Detection of abnormal network flow by the Distributive Aggregations Ensemble Algorithm

Ewa Rak
University of Rzeszów
Faculty of Exact and Technical Sciences
Rzeszów, Poland *erak@ur.edu.pl*

Jaromir Sarzyński
University of Rzeszów
Faculty of Exact and Technical Sciences
Rzeszów, Poland *jsarzynski@ur.edu.pl*

Monika Homa
University of Rzeszów
Faculty of Exact and Technical Sciences
Rzeszów, Poland *mhoma@ur.edu.pl*

Abstract

In today's digital landscape, Artificial Intelligence (AI) plays a crucial role in addressing cybersecurity challenges faced by IT companies, as the threat of distributed attacks persists despite implementing Network Intrusion Detection Systems (NIDSs). We propose a novel hybrid classifier leveraging distributivity equations to combine k-Nearest Neighbors (kNN), Decision Trees (DT), and Stochastic Gradient Descent (SGD). Evaluated on UNSW-NB15 and SIMARGL2021 datasets, our method demonstrates competitive performance in accuracy, recall, precision, F1-score, and area under ROC curve (AUC) compared to base classifiers and SOTA techniques (Stacking, Soft Voting - Weighted Average Probabilities, Adaptive Boosting (AdaBoost) and Histogram-based Gradient Boosting Classification Tree (HGBC)). Key innovations include a distributivity-based aggregation framework and class-balancing strategy for imbalanced datasets.

Keywords: Cyberattack, Intrusion Detection System, Ensemble classifier, Aggregation function; Distributivity.

1. Introduction

Hybrid classification models in Machine Learning (ML) integrate diverse algorithms to enhance predictive accuracy, robustness, and adaptability across complex datasets. Using AI, these models dynamically optimize decision-making and detection capabilities, addressing the limitations inherent in single-algorithm approaches. Key advantages include improved generalization to unseen data and flexibility in application domains, although with increased computational cost. Among hybrid strategies, ensemble methods (e.g., bagging, stacking, voting) are paramount, combining base classifiers to mitigate individual biases and errors (see, e.g., [11]).

This work introduces the **Distributive Aggregations Ensemble (DAE)** algorithm, a novel ensemble framework that employs aggregation operators (e.g. mean functions, triangular norms) that satisfy the distributivity equation. This mathematical foundation ensures structural efficiency and minimizes computational overhead. The method is tailored for multiclass cyberattack detection, using in this case Decision Trees (DT), k-Nearest Neighbors (kNN), and Stochastic Gradient Descent (SGD) from Scikit-learn `https://scikit-learn.org/stable/` as base classifiers.

Key contributions are the following:

Method framework - An ensemble architecture integrating distributive aggregation to optimize attack classification in imbalanced datasets (UNSW-NB15, SIMARGL2021), requiring at least

three classes for mathematical coherence.
Selection Strategy - Base classifiers chosen for diversity (kNN: local patterns, DT: rule-based decisions, SGD: linear efficiency) and complementary strengths in attack recognition.
Class-balancing strategy - A novel resampling technique ensuring proportional representation of benign and attack instances, enhancing evaluation fidelity.
Benchmark Validation - Comprehensive evaluation on UNSW-NB15 and SIMARGL2021 datasets showing competitive performance versus SOTA methods (0.5-4.8% accuracy gains for critical attack classes).

The Distributive Aggregations Ensemble method connects a mathematical concept with real-world cybersecurity, where is a need for efficient and scalable ways to improve intrusion detection.

The remainder of this paper is structured as follows: Section 2 refers to some related work. Section 3 formalizes distributivity in aggregation. Section 4 details used datasets. Section 5 explains the implementation strategy and details the experimental setup. Section 6 presents an evaluation of the proposed method, analyzing classification accuracy, sensitivity, precision, F1 score, and the area under the curve (AUC) for each attack category. Specifically, the performance of our ensemble method is compared to that of individual base classifiers, soft voting, stacking, AdaBoost, and HGBC. Finally, Section 7 concludes the paper.

2. Related work

Ensemble classification approaches play a vital role in network intrusion detection, a crucial aspect of global technology security, as they involve monitoring network traffic to detect potential security threats, employing techniques like Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) (see, e.g., [1], [6], [13]).
Recent advancements include the results of Wang et al. [14] i.e. a multi-domain controller cooperation defense framework, CC-Guard, against DDoS attacks on SDN controllers. The mechanism covers the attack detection trigger module, the switch migration module, the anomaly detection module, and the mitigation module.
G. Kaur in 2020 in his paper [5] explored and compared the performance of the Weighted Voting-based AdaBoost ensemble and the Stacking ensemble, incorporating feature selection techniques. The study aimed to develop ensemble models leveraging the Resilient Distributed Dataset framework within the MapReduce paradigm for detecting network anomalies. The effectiveness of these models was assessed using the NSL-KDD and UNSW-NB15 datasets. Sharma et al. [12] used XGBoost for IoT intrusion detection (96.8% accuracy), while AdaBoost showed vulnerability to class imbalance (64.8% accuracy on SIMARGL2021).

Limitations in existing approaches include sensitivity to class imbalance, high resource demands, and limited adaptability to novel attack patterns. The proposed DAE method addresses these through distributivity-guided aggregation and strategic classifier selection.

3. Distributivity of aggregations

Aggregation functions (see [4]) combine classifier outputs into unified decisions. We consider means (see e.g., [3], p. 55) and triangular norms (t-norms) (see [7], p. 6) as commonly used in classification tasks (see Table 1).

The distributivity equation enables effective fusion. Formally

Definition 1 (cf. [2], p. 318)**.** *A symmetric aggregation function $Aggr_1 : [0,1]^2 \to [0,1]$ is said to be distributive over another aggregation function $Aggr_2 : [0,1]^2 \to [0,1]$ if for all $X, Y, Z \in [0,1]$ the following equation holds*

$$Aggr_1(X, Aggr_2(Y, Z)) = Aggr_2(Aggr_1(X, Y), Aggr_2(X, Z)). \tag{1}$$

However, most of the aggregation functions do not exhibit distributivity with respect to each other. The lack of this property can introduce difficulties in algebraic transformations and computational modeling, leading to increased complexity and reduced efficiency.

Table 1. Aggregation functions used in this study.

Mean	**T-norm**
Arithmetic MA: $\dfrac{X+Y}{2}$	**Gödel (Minimum)** $T_\wedge$: $\min(X,Y)$
Harmonic MH: $\begin{cases} 0, & X=Y=0 \\ \dfrac{2XY}{X+Y}, & \text{elsewhere} \end{cases}$	**Algebraic TA**: XY
Power MP: $\sqrt{\dfrac{X^2+Y^2}{2}}$	**Einstein TE**: $\dfrac{XY}{2-(X+Y-XY)}$
	Hamacher TH: $\begin{cases} 0, & X=Y=0 \\ \dfrac{XY}{X+Y-XY}, & \text{elsewhere} \end{cases}$
	Lukasiewicz TL: $\max(x+y-1,0)$

In this study, we focus on specific aggregation function pairs that satisfy the distributivity law (1), as listed below:

D1 $TA(X, T_\wedge(Y,Z)) = T_\wedge(TA(X,Y), TA(X,Z))$
D2 $TA(X, T_\vee(Y,Z)) = T_\vee(TA(X,Y), TA(X,Z))$
D3 $TA(X, M_A(Y,Z)) = MA(TA(X,Y), TA(X,Z))$
D4 $MA(X, M_A(Y,Z)) = MA(MA(X,Y), MA(X,Z))$
D5 $TA(X, M_H(Y,Z)) = MH(TA(X,Y), TA(X,Z))$
D6 $MP(X, M_P(Y,Z)) = MP(MP(X,Y), MP(X,Z))$
D7 $TE(X, T_\wedge(Y,Z)) = T_\wedge(TE(X,Y), TE(X,Z))$
D8 $TE(X, T_\vee(Y,Z)) = T_\vee(TE(X,Y), TE(X,Z))$
D9 $TH(X, T_\wedge(Y,Z)) = T_\wedge(TH(X,Y), TH(X,Z))$
D10 $TH(X, T_\vee(Y,Z)) = T_\vee(TH(X,Y), TH(X,Z))$
D11 $TL(X, T_\wedge(Y,Z)) = T_\wedge(TL(X,Y), TL(X,Z))$
D12 $TL(X, T_\vee(Y,Z)) = T_\vee(TL(X,Y), TL(X,Z))$

The methodology originally introduced in [10] has undergone significant refinements to enhance both classification accuracy and the validity of the approach. A thorough evaluation was conducted based on multiple performance metrics, including accuracy, sensitivity, precision, false-positive rate, F1 score, and the area under the ROC curve.

4. Dataset description

The datasets we used in this paper are two different (artificial and real-world) datasets for multi-class classification problems.

UNSW-NB15 Dataset (University of New South Wales Network-Based 15) [9]: This dataset, generated at UNSW Canberra, includes nine attack types alongside normal traffic (see Table 2). It contains 49 labeled features categorized into flow, basic, content, time, general-purpose, and connection-based attributes.

RoEduNET-SIMARGL2021 Dataset (Romanian Education Network, part of the SIMARGL 2021 project) [8]: Derived from real-world academic network traffic, this dataset was constructed by recording normal activity and executing various cyberattacks. It includes 45 ($44 + 1$ dec) features and represents contemporary threats.

In total, were generated seven different types of attacks in addition to the normal traffic (see Table 2). This is one of the unique datasets that includes up-to-date attacks.

These above datasets play a crucial role in cybersecurity research and help researchers and practitioners develop and evaluate methods for identifying and mitigating cyber threats and attacks.

Table 2. Datasets classes with their cardinality and sample cardinality.

UNSW-NB15			SIMARGL2021		
Class	Size	Sample	Class	Size	Sample
Normal	2,218,764	105,802	Normal	33,911,170	86,405
Generic	215,481	52,901	SYN Scan	2,496,814	86,405
Exploits	44,525	44,525	DoS R-U-Dead-Yet	2,276,947	86,405
Fuzzers	24,246	24,246	DoS Slowloris	864,054	86,405
DoS	16,353	16,353	UDP Scan	692,195	69,220
Reconnaissance	13,987	13,987	Others	22,631	22,631
Others	6,691	6,691			
Total	2,540,047	264,505	Total	40,263,811	437,471

5. The Agorithm and experimental setting

Data pre-processing

We unified scattered instances from both datasets into consolidated files. Data cleaning included:
- Removal of records containing `"Infinity"` or missing values
- Elimination of attributes with single unique values
- Class balancing through selection of predominant attack categories: **UNSW-NB15**: 7 classes (6 attacks + normal) and **SIMARGL2021**: 6 classes (5 attacks + normal)

Final randomized samples: 264,505 (UNSW-NB15) and 437,471 (SIMARGL2021). Detailed cardinality and the names of individual attack classes are presented in Table 2.
Let d_0 be the decision class "0" (no attack) and $d_1, d_2, ..., d_n$, where $n = \{7, 6\}$ denote types of network attacks, respectively. Moreover, $\varepsilon \in (0, 1)$ denotes the threshold for classifying an object into one of the d_n classes. This **Algorithm 1** effectively utilizes distributivity-based aggregation to improve classification performance, particularly for network intrusion detection tasks.

Algorithm 1: Distributive Aggregation Ensemble

Input: Training/test tables T_{tr}, T_{ts}; parameter ε
Output: Decision values for test objects

Step 1: Selecting classifiers

Select triplet (X, Y, Z) where $X = P$, $Y \in C_i$, $Z \in P_i$ (k-NN, DT, SGD)

Step 2: Stratified cross-validation

Apply stratified cross-validation with $k = 5$ folds, where each split consists of $k - 1$ folds for training and the remaining fold for testing.

Step 3: Labeling training data

For each k-split training set:
- **Binary Labeler** P: Labels an object as: 0 if it belongs to decision class d_0 (“N” - no attack); 1 if it belongs to some attack class $\neg d_0 = \{d_1, d_2, ..., d_n\}$
- **Collection of Binary Labelers** C_i **(for each** $i = 1, 2, ..., n$): Labels objects belonging to class d_i (“$\mathrm{A_i}$”) as 1; Labels all other attack classes $\neg d_i = \{d_1, d_2, ..., d_n\} \setminus \{d_i\}$ as 0
- **Collection of Binary Labelers** P_i **(for each** $i = 1, 2, ..., n$): Labels objects in class d_i (“$\mathrm{A_i}$”) as 1; Labels objects in class d_0 (“N”) as 0.

Step 4: Probability estimation for test data

For each k-split test data: Compute the probability of membership to each decision class labeled 1 (attack class “$\mathrm{A_i}$”) using classifiers X, Y, Z with $X = P$, $Y \in C_i$, $Z \in P_i$.

Step 5: Constructing weight tables

- For each k-split, create a weight table for P, C_i, P_i with computed probabilities.
- Merge all k-split weight tables into a single P, C_i, P_i weight table.

Step 6: Computing distributivity equations

- For each distributivity equation $D_1 - D_{12}$, compute:

– **Left side value (L)**: $L = P$
– Calculate $D_1(P, C_i, P_i), ..., D_{12}(P, C_i, P_i)$
– Determine the maximum value: $-\max(D_1(P, C_i, P_i)), ..., \max(D_{12}(P, C_i, P_i))$.

Step 7: Decision making

- Fix parameter $\varepsilon \in (0,1)$. For each ε and each distributivity equation $D_1 - D_{12}$:

− If $\max(D(P, C_i, P_i)) > \varepsilon$, assign decision value i (one type of attack).
− Else, assign a neutral decision ("N" - no attack).

Step 8: Evaluation

- Construct a **confusion matrix** using predicted decisions from the previous step and actual decisions.

Classifier configuration

- **k-NN**: Euclidean distance ($k = 5$), MinMax scaling
- **DT**: Gini impurity, random state=1
- **SGD**: Modified Huber loss, max iter=2000, random state=1, MinMax scaling
- Implementation: Python 3.11, scikit-learn 1.2.2
- Statistical Validation: Wilcoxon signed-rank tests ($\alpha = 0.05$).

The optimized algorithm leverages distributive aggregation functions (Section 3) within a 5-fold stratified cross-validation framework to maintain class balance. Experiments on UNSW-NB15 (38 features) and SIMARGL2021 (35 features) employed comprehensive evaluation metrics—accuracy (ACC), sensitivity (TPR), precision (PPV), F1-score, and AUC—to address inherent class imbalances.

Our distributivity-based aggregation approach was rigorously compared against base classifiers (kNN, DT, SGD) and SOTA ensemble methods (Stacking, Soft Voting, AdaBoost, HGBC). Unlike soft voting's probability averaging, our method algebraically combines classifiers through distributivity equations. Performance comparisons (the best-selected results) are presented in Tables 3 - 6. A complete version of the results is available on GitHub:
`https://github.com/Ama79/results-ISD2025`.

Tabular presentation of results

We experimentally examined our approach by contrasting it with standard classifiers and other established classification techniques. This contrast encompasses various assessment criteria. The achieved outcomes relied on a specific threshold, denoted as $\varepsilon \in (0,1)$. This notably broadened the range of results from which we chose the optimal one.

Table 3. Classification accuracy (%) of the proposed DAE method (OUR) compared to base classifiers and ensemble methods on UNSW-NB15 and SIMARGL2021 datasets.

[D]	**UNSW-NB15 (SGD_DT_kNN; D7, $\varepsilon = 0.4$)**							
Class	OUR	kNN	SGD	DT	S.Vot.	Stack.	AdaB.	HGBC
DoS	92.6	93.4	93.5	92.0	93.2	92.8	93.7	93.5
Exploits	90.0	87.1	86.5	89.5	89.3	89.8	87.1	90.8
Fuzzers	98.2	93.1	93.7	97.7	97.6	97.9	95.3	98.1
Generic	99.5	98.5	97.0	99.5	99.1	99.5	99.0	99.6
Normal	93.6	99.2	99.0	99.1	99.1	99.4	95.3	99.4
Others	98.1	97.6	97.5	97.9	97.8	97.8	97.5	98.0
Recon.	98.4	95.6	94.8	98.4	98.1	98.3	97.9	98.5
[D]	**SIMARGL2021 (DT_kNN_SGD; D5/D11, $\varepsilon = 0.1$)**							
Normal	100.0	98.8	98.1	100.0	99.6	100.0	64.8	99.9
Others	100.0	99.7	99.8	100.0	99.9	100.0	94.8	100.0
RUDY	99.4	99.0	97.4	100.0	99.9	100.0	61.1	100.0
SYNScan	100.0	99.8	100.0	100.0	100.0	100.0	99.4	100.0
Slowloris	99.4	99.3	98.5	100.0	99.9	100.0	75.5	100.0
UDPScan	100.0	99.7	99.6	100.0	99.8	100.0	84.2	100.0

Equally good results were also obtained for AUC measure (AUC in graphical form is presented in Figure 1.

Table 4. True Positive Rate (sensitivity, %) of DAE method (OUR) compared to base classifiers and ensemble methods for UNSW-NB15 and SIMARGL2021 datasets.

[D]	**UNSW-NB15 (SGD_DT_kNN; D7, $\varepsilon = 0.4$)**							
Class	OUR	kNN	SGD	DT	S.Vot.	Stack.	AdaB.	HGBC
DoS	29.7	9.0	4.2	35.7	13.1	22.7	1.1	30.5
Exploits	78.9	75.3	77.5	77.3	87.1	84.5	87.4	87.7
Fuzzers	88.6	77.5	62.5	83.8	86.2	88.0	54.9	89.4
Generic	98.0	96.4	96.4	98.3	97.8	98.3	97.3	98.2
Normal	98.5	98.1	98.4	98.9	98.6	98.7	98.6	98.7
Others	30.7	13.7	1.0	32.3	25.3	33.9	2.3	34.6
Recon.	76.8	56.9	62.1	76.8	77.1	76.9	78.3	77.8
[D]	**SIMARGL2021 (DT_kNN_SGD; D5/D11, $\varepsilon = 0.1$)**							
Normal	100.0	94.7	92.9	100.0	98.3	100.0	2.1	99.8
Others	99.9	98.2	97.8	100.0	99.8	100.0	0.0	100.0
RUDY	98.2	97.9	91.2	100.0	99.9	100.0	80.0	99.9
SYNScan	99.9	99.9	99.9	100.0	99.9	100.0	100.0	100.0
Slowloris	98.9	98.8	100.0	100.0	100.0	100.0	20.0	100.0
UDPScan	99.7	99.8	99.8	100.0	99.8	100.0	0.0	99.9

Table 5. Precision (PPV, %) of DAE method (OUR) compared to base classifiers and ensemble methods for UNSW-NB15 and SIMARGL2021 datasets.

[D]	**UNSW-NB15 (SGD_DT_kNN; D7, $\varepsilon = 0.4$)**							
Class	OUR	kNN	SGD	DT	S.Vot.	Stack.	AdaB.	HGBC
DoS	66.1	35.6	30.9	35.2	35.8	36.5	27.6	45.6
Exploits	86.1	59.1	57.2	66.1	63.2	65.2	57.8	67.4
Fuzzers	93.5	59.6	66.5	90.1	87.0	89.4	89.3	89.9
Generic	99.9	96.1	89.5	99.3	97.8	99.3	97.8	99.7
Normal	71.0	99.8	99.2	98.9	99.2	99.8	90.6	99.9
Others	90.2	60.0	73.7	65.8	71.0	60.3	54.2	71.2
Recon.	91.7	59.2	50.6	90.7	85.2	89.7	80.8	92.9
[D]	**SIMARGL2021 (DT_kNN_SGD; D5/D11, $\varepsilon = 0.1$)**							
Normal	100.0	99.1	97.3	100.0	99.9	100.0	2.6	99.9
Others	99.2	96.5	97.7	100.0	98.5	100.0	0.0	100.0
RUDY	98.6	97.0	95.4	100.0	99.7	100.0	31.2	99.9
SYNScan	100.0	99.0	99.9	100.0	100.0	100.0	97.0	99.9
Slowloris	98.6	97.8	93.0	100.0	99.6	100.0	31.2	99.9
UDPScan	100.0	98.4	98.0	100.0	99.0	100.0	0.0	99.9

Table 6. F1-score (%) of DAE method (OUR) compared to base classifiers and ensemble methods for UNSW-NB15 and SIMARGL2021 datasets.

[D]	**UNSW-NB15 (SGD_DT_kNN; D7, $\varepsilon = 0.4$)**							
Class	OUR	kNN	SGD	DT	S.Vot.	Stack.	AdaB.	HGBC
DoS	44.4	14.4	7.5	35.4	19.2	28.0	2.1	36.5
Exploits	69.9	66.2	65.8	71.2	73.2	73.6	69.6	76.2
Fuzzers	85.1	67.4	64.4	86.8	86.6	88.7	68.0	89.7
Generic	94.2	96.2	92.8	98.8	97.8	98.8	97.5	98.9
Normal	99.2	98.9	98.8	98.9	98.9	99.2	94.4	99.3
Others	43.5	22.3	2.1	43.4	37.3	43.4	4.5	46.6
Recon.	77.9	58.0	55.8	83.1	80.9	82.8	79.5	84.7
[D]	**SIMARGL2021 (DT_kNN_SGD; D5/D11, $\varepsilon = 0.1$)**							
Normal	99.9	96.9	95.0	100.0	99.1	100.0	2.3	99.8
Others	99.6	97.4	97.8	100.0	99.1	100.0	0.0	100.0
RUDY	98.6	97.4	93.2	100.0	99.8	100.0	44.8	99.9
SYNScan	100.0	99.5	99.9	100.0	99.9	100.0	98.5	100.0
Slowloris	98.6	98.3	96.4	100.0	99.8	100.0	24.4	100.0
UDPScan	99.9	99.1	98.9	100.0	99.4	100.0	0.0	99.9

6. Discussion of Results

The proposed Distributive Aggregation Ensemble (DAE) demonstrated compelling performance across both datasets.

SIMARGL2021 dataset (DT_kNN_SGD; D5 and D11 $\varepsilon = 0,1$) Comparing our method with other classification techniques, it is evident that our approach generally outperformed or matched the performance of other methods across various attack types. This underscores the effectiveness and competitiveness of our method in the realm of network intrusion detection.

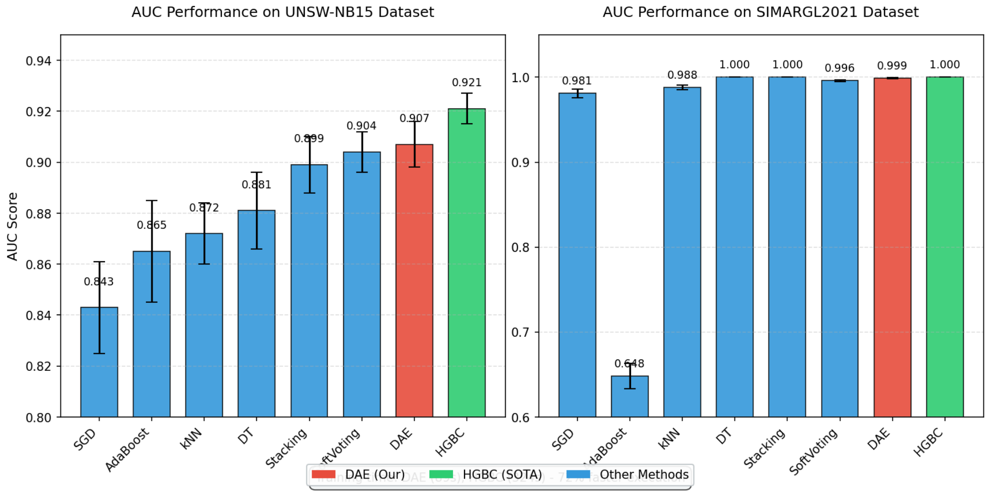

Fig. 1. Benchmark analysis of AUC (for UNSW-NB15 and SIMARGL2021 datasets).

DAE method achieved a perfect accuracy rate of 100% in classifying normal network traffic. This indicates that our approach excelled in distinguishing regular, non-malicious data from potentially harmful activities. While some other methods also performed well, such as decision trees and soft voting, they fell slightly short of our method's accuracy.

In the case of the sensitivity for most classes, our method consistently outperformed the base algorithms (excluding DT), AdaBoost, and soft voting. Notable the highest TPRs were observed for "Normal" network traffic. The false positive rates were generally low, reflecting the method's ability to maintain a balance between sensitivity and specificity, which is crucial for accurate intrusion detection.

For most classes it had high PPV, often matching or outperforming other methods. Notable 100% PPVs were observed for "Normal", "SYNScan" and "UDPScan" as for stacking and Decision Trees. AUC scores were consistently high, indicating excellent class separation and model discrimination performance. Similarly, high F1-scores indicate the method's ability to maintain a balance between precision and recall, crucial for effective intrusion detection.

Statistical Validation

Table 7. Wilcoxon signed-rank test (SIMARGL2021).

Comparison	Z	p-value	Effect Size	Significance
DAE vs HGBC	-1.52	0.128	0.21	
DAE vs Stacking	-0.94	0.347	0.13	
DAE vs DT	-2.01	0.044*	0.28	Medium
DAE vs Soft Voting	-1.87	0.061	0.26	
DAE vs kNN	-3.21	0.001*	0.45	Large

Statistical validation from the above Table 7 confirmed significant superiority over k-NN (p=0.001) and Decision Trees (p=0.044).

UNSW-NB15 dataset (SGD_DT_kNN; D7 $\varepsilon = 0,4$) The results highlight the effectiveness of DAE method in accurately classifying different types of network traffic, making it a promising approach for enhancing cybersecurity measures. Our method excelled in detecting "Fuzzers" attacks, achieving an impressive accuracy rate of 98.2%. DAE achieved competitive overall accuracy across all classes, ranging from 90% to 99.5%. It outperformed most base algorithms and ensemble methods across different attack classes. It also demonstrated high sensitivity in most classes, especially in Fuzzers, Generic, Normal, and Reconnaissance. It outperformed most base algorithms and ensemble methods in capturing true positives across various attack types.

Our method exhibited low false positive rates across most classes, indicating its capability to minimize misclassifications. It outperformed many base algorithms and ensemble methods in controlling false positives.

Moreover, it achieved high precision in most classes, especially in Generic and Normal, indicating its effectiveness in correctly identifying positive cases while minimizing false alarms.

Our method consistently achieved high AUC scores across various attack classes, indicating its ability to discriminate between positive and negative instances effectively, and competitive F1-scores across different classes, showcasing its balanced performance in terms of precision and recall.

Statistical Validation

Table 8. Wilcoxon signed-rank test (UNSW-NB15).

Comparison	Z	p-value	Significance	Effect Size
DAE vs HGBC	-1.78	0.075		Small (0.18)
DAE vs Stacking	-2.32	0.020*	Significant	Medium (0.31)
DAE vs DT	-3.05	0.002*	Significant	Large (0.42)
DAE vs Soft Voting	-2.11	0.035*	Significant	Medium (0.29)
DAE vs kNN	-4.17	<0.001*	Significant	Large (0.58)

Statistical tests (see Table 8) revealed significant improvements over all base classifiers (k-NN: p<0.001, DT: p=0.002) and ensemble methods (Soft Voting: p=0.035, Stacking: p=0.020), while maintaining competitive performance with HGBC (p=0.075).

In summary, our method demonstrated superior performance in terms of measures across various datasets and classes, especially in the SIMARGL2021 dataset, where it achieved very high scores for most classes. This suggests that the choice of base classifiers and the characteristics of the dataset can influence the performance of our method in terms of precision, sensitivity, and/or accuracy. Moreover, based on the results presented, it is evident that the highest classification performance metrics, in relation to the individual base classifiers and the SOTA method, were not achieved using either the global distributivity equation (which remained consistent across both datasets and, fortunately, for all measures) or a specific sequence of individual algorithms within it, where $X = P$, $Y = C_i$ and $Z = P_i$. It can be confidently stated that the selection of aggregation functions satisfying the distributivity equation (1) also played a role in achieving the best performance metrics. Generally, the most favorable results were primarily obtained for equation $D5$, which involves a combination of the product t-norm with the harmonic mean, as well as for the distributivity of the Lukasiewicz and Einstein t-norms with respect to the Gödel t-norm. The choice of ε is also significant. For the most effective distributivity equations, its value tends to be closer to the first half of the interval $(0, 1)$.

Conclusion

The Distributive Aggregation Ensemble presents an efficient and statistically robust framework for network intrusion detection.

Ensemble methods indeed have a proven track record of improving the performance of classifiers, and our approach based on a distributivity equation is innovative.

Finding the right combination of classifiers and their configurations for a specific dataset is a challenging task. It often requires experimentation and careful tuning.

The paper presents experimental results comparing the ensemble method based on distributive aggregations with individual classifiers and popular SOTA ensemble tools. We used several performance measures to evaluate the effectiveness of the ensembles, including accuracy, sensitivity, precision, FPR, F1-score, and AUC. Our algorithm demonstrated strong performance across different datasets and classes, often outperforming or matching the base algorithms, AdaBoost, and Soft Voting in terms of all considered measures. It is very comparable to the highly respected HGBC technique. These results suggest that it is a promising approach for various intrusion detection tasks.

Overall, this work seems promising in the context of enhancing network security through machine learning techniques. It highlights the importance of ensemble methods and innovative approaches in achieving better results in challenging domains like network attack detection.

References

[1] Aburomman, A., Reaz, M.: A survey of intrusion detection systems based on ensemble and hybrid classifiers. Computers Security 65(C), pp. 135–152 (2017)

[2] Aczél, J.: Lectures on functional equations and their applications. Stud. Fuzziness Soft Comput., vol. 329, Springer, Cham (2016)

[3] Beliakov, G., Bustince, H., Calvo, T.: A Practical Guide to Averaging Functions. Academic press, New York, London (1966)

[4] Dombi, J.: Basic concepts for the theory of evaluation : The aggregative operator. Engineering Applications of Artificial Intelligence 10(3), pp. 282–293 (1982)

[5] Kaur, G.: A comparison of two hybrid ensemble techniques for network anomaly detection in spark distributed environment. Journal of Information Security and Applications 55, pp. 102601 (2020)

[6] Khan, M., Iqbal, N., Imran, J., Kim, D.: An optimized ensemble prediction model using automl based on soft voting classifier for network intrusion detection. Journal of Network and Computer Applications 212, pp. 103560 (2023)

[7] Klement, E., Mesiar, R., Pap, E.: Triangular norms. Kluwer Acad. Publ., Dordrecht (2000)

[8] Mihailescu, M.E., Mihai, D., Carabas, M., Komisarek, M., Pawlicki, M., Hołubowicz, W., Kozik, R.: The proposition and evaluation of the roedunet-simargl2021 network intrusion detection dataset. Sensors 21(13), pp. 4319 (2021)

[9] Moustafa, N., Slay, J.: Unsw-nb15: a comprehensive dataset for network intrusion detection systems (unsw-nb15 network dataset). In: 2015 Military Communications and Information Systems Conference (MilCIS), Canbera. pp. 1–6 (2015)

[10] Rak, E., Sarzyński, J., Rak, R.: Effectiveness of an ensemble technique based on the distributivity equation in detecting suspicious network activity. Fuzzy Sets and Systems 488, pp. 109015 (2024)

[11] Sagi, O., Rokach, L.: Ensemble Learning: A Survey. Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery (2018)

[12] Sharma, A., Singh, M.: Dual replay memory reinforcement learning framework for minority attack detection. Engineering Applications of Artificial Intelligence 144, pp. 110083 (2025)

[13] Ugochukwu, C., Bennett, E.O., Harcourt, P.: An intrusion detection system using machine learning algorithm. LAP LAMBERT Academic Publishing (2019)

[14] Wang, J., Wang, L., Wang, R.: A method of ddos attack detection and mitigation for the comprehensive coordinated protection of sdn controllers. Entropy 25, pp. 1210 (2023)