, pp. 415-428, 2024, doi: https://doi.org/10.1002/for.3037.

[20] H. Ghasemi and S. Babaie, "A new intrusion detection system based on SVM–GWO algorithms for Internet of Things," Wireless Networks, vol. 30, pp. 2173–2185, 2024, doi: https://doi.org/10.1007/s11276-023-03637-6.

[21] M. B. Bagherabad, E. Rivandi, and M. J. Mehr, "Machine Learning for Analyzing Effects of Various Factors on Business Economic," Authorea Preprints, 2025, doi: https://doi.org/10.36227/techrxiv.174429010.09842200/v1.