Explainable Artificial Intelligence with Single Layer Feedforward Neural Network and Improved Crowned Porcupine Optimization Algorithm for Classification Problems

S. Caxton Emerald
Department of Computer Science, Pondicherry University, Puducherry, India
scaxtonemerald@pondiuni.ac.in (corresponding author)

T. Vengattaraman
Department of Computer Science, Pondicherry University, Puducherry, India
vengattaraman.t@pondiuni.ac.in

Received: 29 December 2024 | Revised: 22 January 2025 and 3 February 2025 | Accepted: 5 February 2025

ABSTRACT

The increasing occurrence of network intrusions calls for the development of advanced Artificial Intelligence (AI) techniques to tackle classification challenges in Intrusion Detection Systems (IDSs). However, the complex decision-making processes of AI often prevent human security professionals from fully understanding the behavior of the model. Explainable AI (XAI) enhances trust in IDSs by providing transparency and assisting professionals in interpreting data and reasoning. This study explores AI techniques that improve both accuracy and interpretability, strengthening trust management in cybersecurity. Integrating performance with explainability improves decision-making and builds confidence in automated systems for classifying network intrusions. This study presents an Explainable Artificial Intelligence Kernel Extreme Learning Machine Improved with the Crowned Porcupine Optimization Algorithm (XAIKELM-ICPOA) approach. Initially, the proposed XAIKELM-ICPOA method preprocesses the data using min-max scaling to ensure uniformity and improve model performance. Next, the Kernel Extreme Learning Machine (KELM) model is employed for classification. The Improved Crowned Porcupine Optimization (ICPO) method is used to optimize KELM hyperparameters, improving classification performance. Finally, SHAP is employed as an XAI technique to provide insights into feature contributions and decision-making processes. The XAIKELM-ICPOA method was evaluated on the NSL-KDD dataset, achieving an accuracy of 96.82%.

Keywords-explainable artificial intelligence; kernel extreme learning machine; intrusion detection; crowned porcupine optimization; min-max scaling

I. INTRODUCTION

In recent years, the increasing frequency of cyber-security attacks has raised alarms around the world [1]. Network IDS (NIDS) monitor traffic to detect potential risks and alert cybersecurity teams [2]. As organizations build more integrated cybersecurity ecosystems, managing trust between people, technology, and processes is important for defending against cyber threats [3]. XAI presents transparency in decision-making, assisting cybersecurity professionals in understanding AI-driven threat detection, unlike conventional black-box AI, thus fostering trust in automated systems [4]. XAI improves decision-making and responses to cyber threats. Furthermore, the rise of XAI models for post-hoc understandability has created new cybersecurity roles that include explainability layers for human-in-the-loop systems [5].

AI and Machine Learning (ML) have advanced across various domains, improving tasks such as strategic games, visual recognition, and daily life applications [6]. These methods, especially Deep Learning (DL), are crucial in science for tasks such as prediction and simulation [7]. The growth of XAI research has led to domain-specific methods for interpreting ML techniques, increasing the popularity of ML and DL in business applications [8]. XAI provides transparency, helping humans understand decisions made by systems [9]. It addresses the need for transparency in ML techniques, explaining black-box models [10]. The motivation for this research is the growing complexity of cyber threats and the increasing need for transparent and explainable systems to allow professionals to make informed decisions and strengthen cybersecurity defenses.

This study presents an Explainable Artificial Intelligence for Kernel Extreme Learning Machine using the Improved Crowned Porcupine Optimization Algorithm (XAIKELM-ICPOA) approach. Initially, the proposed XAIKELM-ICPOA method preprocesses the data using min-max scaling to ensure uniformity and improve model performance. Next, the Kernel Extreme Learning Machine (KELM) model is employed for classification. The Improved Crowned Porcupine Optimization (ICPO) model is used to optimize KELM hyperparameters, enhancing classification performance. Finally, SHAP is employed as an Explainable AI (XAI) technique to provide insights into feature contributions and decision-making processes. The performance of the XAIKELM-ICPOA method was evaluated on the NSL-KDD dataset. The major contributions of the proposed method are as follows:

- Min-max scaling is used to normalize the input data, ensuring that all features fall within a consistent range. This improves the efficiency and accuracy of KELM by eliminating scale-related biases.
- KELM is used to classify cybersecurity threats, integrating kernel methods with ELM to handle complex, nonlinear patterns in data. This allows more accurate detection of diverse and growing attack scenarios.
- ICPO optimizes KELM hyperparameters, improving threat detection performance and efficiency. This also ensures a more accurate and robust classification, specifically in dynamic cybersecurity environments.
- SHAP is used as an XAI technique to provide detailed insights into the contributions of features and decision-making processes of the model. This improves transparency and trust in model predictions.
- The novelty of this approach lies in the integration of ICPO to optimize the KELM hyperparameters and SHAP to provide explainability, providing a model that is both highly accurate and transparent. This incorporation ensures a robust and interpretable system for detecting cybersecurity threats.

II. RELATED WORKS

In [11], DL was utilized for intrusion detection. A filter-based method was used to detect key features and reduce complexity, with dual DL techniques, namely DNN and CNN, applied to the dataset. XAI was used to explain the techniques, using LIME for DNN transparency and SHAP for additional insights. In [12], the XAI Enabled Intrusion Detection Model for Secure Cyber-Physical Systems (XAIID-SCPS) model was proposed, using the Hybrid Enhanced Glow Worm Swarm Optimizer (HEGSO) for feature selection, Improved Elman Neural Network (IENN) with an Enhanced Fruitfly Optimizer (EFFO) for intrusion detection, and LIME to improve model interpretability. In [13], XAI methods were used for local and global explanations and feature extraction to identify key intrusion aspects. In [14], an IDS used XAI and ML models to classify cyberattacks, utilizing Apache Spark, Kafka, SHAP, and Scikit-learn, with XAI providing a rationale for each classification. In [15], a Hybrid Adaptive Ensemble for Intrusion Detection (HAEnID) technique was proposed, utilizing Stacking Ensemble (SEM), Conditional Ensemble Method (CEM), and Bayesian Model Averaging (BMA) models. In [16], XAI was integrated into an ML-based IDS, using SHAP for global explanations and reevaluating low-credit results with subsequent classifiers. In [17], an end-to-end XAI structure for NIDS was proposed, evaluating global and local explanations using LIME and SHAP with six diverse metrics.

III. MATERIALS AND METHODS

The main intention of the XAIKELM-ICPOA method is to provide a robust intrusion detection framework that integrates XAI with advanced optimization techniques. Figure 1 presents the workflow of the XAIKELM-ICPOA model.

A. Stage I: Data Normalization

Primarily, the proposed XAIKELM-ICPOA method utilizes a data preprocessing stage with min-max scaling to ensure uniformity and enhance model performance. Normalization is a significant preprocessing stage that ensures that every feature contributes similarly to the method [18]. It prevents features with additional wide ranges from overlooking the learning procedure. Min-max scaling converts all features within the range of $[0, 1]$:

$$\tilde{x}_{t,j} = \frac{x_{t,j} - min(x_{:,j})}{max(x_{:,j}) - min(x_{:,j})} \tag{1}$$

This scaling is beneficial when the data does not have important outliers and the features are almost equally spread.

B. Stage II: KELM-based Classification Process

Next, the KELM model, a recent approach, is employed for the classification process [19]. ELM, proven effective in many real-world applications, is used for single-hidden layer feed-forward Neural Networks (NNs). It operates in generalized SLFNs without fine-tuning the hidden layer parameters. The output function for ELM in general SLFNs is given by:

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = h(x)\beta \tag{2}$$

$$min\,||H\beta - T||^2 \; and \; ||\beta|| \tag{3}$$

$$H = [h(x_1) h(x_2) h(x_3)] = \begin{pmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{pmatrix} \tag{4}$$

$$\beta = H^{\dagger} T \tag{5}$$

$$\Omega_{ELM} = HH^T : \Omega_{ELM_{(i,j)}} = h(x_i) \cdot h(x_j) = K(x_i, x_j) \tag{6}$$

where β is the output weighting vector and $h(x)$ is the hidden layer output for input x. ELM minimizes training error and output weight norm (3). The matrix H in (4) represents the hidden layer outputs, and β is calculated using the least squares method in (5). The kernel matrix Ω_{ELM} is defined in (6) for unknown feature maps or multiple classes. $h(x)$ refers to a functional map that ensures the data is linearly independent in the hidden layer feature space H. The orthogonal projection model calculates the Moore-Penrose generalized inverse of the matrix, $H^{\dagger} = H^T(HH^T)^{-1}$, and a positive constant C is added to the diagonal of HH^T. Finally, the output operation of the ELM is defined as:

$$F(x) = h\beta = h(x)H^{\dagger}\left(\frac{1}{C} + HH^{\dagger}\right)^{-1} T =$$

$$= [K(x, x_1) \ \dots \ K(x, x_N)\ ]^T \left(\frac{1}{C} + \Omega_{ELM}\right)^{-1} T \qquad (7)$$

The HL feature maps can include various kernels, with the Radial Basis Function (RBF) kernel being commonly used. The RBF kernel is defined as $K(x, x_i) = exp(|x - x_i||^2)$, where γ and penalty parameter C are key parameters in the kernel.

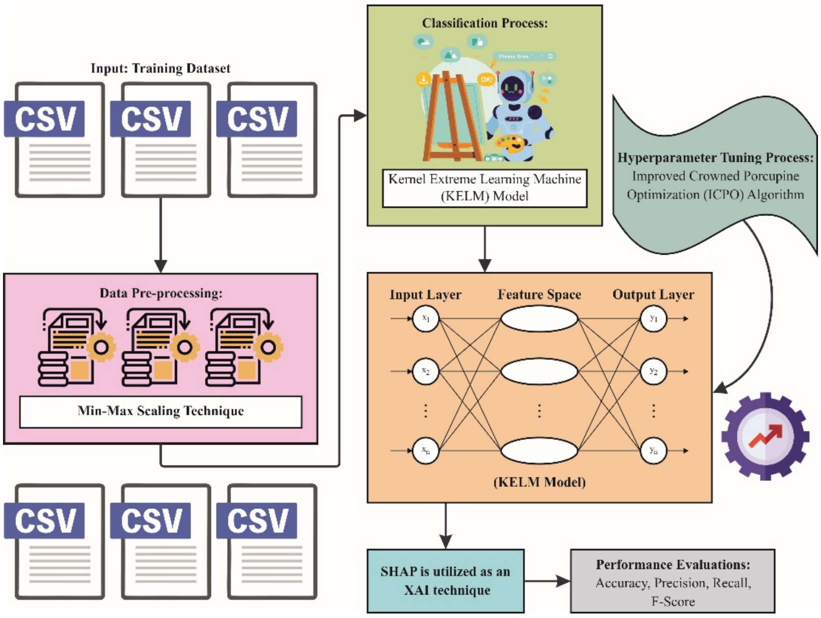
Input: Training Dataset
CSV
CSV
CSV
Data Pre-processing:
Min-Max Scaling Technique
CSV
CSV
CSV
Classification Process:
Kernel Extreme Learning Machine (KELM) Model
Hyperparameter Tuning Process:
Improved Crowned Porcupine Optimization (ICPO) Algorithm
Input Layer
Feature Space
Output Layer
x1
x2
xn
y1
y2
yn
(KELM Model)
SHAP is utilized as an XAI technique
Performance Evaluations:
Accuracy, Precision, Recall, F-Score

Fig. 1. Workflow of the XAIKELM-ICPOA model.

C. Stage III: ICPO-based Parameter Tuning

ICPO [20] is employed to optimize the hyperparameters of KELM and achieve superior classification performance. The CPO model is an optimization approach inspired by the defensive behaviors of crowned porcupines. It uses four defense mechanisms: auditory (warning sounds), visual (sharp spines), physical (spines for conflict), and olfactory (foul smells). These behaviors are divided into two main phases: exploitation and exploration.

1) Exploration Phase

The CPO model explores solutions using porcupine defense mechanisms. The first defense approach mimics the porcupine raising its feathers to deter predators, represented by:

$$x_i^{\overrightarrow{t+1}} = \overrightarrow{x_i^t} + \tau_1 \times\!\rightarrow \left|2 \times \tau_2 \times \overrightarrow{x_{CP}^t} - \overrightarrow{y_i^t}\right| \qquad (8)$$

where $\overrightarrow{x_{CP}^t}$ is the best solution at iteration t, τ_1 is a random number from a standard distribution, τ_2 is a value in $[0, 1]$, and $\overrightarrow{y_i^t}$ is the average of $\overrightarrow{x_i^t}$ and $\overrightarrow{x_r^t}$, with r being a random index in the population. The second defense simulates the porcupine producing alert sounds as:

$$x_i^{\overrightarrow{t+1}} =$$

$$\left(1 - \overrightarrow{U_1}\right) \times x_i^t + \vec{U}_1 \times\!\rightarrow \left(\vec{y} + \tau_3 \times \left(\overrightarrow{x_{r_1}^t} - \overrightarrow{x_{r_2}^t}\right)\right) \qquad (9)$$

where $\vec{U}_1$ refers to a binary randomly generated vector using values of 0 or 1, r_1 and r_2 are dual randomly formed integers amongst $[1, N]$, with N denoting the population size. τ_3 signifies a randomly generated value between $(0,1)$.

2) Exploitation Phase

In the exploitation phase, the model uses physical and odor-based defense mechanisms. The third defense approach mimics the porcupine releasing foul odors, represented by:

$$x_i^{\overrightarrow{t+1}} = \left(1 - \vec{U}_1\right) \times \overrightarrow{x_i^t} + \vec{U}_1 \times\!\rightarrow$$

$$\left(\overrightarrow{x_{r_1}^t} + \overrightarrow{S_i^t} \times \left(\overrightarrow{x_{r_2}^t} - \overrightarrow{x_{r_3}^t}\right) - \tau_3 \times \vec{\delta} \times \gamma_t \times S_i^t\right) \qquad (10)$$

where $\overrightarrow{S_i^t}$ characterizes the odor diffusion feature, γ_t denotes the defense feature, and $\vec{\delta}$ influences the search direction. The fourth defense simulates a physical attack by the porcupine, as shown in:

$$x_i^{\overrightarrow{t+1}} = \overrightarrow{x_{CP}^t} + (\alpha(1 - \tau_4) + \tau_4) \rightarrow\!\times \left(\delta \times \overrightarrow{x_{CP}^t} - \overrightarrow{x_i^t}\right) -$$

$$\tau_5 \times \delta \times \gamma_t \times \overrightarrow{F_i^t} \qquad (11)$$

where α is the convergence speed factor, and τ_4 and τ_5 are random values in $[0, 1]$. $\overrightarrow{F_i^t}$ is the average strength applied to predators. The CPO model faces slower convergence and computational challenges due to its reliance on exploration. To address these challenges, two key improvements were made.

3) Logistic Chaotic Mapping (LCM) for Population Initialization

To address CPO's limited search space, this study uses LCM for population initialization. LCM is a simple, dynamic 1D nonlinear method, defined by:

$$x_{n+1} = \mu x_n(1 - x_n) \qquad (12)$$

where $\mu = 4$, and x_0 is randomly generated in $(0, 1)$, iterating 100 times to determine the model parameters.

4) Elite Preservation Approach

To address CPO's limitation in losing high-quality solutions, this study introduces an elite preservation approach. The elite set E_t at an iteration t is defined by:

$$E_t = f(x) \leq f(y)$$

$$\text{for all } y \in P_t \setminus E_t \text{ and } |E_t| = \lceil r \cdot |P_t| \rceil \qquad (13)$$

where P_t is the population at iteration t, $|P_t|$ is its size, x_i^t is the i^{th} individual, and $f(x)$ is the Fitness Function (FF). r is the elite ratio, and σ ranks the population based on fitness:

$$\sigma(P_t) = \left(x_{(1)}^t, x_{(2)}^t, \ldots, x_{(|P_t|)}^t\right) \qquad (14)$$

Now, $f(x_{(i)}^t) \leq f(x_{(j)}^t)$ for every $i < j$. The elite set is defined in (15), and the updated population is given by (16):

$$E_t = x_{(i)}^t | 1 \leq i \leq \lceil r \cdot |P_t| \rceil \qquad (15)$$

$$P_{t+1} = E_t \cup S_t \qquad (16)$$

where S_t epitomizes a solution set produced by the normal ICPO processes, and $|S_t| = |P_t| - |E_t|$. The particular phases are given below:

i) Initializing population P_0.

ii) For all iterations t:

(a) Estimate the fitness $f(x)$ for every $x \in P_t$.

(b) Sort the population: (P_t).

(c) Select the elite set: $E_t = x_{(i)}^t | 1 \leq i \leq \lceil r \cdot |P_t| \rceil$.

(d) Generate novel solution: Update S_t utilizing ICPO.

(e) Update the population: $P_{t+1} = E_t \cup S_t$.

iii) Repeat the above-mentioned phases till the termination conditions are encountered.

The ICPO technique generates an FF to improve classification performance. It uses a positive value to indicate a better outcome for the candidate solution. In this context, the FF is based on minimizing the classifier's error rate. Its mathematical formulation is given by:

$$fitness(x_i) = ClassifierErrorRate(x_i) = \\ = \frac{no.of\ misclassified\ samples}{Total\ no.of\ samples} \times 100 \qquad (17)$$

D. Stage IV: XAI-based SHAP

Finally, SHAP is utilized as an XAI technique to provide insights into feature contributions and decision-making processes [21]. SHAP integrates Shapley values and LIME, combining a solid theoretical foundation for interpreting black-box models. LIME fits an interpretable model around a specific sample, contrasting with global substitution models. It focuses on explaining important attributes of a data sample for better predictions. The description of an observation x in LIME is given by:

$$E(x) = L(f, g, d^x) + \Omega(g) \qquad (18)$$

In LIME, g is an interpretable model used to explain black-box predictions, with $\Omega(g)$ representing its complexity. The loss function $L(f, g, d^x)$ measures the difference between the surrogate model and the original model. LIME creates synthetic data to reduce complexity, while Shapley values, from game theory, measure each feature's contribution to predictions, ensuring fairness and significance.

$$\varphi_j(val) = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!(m-|S|-1)!}{m!} [val(S \cup \{j\}) - \\ val(S)], \quad j = 1 \ldots m \qquad (19)$$

In Shapley values, S is a subset of features, $|S|$ is the number of features in S, and M is the total feature set. j represents a specific feature, with $val(S)$ and $val(S + j)$ indicating the model outputs for the subsets of features S and $S \cup \{j\}$, respectively. The Shapley value includes four properties: Dummy, Efficiency, Additivity, and Symmetry. Efficiency ensures feature contributions match the difference between the average and actual predictions.

$$\sum_{i=1}^{m} \varphi_i(val) = val(M) \qquad (20)$$

- Symmetry: if features i^{th} and k^{th} are similar in subset S, then:

  $$val(S + i) = val(S + k), S \subseteq M \setminus \{i, k\} \qquad (21)$$

  Equation (21) specifies that the dual features i^{th} and k^{th} give as just like promising coalitions. As a result, the support of dual features must be equivalent and the function φ is symmetrical:

  $$\varphi_i(val) = \varphi_k(val) \qquad (22)$$

- Dummy: If a feature does not affect the prediction, its Shapley value is 0:

  $$val(S + i) = ual(S), \ S \subseteq M \setminus \{i\} \qquad (23)$$

  Then $\varphi_i(val) = 0$ (24)

- Additivity: the improvement by a combination of dual functions val and val' is equivalent to the calculation of individual achievements from all functions for all features i^{th}:

  $$\varphi_i(val + val') = \varphi_i(val) + \varphi_i(val') \qquad (25)$$

SHAP provides a unified model for explaining black-box predictions. It connects Shapley values and LIME via a linear approach. For a sample x, the explanation is:

$$g(z') = \varphi_0 + \sum_{i=1}^{m} \varphi_i z_i', \quad z' \in \{0,1\}^m \qquad (26)$$

In SHAP, m is the coalition size, g is the explanation method, z' are streamlined attributes, and φ_i is the Shapley value for the i^{th} feature. SHAP spreads effects more evenly than LIME, which weighs samples based on their distance to the original. SHAP uses the SHAP kernel for weighting.

$$\pi_x(z') = \frac{(m-1)}{(m\,|z'|\,)}\ |z'|(m-|z'|) \tag{27}$$

The SHAP value calculation for a sample x involves several stages. First, sample data $z'_k \in \{0,1\}^m$ are collected. Next, predictions are obtained for all z'_k by transforming them to the feature space and applying the model $f(h_x(z'_k))$, where $h_x: \{0,1\}^m \to R$. The weight for each z'_k is calculated using (27). Then, a linear model g is trained by minimizing the loss function L using training data Z. Finally, the Shapley values and coefficients are returned based on the results of the linear model. The loss function is given by:

$$L(f,g,\pi_x) = \sum_{z'\in Z}\left[f\big(h_x(z')\big) - g(z')\right]^2 \pi_x(z') \tag{28}$$

IV. PERFORMANCE VALIDATION

The performance of the XAIKELM-ICPOA approach was evaluated using the NSL-KDD dataset [22-23]. The dataset comprises 125973 rows under two class labels, such as normal and anomaly, as shown in Table I. The proposed technique was simulated using Python 3.6.5 on a PC with an i5-8600k, 250GB SSD, GeForce 1050Ti 4GB, 16GB RAM, and 1TB HDD. The parameter settings were: learning rate: 0.01, activation: ReLU, epoch count: 50, dropout: 0.5, and batch size: 5.

TABLE I. DATASET DETAILS

Labels	**Counts**
Normal	67343
Anomaly	58630
Total	**125973**

TABLE II. RESULTS OF XAIKELM-ICPOA

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
XAIKELM-ICPOA	96.82	96.86	96.75	96.80

Compared to existing techniques in [12, 24], the proposed XAIKELM-ICPOA method shows promising results, achieving $accu_y$ of 96.82%, $prec_n$ of 96.86%, $reca_l$ of 96.75%, and F_{score} of 96.80%.

V. CONCLUSION

This study presented the XAIKELM-ICPOA method, which is a robust intrusion detection framework that integrates XAI with advanced optimization techniques. A data preprocessing stage with min-max scaling was used to ensure uniformity and enhance model performance. The KELM method was employed for classification, while the ICPO method optimized KELM's hyperparameters for superior performance. Finally, SHAP was used as an XAI technique to provide insights into feature contributions and decision-making. The XAIKELM-ICPOA method was evaluated using the NSL-KDD dataset. The performance validation of the XAIKELM-ICPOA method illustrated an accuracy of 96.82%. The limitations of the XAIKELM-ICPOA model include its reliance on a specific dataset, limiting generalizability, and challenges with scalability due to computational complexity. Future work will focus on using diverse datasets, real-time data integration, and enhancing efficiency for large-scale systems.

REFERENCES

[1] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, and P. M. Atkinson, "Explainable artificial intelligence: an analytical review," *WIREs Data Mining and Knowledge Discovery*, vol. 11, no. 5, 2021, Art. no. e1424, https://doi.org/10.1002/widm.1424.

[2] B. Kovalerchuk and E. McCoy, "Explainable Machine Learning for Categorical and Mixed Data with Lossless Visualization," in *Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery*, B. Kovalerchuk, K. Nazemi, R. Andonie, N. Datia, and E. Banissi, Eds. Springer Nature Switzerland, 2024, pp. 73–123.

[3] C. I. Nwakanma *et al.*, "Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review," *Applied Sciences*, vol. 13, no. 3, Jan. 2023, Art. no. 1252, https://doi.org/10.3390/app13031252.

[4] D. Minh, H. X. Wang, Y. F. Li, and T. N. Nguyen, "Explainable artificial intelligence: a comprehensive review," *Artificial Intelligence Review*, vol. 55, no. 5, pp. 3503–3568, Jun. 2022, https://doi.org/10.1007/s10462-021-10088-y.

[5] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "XAI—Explainable artificial intelligence," *Science Robotics*, vol. 4, no. 37, Dec. 2019, Art. no. eaay7120, https://doi.org/10.1126/scirobotics.aay7120.

[6] R. Confalonieri, L. Coba, B. Wagner, and T. R. Besold, "A historical perspective of explainable Artificial Intelligence," *WIREs Data Mining and Knowledge Discovery*, vol. 11, no. 1, 2021, Art. no. e1391, https://doi.org/10.1002/widm.1391.

[7] F. K. Dosilovic, M. Brcic, and N. Hlupic, "Explainable artificial intelligence: A survey," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, May 2018, pp. 0210–0215, https://doi.org/10.23919/MIPRO.2018.8400040.

[8] K. A. B. Hamou, Z. Jarir, and S. Elfirdoussi, "Application of LightGBM Algorithm in Production Scheduling Optimization on Non-Identical Parallel Machines," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17973–17978, Dec. 2024, https://doi.org/10.48084/etasr.8779.

[9] E. Ramakrishna, J. Gadhamappagari, and P. Sujatha, "Auto tuning of PI Gains using Cuttlefish Optimization for DC Link Voltage Control in a 5-level HB MMC D-STATCOM," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12086–12091, Dec. 2023, https://doi.org/10.48084/etasr.6413.

[10] S. Phimphisan and N. Sriwiboon, "A Customized CNN Architecture with CLAHE for Multi-Stage Diabetic Retinopathy Classification," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18258–18263, Dec. 2024, https://doi.org/10.48084/etasr.8932.

[11] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach," *Expert Systems with Applications*, vol. 238, Mar. 2024, Art. no. 121751, https://doi.org/10.1016/j.eswa.2023.121751.

[12] L. Almuqren, M. S. Maashi, M. Alamgeer, H. Mohsen, M. A. Hamza, and A. A. Abdelmageed, "Explainable Artificial Intelligence Enabled Intrusion Detection Technique for Secure Cyber-Physical Systems," *Applied Sciences*, vol. 13, no. 5, Jan. 2023, Art. no. 3081, https://doi.org/10.3390/app13053081.

[13] O. Arreche, T. Guntur, and M. Abdallah, "XAI-IDS: Toward Proposing an Explainable Artificial Intelligence Framework for Enhancing Network Intrusion Detection Systems," *Applied Sciences*, vol. 14, no. 10, Jan. 2024, Art. no. 4170, https://doi.org/10.3390/app14104170.

[14] X. Larriva-Novo, C. Sánchez-Zas, V. A. Villagrá, A. Marín-Lopez, and J. Berrocal, "Leveraging Explainable Artificial Intelligence in Real-Time Cyberattack Identification: Intrusion Detection System Approach," *Applied Sciences*, vol. 13, no. 15, Jan. 2023, Art. no. 8587, https://doi.org/10.3390/app13158587.

[15] U. Ahmed *et al.*, "Explainable AI-based innovative hybrid ensemble model for intrusion detection," *Journal of Cloud Computing*, vol. 13, no. 1, Oct. 2024, Art. no. 150, https://doi.org/10.1186/s13677-024-00712-x.

[16] S. Wali and I. Khan, "Explainable AI and Random Forest Based Reliable Intrusion Detection system." Dec. 18, 2021, https://doi.org/10.36227/techrxiv.17169080.v1.

[17] O. Arreche, T. R. Guntur, J. W. Roberts, and M. Abdallah, "E-XAI: Evaluating Black-Box Explainable AI Frameworks for Network Intrusion Detection," *IEEE Access*, vol. 12, pp. 23954–23988, 2024, https://doi.org/10.1109/ACCESS.2024.3365140.

[18] B. B. Gupta *et al.*, "Advance drought prediction through rainfall forecasting with hybrid deep learning model," *Scientific Reports*, vol. 14, no. 1, Dec. 2024, Art. no. 30459, https://doi.org/10.1038/s41598-024-80099-6.

[19] M. Wang *et al.*, "Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction," *Engineering Applications of Artificial Intelligence*, vol. 63, pp. 54–68, Aug. 2017, https://doi.org/10.1016/j.engappai.2017.05.003.

[20] W. Lei, Y. Gu, and J. Huang, "An Enhanced Crowned Porcupine Optimization Algorithm Based on Multiple Improvement Strategies," *Applied Sciences*, vol. 14, no. 23, Jan. 2024, Art. no. 11414, https://doi.org/10.3390/app142311414.

[21] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in IoT networks," *Information Sciences*, vol. 639, Aug. 2023, Art. no. 119000, https://doi.org/10.1016/j.ins.2023.119000.

[22] "NSL-KDD." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd.

[23] W. F. S. Stolfo *et al.*, "KDD Cup 1999 Data." UCI Machine Learning Repository, 1999, https://doi.org/10.24432/C51C7N.

[24] N. Omer, A. H. Samak, A. I. Taloba, and R. M. Abd El-Aziz, "A novel optimized probabilistic neural network approach for intrusion detection and categorization," *Alexandria Engineering Journal*, vol. 72, pp. 351–361, Jun. 2023, https://doi.org/10.1016/j.aej.2023.03.093.