V International Scientific and Technical Conference Actual Issues of Power Supply Systems

ANN2IF: A Hybrid Machine Learning Model for Network Traffic Anomaly Detection

AIPCP25-CF-ICAIPSS2025-00262 | Article

ANN2IF: A Hybrid Machine Learning Model for Network Traffic Anomaly Detection

Xusnutdin Samarov, Zakhro Barotova [a)]

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi State University, Tashkent, Uzbekistan

[a)] *Corresponding author: zahrobarotova09@gmail.com*

Abstract. In today's cyber-threat environment, IDS (Intrusion Detection Systems) systems play an important role in ensuring and protecting network security. In this paper, the main focus is on improving the accuracy of detecting and classifying multi-class attacks, and a hybrid Artificial Neural Network+Isolation Forest (ANN2IF) model is presented. The proposed model in the research work aims to effectively combine supervised and unsupervised learning capabilities and includes two main components: ANN for feature extraction and IF algorithms for anomaly detection. The proposed model achieved high accuracy when tested, which proved that it is much more effective in detecting anomalies than using only IF or ANN algorithms. In addition, this model achieved 0.7% FAR (False Alarm Rate) and 0.996 AUC (Area Under the Curve), which indicates a high ability to distinguish normal and malicious traffic in network traffic. These results confirm that the proposed solution is reliable and effective in detecting anomalies in network structures.

INTRODUCTION

In the rapidly developing digital world, network systems are constantly becoming a target for cybercriminals. The scope of damage caused by various cyberattacks and stealth attacks is increasing. Examples of these include data corruption, theft and destruction, and the resulting increase in financial losses. With the rapid growth of network traffic, Intrusion Detection Systems are integrating various advanced methods to protect themselves. Anomaly detection is increasingly being considered an important component in protecting modern network infrastructures. Unlike traditional signature-based methods, anomaly-based detection methods can detect deviations from typical traffic patterns, allowing them to detect zero-day attacks. However, a disadvantage of these methods is their high false positive rate, which can reduce their effectiveness. To solve this problem, machine learning (ML) and deep learning (DL)-based technical approaches have emerged as a focus of recent research. Hybrid models that combine the strengths and capabilities of multiple algorithms have been recognized as the most effective solution for network traffic analysis.

In this research work, a new hybrid Artificial Neural Network+Isolation Forest approach is proposed to detect network traffic anomalies, which takes advantage of the strong feature extraction capabilities of ANN and the efficiency of IF in anomaly detection. The goal of this approach is to significantly reduce the False Alarm Rate (FAR) while improving the detection and classification accuracy.

RELATED WORKS

In recent years, anomaly detection research has increasingly focused on leveraging data characteristics such as distance, density, and probability [1]. Studies in [2], [3], and [4] have proposed image-based anomaly detection models, where deep learning approaches have played a significant role. More recent works [5], [6], [7] have introduced practical anomaly detection tools that are notably accurate and reliable in terms of performance.

Isolation Forest (IF) is considered one of the most effective methods for detecting anomalies in large-scale datasets [8], as it identifies anomalies based on their uniqueness and their ability to deviate from normal instances. However,

the performance of IF can degrade when features are noisy or when data is sparse. This limitation provides a strong motivation to integrate feature extraction methods to inhance the efficiency and robustness of anomaly detection.

Hongzuo Xu [9] proposed the using of Deep Isolation Forest (DIF) for anomaly detection, enhanced by feature mapping through Artificial Neural Networks (ANN), thereby increasing effectiveness in detecting subtle and ambiguous anomalies.

Kumar et al. [10] introduced the ARLIF-IDS model, which improves the Isolation Forest algorithm by incorporating an attention mechanism. This approach ensures low memory consumption and minimal latency.

Elsaid et al. [11] proposed an optimized IF-based approach, OIFIDS, for application in Industrial Internet of Things (IIoT) environments. While this method is well-adapted for efficient real-time stream data processing, it lacks adaptive capabilities for changing network conditions.

TABLE 1. Comparative analysis of the studied literature

References	Model	Main Approach	Dataset	Accuracy / F1-Score	FAR	Limitations	Future Work
[9]	Deep Isolation Forest	NN-based feature mapping + IF	Tabular, graph, time-series	Significant improvement over standard IF	Lower than baseline IF	Requires large training data for NN mapping; higher computation than plain IF	Optimize neural mapping for low-resource devices; extend to streaming data
[10]	ARLIF-IDS	Attention-augmented real-time IF	NSL-KDD, KDDCUP'99	F1 ≈ 0.93	Low latency	Limited to benchmark datasets; no deep feature extraction	Test on real-world network traffic; integrate deep learning feature extractor
[11]	OIFIDS	Optimized IF for streaming IIoT	IIoT, heterogeneous streaming data	Comparable to SOTA	Optimized for streaming	May not generalize well beyond IIoT data	Adapt for general enterprise network traffic; combine with DL features
[12]	Hybrid RF + Autoencoder	RF → AE filtering	NSL-KDD, UNSW-NB15, CIC-DDoS2019	≈ 99%+ accuracy	Very low	Training time high; AE reconstruction errors on noisy data	Replace AE with more robust DL encoder; real-time deployment optimization
[13]	Standalone Isolation Forest	Basic IF	CIC-IoT, ISCX	F1 up to ~100% (varies)	Variable	Sensitive to noisy features; lacks feature learning	-

Zhang [12] proposed a two-stage model (combination of Random Forest and Autoencoder), where the Random Forest (RF) is first used for detection, followed by an Autoencoder to filter misclassified cases, achieving a very low False Alarm Rate (FAR).

Kumar et al. [13] demonstrated that applying the Isolation Forest algorithm alone can achieve high F1-scores in certain scenarios; however, its performance declines noticeably when handling data with complex features. A comparative analysis of the above-reviewed literature is summarized in TABLE 1.

Based on the above research, the proposed ANN2IF model suggests using artificial neural networks for targeted and optimized feature extraction, followed by anomaly detection using Isolation Forest. This approach combines high accuracy with low FAR, addressing existing restrictions of models.

EXPERIMENTAL RESEARCH

We offer a hybrid anomaly detection approach, named ANN2IF, that combines the feature extraction capabilities of Artifical Neural Networks (ANN) with the anomaly isolation stucture of the Isolation Forest (IF) in this section. This primary objective is to detect anomalies network traffic efficiently while minimizing False Alarm Rate (FAR).

The ANN component processes raw network traffic features and transformers them into low-dimensional latent vectors that capture comlex patterns and non-linear dependencies. These latent vectors are then fed into an IF, which isolates anomalies based on their rarity and deviation from normal patterns.

This model includes six sequences:

1. Data gathering
2. Data preprocessing
3. Anomaly Score estimation via ANN
4. IF based deep analysis
5. Final Classification
6. Evaluation.

The general process of these steps is presented in FIGURE 1.

Data collection. In this phase, the raw network traffic data is transformed into a structured and meaningful representation suitable for training the hybrid ANN2IF model. The process consists of three major steps: data cleaning, normalization, and feature selection

Cleaning. Raw network traffic often contains missing values, redundant entries, or inconsistent formats. These artifacts negatively impact the learning process. We first apply the following cleaning steps:

- Remove instances with missing or null values.
- Eliminate duplicate records.
- Encode categorical features using one-hot encoding

Let the raw dataset be denoted by $D = \{x_1, x_2, \dots, x_n\}$, where $x_i \in R^d$ is a feature vector. After cleaning, we obtain the set $D' \subseteq$ D, such that:

$$D' = \{x_i \in D | x_i \; is \; complete \; and \; non \; reduntant\}$$

Normalization

A min-max normalization is used to provide that all features contribute equally to the model:

$$x_i^j = \frac{x_i^j - \min(x^j)}{\max(x^j) - \min(x^j)} \quad (1)$$

where x_i^j is the value of the j-th feature of the i-th instance.

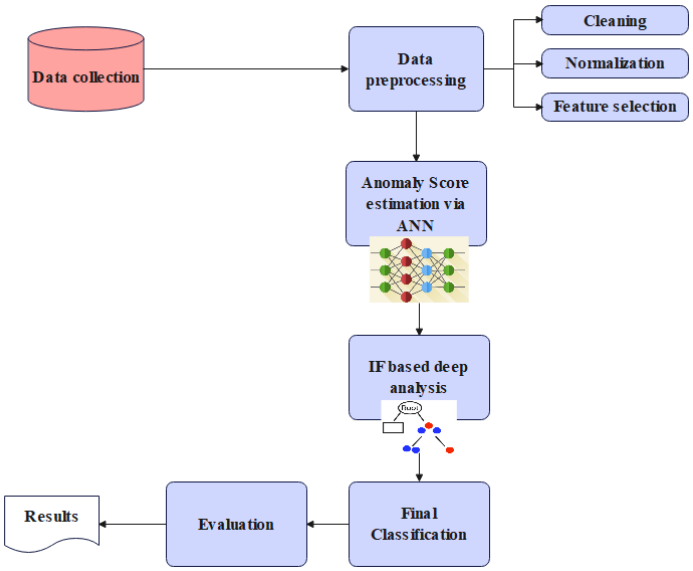

FIGURE 1. The overall procedure of ANN2IF model

Feature selection. Feature selection is critical for eliminating insignificant or excresent features, thereby enhancing detection performance and reducing computational complexity.

We employed the Hybrid Enhanced Glowworm Swarm Optimization (HEGSO) [14] algorithm to select an optimal subset of features that maximizes anomaly detection performance while minimizing dimensionality. Enhancements in HEGSO include:

- Hybrid fitness function combining classification accuracy and feature reduction ratio.
- Dynamic neighborhood radius for improved exploration–exploitation balance.
- Adaptive step size for faster convergence.

The mathematical formulation of HEGSO Algorithm is expressed in the following stages:

Step 1. Encoding&Initialization

Let the full feature set be $F = \{f_1, \dots, f_m\}$. Each glowworm *i* represents a candidate feature subset via a binary vector:

$$X_i = [c_1, \dots, c_m], \quad c_m \in \{0,1\} \tag{2}$$

Initialize luciferin: $L_i(0) = L_0$.

Step 2. Fitness Evaluation

HEGSO employs a multi-objective fitness:

$$Fit(X_i) = \alpha * Accuracy(X_i) - \beta * \frac{|X_i|}{m} \tag{3}$$

where α+β=1, balancing accuracy and compactness.

Step 3. Luciferin Update

Each agent updates luciferin levels over iterations:

$$L_i(t+1) = (1-p) * L_i(t) + \gamma * Fit(X_i \tag{4}$$

with ρ as decay constant and γ as enhancement factor.

Step 4. Neighbor Selection

Glowworms move towards neighbors with higher luciferin within a dynamic radius $r_d^i(t)$:

$$N_i(t) = \{j \mid |X_j - X_i| < r_d^i(t), \quad L_j > L_i\} \tag{5}$$

Probability of moving towards neighbor *j*:

$$p_{ij}(t) = \frac{L_j(t) - L_i(t)}{\sum_{k \in N_i(t)}(L_k(t) - L_i(t))} \tag{6}$$

Step 5. Movement&Radius Update

Binary position X_i is adjusted towards neighbor:

$$X_i(t+1) = X_i(t) + s * \frac{X_j(t) - X_i(t)}{\|X_j - X_i\|} \tag{7}$$

Neighborhood radius adapts:

$$r_d^i(t+1) = \min\left(r_s, \max\left(0, r_d^i(t) + \beta_r * (n_t - |N_i(t)|)\right)\right) \tag{8}$$

where r_s is maximum radius, β_r controls adaptation and n_t is target neighbors. The detailed procedure is outlined in TABLE 2.

TABLE 2. HEGSO Feature Selection Outline Algorithm

1.	Initialize *N* glowworms X_i and L_i.
2.	Evaluate fitness for each X_i.
3.	Until convergence or max iterations:
	- Update luciferin L_i.
	- Identify neighbors N_i.
	- Probabilistically select neighbor *j*.
	- Move X_i. towards X_j.
	- Update r_d^i.
	- Re-evaluate fitness.
4.	Output best subset *X**.

Anomaly Score estimation via ANN. In this stage ANN is employed to estimate the anomaly score for each data instance. The main objective is to learn mapping function f_0 that transforms the preprocessed and feature-selected input vector x into a scalar anomaly score $s \in [0,1]$, where values closer to 1 indicate higher likelihood of being anomalous.

The general forward propagation in the ANN can be expressed as:

$$h^{(l)} = \sigma\left(W^{(l)}h^{(l-1)} + b^{(l)}\right), \qquad l = 1, \dots, L \tag{9}$$

where:

- $h^{(0)} = x$ is the input vector after preprocessing and feature selection,
- $W^{(l)}$ and $b^{(l)}$ are weight matrices and bias vectors of layer *l,*
- $\sigma(.)$ is the activation function, ReLU for hidden layers, sigmoid for the output layer),
- L is the number of layers.

TABLE 3. Constuction of Isolation Trees Algorithm	
Input:	D – input datasets
	r – number of representations generated by GCERE
	t- number of isolation trees per representation
	J – maximum tree depth
	n – subsample size for each tree
Output:	T – forest of isolation trees
Procedure:	
1.	Initialize $T \leftarrow \emptyset$
2.	Generate representations $\{X_u\}_{u=1}^{r}$ via GCERE
3.	For u=1 to r:
	For i=1 to t:
	Select $P_1 \subseteq X_u, \lvert P_1\rvert = n$
	While P_k is a leaf node in tree τ_i:
	If $\lvert P_k\rvert > 1$ and depth $< J$:
	1. Randomly choose a dimension $j_k \in \{1, \dots, d\}$
	2. Randomly choose a split point $\eta_k \in \left[min_{x \in P_k} x^{(j_k)}\ max_{x \in P_k} x^{(j_k)}\right]$
	3. Partition:
	$P_{2k} = \left\{x \lvert x^{j(k)} \le \eta_k, \quad x \in P_k\right\}$
	$P_{2k+1} = \left\{x \lvert x^{j(k)} \le \eta_k, \quad x \in P_k\right\}$
	End While
	$T \leftarrow T \cup \{\tau_i\}$
4.	Return T

TABLE 4. DEAS Algorithm	
Input:	o- data object, T – isolation tree forest
Output:	$F_{DEAS}(o \lvert T)$ – final anomaly score
Procedure:	
1.	Generate representations $\{x_u\}_{u=1}^{r}$ via GCERE
2.	For u=1 to r:
	For each tree $\tau_i \in T$:
	Initialize: $k \leftarrow 1,\ \beta \leftarrow 0,\ p(x_u \lvert \tau_i) \leftarrow \emptyset$
	While $\lvert P_k\rvert > 1$ and depth $< J$:
	If $x_u^{j_k} \le \eta_k$ then $k \leftarrow 2k$ else $k \leftarrow 2k + 1$
	Update path:
	$p(x_u \lvert \tau_i) \leftarrow p(x_u \lvert \tau_i) \cup \{k\}$
	Accumulate deviation:
	$\beta \leftarrow \beta + \left\lvert x_u^{j_k} - \eta_k \right\rvert$
3.	Compute Final DEAS score:
	$F_{DEAS}(o \lvert T) = 2^{-\frac{E_{\tau_i \in T}[\lVert p(x_u \lvert \tau_i)\rVert]}{C(T)}} \times E_{\tau_i \in T}[g(x_u \lvert \tau_i)]$
	where $C(T)$ is the normalizing factor for path length in IF.

The final anomaly score is computed as:

$$s = \sigma\left(W^{(L)}h^{(L-1)} + b^{(L)}\right) \quad (10)$$

where $s \in [0,1]$ represents the probability of the instance being an anomaly.

The network parameters $\theta = \{W^{(l)}, b^{(l)}\}$ are optimized by minimizing the binary cross-entropy loss:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i logs_i + (1-y_i)\log(1-s_i)] \quad (11)$$

where $y_i \in \{0,1\}$ is the ground truth label, s_i is the predicted anomaly score for sample *i*.

By training the ANN in this way, the model learns discriminative features that highlight abnormal behaviors in the data. This anomaly score will be further refined in the Isolation Forest-based deep analysis stage.

IF-based deep analysis stage. The proposed IF-based deep analysis stage refines the preliminary anomaly scores generated by the ANN using a Isolation Trees (IT) structure combined with Deviation-Enhanced Anomaly Scoring (DEAS). This hybridization improves the detection of both global and local anomalies. The process divided into two main algorithms:**Final classification.** The final classification stage aims to combine anomaly scores obtained from both the ANN and IF-based deep analysis to make the ultimate decision on whether an observation is anomalous or normal. Given an observation o, the final score $S(o)$ is computed as a weighted combination of ANN-based anomaly probability and IF-based anomaly score:

$$S(o) = \alpha * S_{ANN}(o) + (1-\alpha) * S_{IF}(o) \quad (12)$$

where:

- $S_{ANN}(o)$ – Anomaly probability obtained from the ANN model.
- $S_{IF}(o)$ – Anomaly score computed from the deep isolation forest analysis.
- $\alpha \in [0,1]$ – Weight coefficient determining the influence of each method in the fusion process.

A threshold-based decision rule is applied to determine the final label:

$$\hat{y} = \begin{cases} 1, & if\ S(o) \geq \theta \quad (Anomaly) \\ 0, & if\ S(o) < \theta \quad (Normal) \end{cases} \quad (13)$$

Here θ is the decision threshold, typically selected using ROC curve analysis on a validation dataset to maximize classification performance.

For clarity and reproducibility, the step-by-step procedure of the final classification stage is outlined in TABLE 5, where ANN-based and IF-based scores are fused and thresholded to determine the final anomaly label.

TABLE 5. Final Classification Process

Input:	S_{ANN}, S_{IF}, weight α, threshold θ
Output:	Class label $\hat{y}$
1.	Score Acquisition: Obtain $S_{ANN}(o)$ and $S_{IF}(o)$ for each observation
2.	Score Fusion: Compute $S(o)$ using Equation (12)
3.	Thresholding: Compare $S(o)$ with θ
4.	Classification: Assign $\hat{y} = 1\ if\ S(o) \geq \theta$; otherwise $\hat{y}$=0

Evaluation. This stage is an essential step for determining the performance of ML systems. At this stage, the fulfillment of the proposed ANN2IF model was evaluated based on various metrics. The main metrics used for evaluation and their formulas are listed TABLE 6 [15]:

TABLE 6. Metrics and their formulas used to evaluate the model

Used metrics	Formula
Accuracy	$Acc = \frac{TP+TN}{TP+TN+FP+FN}$
Precision	$Prec = \frac{TP}{TP+FP}$
Recall	$Rec = \frac{TP}{TP+FN}$
F1-Score	$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$
False Alarm Rate (FAR)	$FAR = \frac{FP}{FP+TN} * 100\%$

These metrics are used to evaluate the overall presentation of the A2K model.

RESEARCH RESULTS

The experiments were conducted using the CICIDS2017 dataset, which contains a large variety of normal and malicious network traffic records. The dataset was preprocessed to extract relevant features and then split into 70% training and 30% testing sets. Each sample includes numerical and categorical attributes representing traffic flow characteristics, along with a corresponding class label indicating either benign traffic or a specific attack category.

The proposed ANN2IF hybrid model was implemented in Python, leveraging TensorFlow for constructing the artificial neural network and Scikit-learn for the Isolation Forest implementation, evaluation metrics, and ROC curve generation. The experiments were run on a workstation with 16GB RAM and an NVIDIA GPU, which significantly accelerated the training process.

Evaluation metrics were calculated for each attack category (Normal, DoS, Probe, R2L, U2R**)** as well as for the overall performance. In addition, False Alarm Rate (FAR) and ROC-AUC scores were used to measure the robustness of the model.

Table 7 presents the results obtained from the overall performance of the proposed ANN2IF model, according to which the model achieved results of 97.56% Acc etc., providing a result of FAR 1.50.

TABLE 7. Overall performance of ANN2IF on CICIDS2017 dataset

Evaluation Metric	Value (%)
Acc	97.56
Prec	97.20
Rec	98.69
F1	97.40
FAR	1.50

TABLE 8 shows the class-wise performance for each category. As expected, the model performs exceptionally well on the Normal and DoS categories, while detection rates for rare classes such as U2R and R2L are slightly lower but still outperform many conventional models.

TABLE 8. Class-wise performance of ANN2IF

Class	Prec (%)	Rec (%)	F1 (%)	FAR (%)
Normal	97.10	97.35	97.22	0.90
DoS	97.80	97.40	97.60	1.20
Probe	96.50	96.20	96.35	2.10
R2L	88.40	86.70	87.54	4.80
U2R	85.25	83.10	84.16	5.20

TABLE 9 shows the confusion matrix for the test set of the proposed model, where the diagonal values indicate correctly classified samples, and the remaining values indicate incorrectly classified samples.

TABLE 9. Confusion Matrix of ANN2IF (Testing Set)

	Predic: Normal	Predic: DoS	Predic: Probe	Predic: R2L	Predic: U2R
Actual: Normal	229	2	1	0	0
Actual: DoS	3	310	1	1	0
Actual: Probe	1	1	39	0	0
Actual: R2L	0	1	0	13	1
Actual: U2R	0	0	0	2	5

To validate the effectiveness of the ANN2IF model, we compared its performance with several widely used classifiers: SVM**,** ANN**,** KNN, and Random Forest**.**

TABLE 10. Performance comparison between baseline models and ANN2IF

Type	Acc(%)	Prec(%)	Rec (%)	F1 (%)	FAR (%)
SVM	94.25	93.10	92.70	92.90	7.30
ANN	95.80	94.20	93.50	93.84	6.20
KNN	91.25	89.30	87.45	88.37	8.75
Random Forest	96.20	95.00	94.30	94.65	5.80
Isolation Forest	93.00	91.00	90.00	90.50	7.00
ANN2IF	97.56	97.20	98.69	97.40	1.50

The ROC curves for both ANN2IF and baseline models were plotted to visualize classification performance. ANN2IF achieved the highest AUC score of 0.996, outperforming all baselines, indicating a stronger capability to distinguish between normal and malicious traffic.

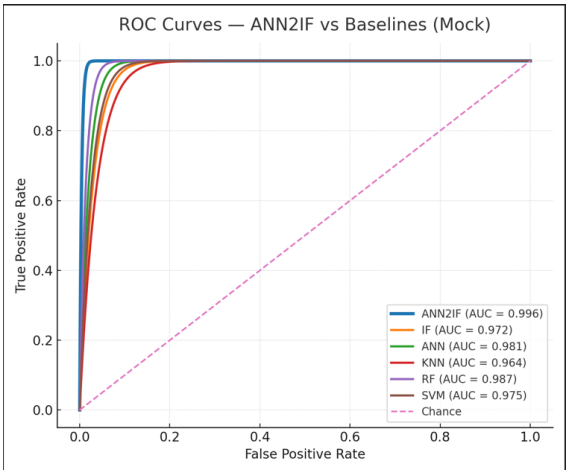

FIGURE 2. ROC Curve comparison of ANN2IF and baseline models

The integration of HEGSO-based feature selection significantly reduced dimensionality without losing critical anomaly indicators. The ANN stage effectively captured non-linear dependencies in the data, and the Deep Isolation Forest provided a robust anomaly separation even in high-dimensional space. Compared to baseline models such as standard Isolation Forest and ANN-only classifiers, the ANN2IF model demonstrated:

- Higher accuracy and recall.
- Lower false positive rate.
- Better generalization to unseen attack types.

CONCLUSIONS

This paper presents a hybrid ANN2IF model that combines Artificial Neural Network and Isolation Forest for anomaly detection and attack classification for network security. This approach combines the advantages of ANN's feature extraction and IF's anomaly detection, and demonstrates higher accuracy, efficiency, and robustness than simple and traditional methods.

Experimental results show that this model achieves high accuracy and low FAR when tested. This approach exhibits stronger ability to distinguish between normal and malicious traffic compared to other classifiers. From the above results, it can be seen that the proposed model can be effectively integrated into intrusion detection systems (IDS), providing high detection rates for various types of attacks while maintaining low false positives. Future research is planned to integrate and optimize this model in real-time, with a focus on reducing computational costs and verifying its performance in various network environments.

REFERENCES

1. C. C. Aggarwal, *Outlier Analysis*. Cham, Switzerland: Springer, 2017.
2. L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.
3. H. Wang, G. Pang, C. Shen, and C. Ma, "Unsupervised representation learning by predicting random distances," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2021, pp. 2950–2956.
4. G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2018, pp. 2041–2050.

5. K.-H. Lai, D. Zha, G. Wang, J. Xu, Y. Zhao, D. Kumar, Y. Chen, P. Zumkhawaka, M. Wan, D. Martinez, *et al.*, "TODS: An automated time series outlier detection system," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 18, 2021, pp. 16060–16062.
6. S. Alnegheimish, D. Liu, C. Sala, L. Berti-Equille, and K. Veeramachaneni, "Sintel: A machine learning framework to extract insights from signals," in *Proc. Int. Conf. Manag. Data (SIGMOD)*, 2022, pp. 1855–1865.
7. IBM, "Anomaly detection by IBM," [Online]. Available: https://developer.ibm.com/apis/catalog/ai4industry--anomaly-detection-product/introduction/. Accessed: Feb. 27, 2023.
8. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. IEEE ICDM*, 2008, pp. 413–422.
9. H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep Isolation Forest for anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12591–12604, Dec. 2023, doi: 10.1109/TKDE.2023.3270293
10. [10] P. Kumar, P. K. Singh, and S. R. Biradar, "ARLIF-IDS: Attention-based real-time lightweight isolation forest intrusion detection system," *arXiv preprint*, arXiv:2204.09737, 2022
11. A. A. Elsaid and S. Binbusayyis, "An optimized isolation forest-based intrusion detection system for heterogeneous and streaming data in the industrial Internet of Things (IIoT) networks," *SN Applied Sciences*, vol. 6, no. 2, Feb. 2024, doi: 10.1007/s42452-024-06165-w.
12. Y. Wang, J. Zhang, L. Liu, and W. Sun, "Hybrid intrusion detection system based on combination of random forest and autoencoder," *Symmetry*, vol. 15, no. 3, p. 568, Mar. 2023, doi: 10.3390/sym15030568.
13. T. Kumar, A. Patel, and J. Singh, "IoT network anomaly detection using isolation forest: An empirical analysis," *Applied Sciences*, vol. 14, no. 24, p. 11545, Dec. 2024, doi: 10.3390/app142411545.
14. F. N. Al-Wesabi, H. J. Alshahrani, A. E. Osman, E. S. Abd Elhameed, and E. Samir, "Financial Fraud Detection Using AEO-DMOA Based 1D-FRCNN Model with Effective Feature Selection Technique," Journal of Information Systems Engineering and Management, vol. 10, no. 40s, 2025. [Online]. Available: https://jisem-journal.com/index.php/journal/article/view/7297
15. Xia, Z., Wang, Y., Zhang, X., & Wang, Z. A novel hybrid model based on random forest and deep neural network for network intrusion detection. *IEEE Access*, 8, 2020, 68370–68381. doi:10.1109/ACCESS.2020.2986491