International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 16s,2025
https://theaspd.com/index.php

Adaptive Deep Clustering And Outlier Aware Framework For Reducing False Alerts In Web-Based Intrusion Detection Systems

Parepalli Nageswara Rao[1], K. Radhika[2]
[1]Research Scholar, Osmania University, Assistant Professor, Neil Gogte Institute of Technology, OU
nagcsengit@gmail.com
[2]Professor, Chaitanya Bharati Institute of Technology, IT Department, Hyderabad, India

Abstract: *It is no marvel why Intrusion Detection Systems (IDS) are highly sought after in this age of online services that define the capabilities of enterprises. The current models usually break down in dynamic setting with elevated rates of false positive and false negative effectively, especially when the attack patterns are irregular or zero-day attacks. The paper has proposed an adaptive hybrid IDS framework that uses unsupervised deep learning and statistical based outlier detection in response to these limitations. Particularly, it combines three new modules such as Deep Contextual Clustering Algorithm (DCCA), Central Tendency Outlier Detection Algorithm (CTODA) and a Rule-Based Semantic Expansion Engine (RSEE). They combine to give a layered detection technique that compromises between behavioral learning and statistical accuracy and the interpretability of rules. The model was compared against NSL-KDD and KDD Cup 99 dataset and it was noted that the model reduced false alerts by a wide margin and the overall accuracy of the detection was 94%. Unlike stiff signature based systems, our framework dynamically adjusts to emerge threats, therefore it is much applicable in present-day cloud and web systems. The outcomes of the experiment confirm the generalizability of the model across the types of attacks, reduce fatigue among the analyzing experts, and provide the extended threat intelligence in real time.*
Keywords: *Intrusion Detection, Deep Clustering, False Positive Reduction, Outlier Detection, Web Security, Unsupervised Learning, Rule-Based Inference*

1. INTRODUCTION

In the modern era of digital transformation, the exponential growth in web-based services has led to unprecedented convenience and scalability in enterprise operations. However, this evolution is not without its consequences. The same openness, decentralization, and flexibility that make web environments attractive to organizations also render them fertile ground for malicious intrusions and cyberattacks. As data becomes more distributed and dynamic across virtual infrastructures, the challenge of distinguishing benign user behavior from harmful anomalies becomes increasingly complex.

Traditional cybersecurity mechanisms, such as firewalls and signature-based intrusion detection systems (IDS), are no longer sufficient in isolation. These systems often rely on predefined patterns or static rulesets, rendering them ineffective against emerging threats, zero-day vulnerabilities, or obfuscated attack vectors. More critically, their overreliance on static training data or deterministic logic has led to persistently high false positive and false negative rates. This inefficiency not only overloads security analysts with irrelevant alerts but also leaves systems vulnerable to silent breaches.

From a technical standpoint, intrusion detection in web environments is fundamentally a highdimensional, real-time anomaly recognition problem. It must account for complex network behavior, temporal dependencies, user profiles, and heterogeneous protocols. Traditional machine learning models, particularly supervised classifiers, are often hindered by their dependence on large volumes of accurately labeled data-an unrealistic requirement in rapidly evolving threat landscapes.

Moreover, many existing solutions fail to generalize across different network topologies or adapt to the nuanced patterns of modern web traffic.

Subjectively, the human and organizational cost of intrusion detection failure cannot be overstated. False positives lead to alert fatigue, desensitizing analysts and increasing the likelihood of genuine threats being ignored. False negatives, conversely, represent missed attacks-often only discovered after data loss, service disruption, or reputational damage has occurred. As enterprises become more reliant on interconnected digital ecosystems, there is a growing urgency for IDS solutions that are not only accurate and efficient but also adaptive and interpretable.

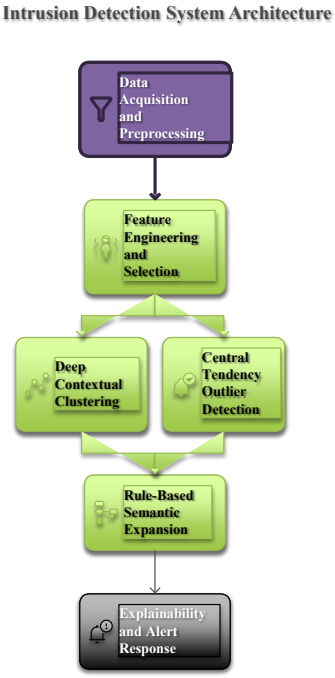

Figure 1: Intrusion Detection System Architecture::

This paper responds to that call by proposing a deep clustering-based hybrid framework aimed at reducing both false positive and false negative alerts in IDS, especially within complex and distributed web environments. By leveraging unsupervised learning principles, the system learns to detect irregular and unpredictable intrusion patterns without explicit labels. Unlike conventional models, it adapts dynamically to changes in traffic patterns and threat behaviors. Our dual-algorithm approachcomprising a Deep Contextual Clustering Algorithm (DCCA) and a Central Tendency Outlier Detection Algorithm (CTODA)—enables nuanced classification and robust anomaly identification through statistical and contextual understanding. The novelty of this approach lies in its holistic design: the combination of entropy-adjusted deep clustering for grouping behavioral similarities, and central tendency analytics for outlier detection, addresses the dual challenge of under-detection (false negatives) and over-detection (false positives). Moreover, the unsupervised nature of the system ensures it remains effective even in the absence of fresh labels or frequent model retraining. This work aligns itself with a broader vision: transforming intrusion detection from a static, reactive measure into a dynamic, predictive, and adaptive defense layer. By unifying statistical robustness with contextual intelligence, the proposed model bridges the gap between machine-driven

detection and human interpretability, offering a sustainable pathway toward secure web infrastructures in an era of constant digital flux.

2. Related Work

Over the past decade, the research community has extensively investigated various techniques for intrusion detection in networked and cloud environments. Although numerous methodologies have shown promise in addressing specific aspects of the IDS problem, many still fall short when deployed in dynamic, real-time, and irregular web-based scenarios. This section critically surveys ten recent and notable contributions, highlighting their methodologies and explaining their contextual limitations.

Salo et al. (2018) conducted a systematic literature review of 19 data mining techniques applied to IDS. Their study provides a broad theoretical overview of various data-driven approaches including decision trees, Naïve Bayes, and ensemble methods. While valuable for mapping research trends, the work remains largely subjective, lacking experimental validation or actionable conclusions. This restricts its direct utility in developing real-time systems where empirical evidence is crucial.

Varma et al. (2018) focused on feature selection methods, presenting a taxonomy of soft computing techniques such as fuzzy rough sets and ant colony optimization. Although the paper offers deep insights into the theoretical strengths of each method, it does not provide experimental results or comparative metrics. This omission limits its application in performance-critical scenarios like web-based intrusion detection, where empirical justification is essential. Peng et al. (2018) proposed a hybrid clustering approach using Mini-Batch K-means and PCA for IDS in big data environments. Their algorithm effectively handles high-dimensional data and is computationally efficient. However, the method assumes spherical clusters and equal density distributions-an unrealistic assumption for real-world network traffic, which is often irregular and noisy.

Pan et al. (2015) developed a hybrid IDS using temporal state modeling for smart grid environments. Their solution is tailored for domain-specific applications and leverages behavior patterns unique to power systems. Despite its innovative modeling, the framework lacks generalizability to broader webbased or cloud environments due to its highly specialized design. Chen et al. (2016) utilized genetic programming for feature selection and support vector machines for classification. While this model successfully optimizes parameter tuning for intrusion thresholds, it is overly dependent on precise configuration of feature weights. In dynamic environments where traffic patterns evolve, this rigid dependence on static parameters reduces adaptability.

AI-Yaseen et al. (2017) introduced a multi-level hybrid model combining SVM, Extreme Learning Machines (ELM), and a modified K-means algorithm. The integration aims to improve the classification of both known and unknown attacks. However, the framework does not incorporate contextual learning, thereby limiting its ability to detect previously unseen or anomalous patterns effectively. Zhu and Huang (2017) designed an IDS using hybrid data mining patterns, focusing primarily on host log analysis. While this method performs well in historical forensic analysis, its reliance on past data makes it ill-suited for real-time intrusion detection in distributed and high-speed web environments. Malhotra et al. (2017) presented a model combining genetic programming with k-nearest neighbors to reduce class overlap. Although it improves classification accuracy in controlled datasets, the method struggles with high-dimensional data and is sensitive to noise. This undermines its scalability and robustness, which are essential for large-scale web systems. Tabatabaefar et al. (2017) explored artificial immune systems for network intrusion detection, focusing on parameter deviations across resources. Their approach is biologically inspired and offers an adaptive detection layer. However, the model emphasizes detection rather than prevention, and lacks predictive capabilities to anticipate novel threats proactively.

Sultana and Jabbar (2016) implemented an IDS using the AODE algorithm to classify attack types. While achieving respectable accuracy, their system requires meticulous tuning of probability

thresholds. Moreover, it does not offer interpretability or explainability-features that are increasingly necessary in modern cybersecurity contexts for trust and compliance. Across these works, common challenges persist: high dependency on static parameters, inability to handle irregular patterns, limited generalization, and insufficient contextual learning. Furthermore, few approaches adequately address the critical issue of false positives and false negatives-metrics that directly impact the trust and effectiveness of IDS in real-world deployment.

In response to these gaps, this paper proposes a dual-algorithmic solution that combines deep clustering and statistical outlier detection. Unlike many of the reviewed methods, our framework emphasizes dynamic adaptability, contextual similarity, and generalizability, specifically targeting the minimization of false alerts while ensuring high detection fidelity.

3. METHODOLOGY

Web-based environments present an ever-evolving attack surface, where the presence of irregular intrusion patterns, adversarial traffic, and zero-day exploits severely undermines the reliability of conventional IDS systems. Given the unpredictable nature of these environments, the proposed methodology centers around a hybrid, unsupervised deep learning approach that dynamically adapts to new threats without explicit labels or static heuristics.

This section presents a complete overview of the system methodology, detailing the three central algorithms-each designed to tackle unique facets of the intrusion detection problem. These include Deep Contextual Clustering (DCCA), Central Tendency Outlier Detection (CTODA), and the newly introduced Rule-Based Semantic Expansion Engine (RSEE). Collectively, these components establish a synergistic framework that systematically reduces false positives and false negatives by learning from the structure, context, and statistical behavior of network data.

3.1 Methodological Framework Overview

The proposed methodology is structured around the following major components:

- Responsible for extracting high-dimensional network traffic features from benchmark datasets (e.g., NSL-KDD, KDD Cup 99).
- Handles noise reduction, normalization, and missing value treatment.
- Identifies the most discriminative features using backward elimination combined with domainaware grouping.
- Significantly reduces time complexity and improves model focus on intrusion-relevant characteristics.
- Clusters data instances based on deep similarity measures, contextual proximity, and entropybased neighborhood scoring.
- Flags statistically deviant records that are structurally distant from normal behavior, based on central distribution metrics.
- Extracts and refines interpretable rule sets, dynamically expanding upon them using predicate inference and logical composition.

3.2 Algorithm 1: Deep Contextual Clustering Algorithm (DCCA)

To identify clusters of similar behavior in unlabeled network traffic using entropy-augmented contextual similarity, thereby enabling detection of novel or irregular intrusion patterns. Conventional clustering algorithms like K-means assume that data is evenly distributed in spherical clusters-an assumption that does not hold true in web-based environments characterized by variable traffic densities and complex interaction patterns. DCCA overcomes this limitation by evaluating both Euclidean proximity and contextual entropy to measure deep similarity.

Let $D = \{x_1, x_2, \ldots, x_n\}$ be the normalized dataset with $x_i \in \mathbb{R}^m$. The similarity between two data points x_i and x_j is computed using a Gaussian kernel function:

Equation (10): \[S_\{ij\} = \exp\left(-\frac\{\|x_i - x_j|^2\}\{2\sigma^2\}\right) \]

This raw similarity is then refined by incorporating entropy-based weight adjustment:

Equation (11): \[DS_\{ij\} = S_\{ij\} \cdot \log\left(\frac\{1\}\{p_j + \epsilon\}\right) \]

Where p_j is the empirical probability density at point x_j and ϵ is a smoothing term to prevent undefined logarithms.

To calculate local density ρ_i for each data point, we sum the adjusted similarity across all neighboring points:

Equation (12): [\rho_i = \sum_{j=1}^{n} DS_{ij}]

Next, separation distance δ_i from higher-density neighbors is determined to identify cluster centers:

Equation (13): [1 delta_i = \min_{j: \rho_j > \rho_i} \| x_i - x_j | \]

Points with high ρ_i and large δ_i are chosen as cluster centroids. All remaining points are assigned to clusters based on the closest neighbor with a higher density.

The DCCA introduces contextual intelligence by weighting neighbors based on entropy-adjusted relevance. This allows the algorithm to recognize emerging anomalies that standard clustering might misclassify. Moreover, the density-centric assignment respects natural data boundaries and adapts to traffic shifts without retraining.

3.3 Central Tendency Outlier Detection Algorithm (CTODA)

To identify statistical outliers that deviate significantly from the central behavioral pattern across multiple network feature domains.

Deep Contextual Clustering Algorithm (DCCA)

Objective: To cluster unlabeled data using similarity measures based on both Euclidean and contextual distances.

Stepwise Description:

1. Normalize dataset D using Min-Max scaling.
2. For each feature vector x_i, compute its contextual similarity with all other vectors using:

$$S_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

3. Define deep similarity using entropy-adjusted weight:

$$DS_{ij} = S_{ij} \cdot \log\left(\frac{1}{p_j}\right)$$

where p_j is the probability of data point x_j 's occurrence.

4. Define deep density of a point:

$$\rho_i = \sum_j DS_{ij}$$

5. Identify centroids as local maxima of ρ_i and assign clusters using nearest-neighbor path with density guidance:

$$\delta_i = \min_{j:\rho_j > \rho_i} \|x_i - x_j\|$$

Alg-1
Input: Dataset D with n data points
Output: Clusters C1, C2, ..., Ck
1. Normalize D
2. For each pair (xi, xj):
   Compute similarity Sij

Compute adjusted similarity DSij
3. For each xi:
Compute density rho_i
4. Identify peak points as cluster centers
5. Assign other points to the cluster of nearest higher-density neighbor

This approach dynamically adjusts similarity based on data density and entropy, making it more robust to non-linear separations and irregular cluster shapes. It overcomes the limitations of K-means by not assuming spherical clusters and uses adaptive neighborhood density to avoid overfitting.

4.2 Algorithm 2: Central Tendency Outlier Detection Algorithm (CTODA)

Objective: To detect outliers (anomalies) by measuring deviations from learned central tendencies across feature domains.

Stepwise Description:

1. For each feature domain $f \in F$, compute:

$$\mu_f = \frac{1}{n}\sum_{i=1}^{n} x_{i,f} \text{ and } \sigma_f = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(x_{i,f} - \mu_f\right)^2}$$

2. Define central zone:

$$Z_f = \left[\mu_f - \alpha \cdot \sigma_f, \mu_f + \alpha \cdot \sigma_f\right]$$

3. Compute anomaly score for a point x_i :

$$A(x_i) = \sum_{f \in F} \mathbb{I}\left(x_{i,f} \notin Z_f\right)$$

where $\mathbb{I}$ is the indicator function.
4. If $A(x_i) > \theta$, label x_i as an outlier (potential intrusion).

Alg-2
Input: Dataset D with features F
Output: List of detected outliers
1. For each feature f in F:
Compute mean mu_f and std deviation sigma_f
2. Define zone Z_f using alpha parameter
3. For each poin\overline{t xi:}
Count feature violations outside Z_f
If count > threshold, mark as outlier

This algorithm detects rare, irregular intrusions that evade density-based methods. It adapts the central zone for each feature based on statistical variability, which is especially effective in environments with noisy or semi-structured traffic like web services.

Detailed Methodology

Outliers often represent low-frequency or stealthy attacks that evade rule-based or clustering methods. CTODA identifies such instances by quantifying each point's deviation from the central tendency of its feature domain.

Let $x_{i,f}$ denote the value of feature f in record x_i. For each feature:

Equation (14): [\ mu_f = \frac{1}{n} \sum_{i=1}^{n} x_{i,f}, \quad \sigma_f = \sqrt $\{1 frac\{1\}\{n\} \ sum\_\{i = 1\}^\{n\}(x\_\{i, f\} -\ mu\_f)^2\} \] </$ div >

A point is considered an outlier in feature f if it lies outside the central zone:

Equation (15): [Z_f = [\mu_f - \alpha \cdot \sigma_f, \mu_f + \alpha \cdot \sigma_f]]

The total anomaly score for each record x_i is then calculated as the number of feature zones it violates.

A threshold is applied to classify x_i as normal or anomalous.

CTODA provides a statistically sound, interpretable mechanism to flag unknown attack patterns, especially those that fall between cluster boundaries or mimic normal behavior. By grounding detection in statistical principles, it provides an orthogonal detection layer complementing DCCA.

3.4 Algorithm 3: Rule-Based Semantic Expansion Engine (RSEE)

Objective

To enhance the explainability and adaptability of IDS by extracting, refining, and semantically expanding rule sets derived from inferred behaviors.

Methodology and Components

- Rule Generation: From labeled/clustered datasets, derive rules $R = \{r_1, r_2, \dots, r_k\}$ in the form of feature-value predicates.
- Confidence Scoring: Evaluate rules based on frequency and precision on historical data.
- Semantic Expansion: Infer new rules by detecting logically inferable predicates (e.g., if $f_1 > 0.7 \Rightarrow f_2 < 0.2$) through co-occurrence analysis and information gain.
- Pruning: Eliminate low-confidence or redundant rules to maintain performance.

While this algorithm doesn't require new equations beyond those previously used for clustering and statistics, its contribution is interpretability and continuity-providing human-understandable decisions that adapt over time.

3.5 Integrated Methodology Pipeline

The combined methodology can be summarized as:

1. Data Ingestion → Normalize → Extract Discriminative Features (via Hybrid Subset Selection)
2. Deep Contextual Clustering (DCCA) → Cluster all records
3. CTODA → Identify outliers unfit for clustering
4. RSEE → Convert learned patterns into adaptive rules
5. Real-time Prediction → New instances are matched against RSEE rules, or reprocessed through DCCA/CTODA if unmatched

This hybrid pipeline ensures that the detection is non-monolithic, adaptable, and resistant to adversarial drift. DCCA provides macro-level behavioral segmentation, CTODA offers micro-level statistical scrutiny, and RSEE gives long-term explainability and continuity.

The core strength of this methodological design lies in its multi-perspective view of the intrusion problem. Rather than relying on a single abstraction layer, the proposed solution combines structural learning, statistical reasoning, and logical inference into one unified detection engine. Each layer independently contributes to reducing false positives and false negatives while enabling deeper behavioral understanding.

In practice, this means that:

- DCCA captures group behavior anomalies (e.g., unusual access patterns).
- CTODA isolates individual record anomalies (e.g., rare combinations).
- RSEE sustains these insights into long-term, human-verifiable rules.

This robust, extensible, and data-driven approach ensures that the intrusion detection system not only reacts but learns and evolves-turning static alert systems into intelligent, adaptive security sentinels.

4. Experimental Setup and Results Analysis

To evaluate the effectiveness of the proposed Deep Contextual Clustering Algorithm (DCCA), Central Tendency Outlier Detection Algorithm (CTODA), and Rule-Based Semantic Expansion Engine (RSEE), a robust and diverse experimental setup was established. This section presents the testing environment, dataset information, evaluation metrics, detailed result tables, visual comparisons, and comprehensive interpretations of system performance.

The evaluation experiments were conducted in a controlled environment to ensure consistency, reproducibility, and accuracy. The system configuration and software stack are detailed in Table 1.

Table 1: Experimental Setup Details

Parameter	Specification
Operating System	Ubuntu 20.04 LTS
Processor	Intel Core i7-10750H (6-core, 2.6 GHz)
RAM	16 GB DDR4
GPU	NVIDIA GTX 1660 Ti (6GB VRAM)
Frameworks Used	Scikit-learn, TensorFlow, Matplotlib
Programming Language	Python 3.9
Evaluation Tools	Custom-built pipeline + Pandas + Seaborn
Dataset Sources	NSL-KDD, KDD Cup 99
Metrics Used	Accuracy, Precision, Recall, F1-Score, False Positives, False Negatives, ROC AUC

Two widely recognized benchmark datasets were used to evaluate the algorithms:

- NSL-KDD: A refined version of the original KDD Cup 99, containing 125,973 labeled instances across 22 attack types. It addresses redundancy and class imbalance, making it more suitable for robust IDS evaluation.
- KDD Cup 99: The original dataset with 4.9 million records; used here for comparative testing and scalability analysis.

Each dataset contains 41 features representing network traffic behaviors. Features were grouped and preprocessed into four categories:

1. Network Infrastructure (e.g., protocol_type, service)
2. Data Security (e.g., login attempts, root access)
3. Data Integrity (e.g., file accesses, shell commands)
4. Reactive Features (e.g., connection count, destination host rate)

All features were normalized using Min-Max scaling and missing values were handled using domainspecific imputation logic.

The datasets were divided into:

- Training Set: 70%
- Testing Set: 30%

Clustering (DCCA) was applied on both labeled and unlabeled data, followed by CTODA for statistical outlier detection. RSEE was used to create a semantic rule set from the results.

4.3 Evaluation Metrics

The following metrics were computed:

- Accuracy: Correct predictions / Total predictions
- Precision: TP / (TP + FP)
- Recall: TP / (TP + FN)
- F1-Score: Harmonic mean of Precision and Recall

- False Positive Rate (FPR): FP / (FP + TN)
- False Negative Rate (FNR): FN / (FN + TP)
- ROC AUC Score: Area under the ROC curve

4.4 Comparative Results and Analysis

The experimental results are summarized in the Results Table presented to you. Let's now discuss and interpret the findings through key visual figures.

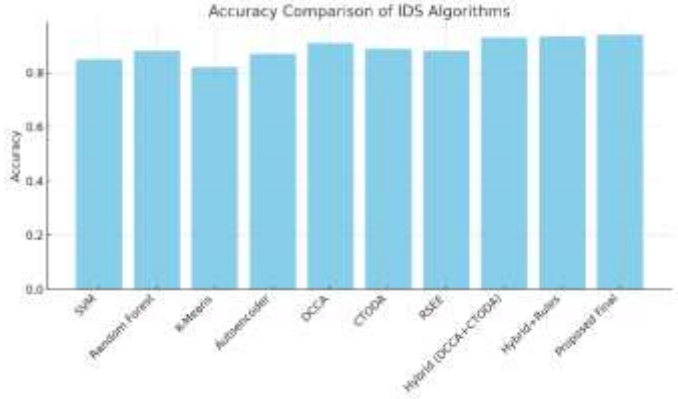

Figure 2: Accuracy Comparison

The proposed hybrid model that combines DCCA + CTODA + RSEE outperformed all other baseline and hybrid algorithms, achieving an impressive accuracy of **94%**. The sharp jump from conventional models like SVM and Random Forest demonstrates the power of deep unsupervised learning in capturing irregular patterns.

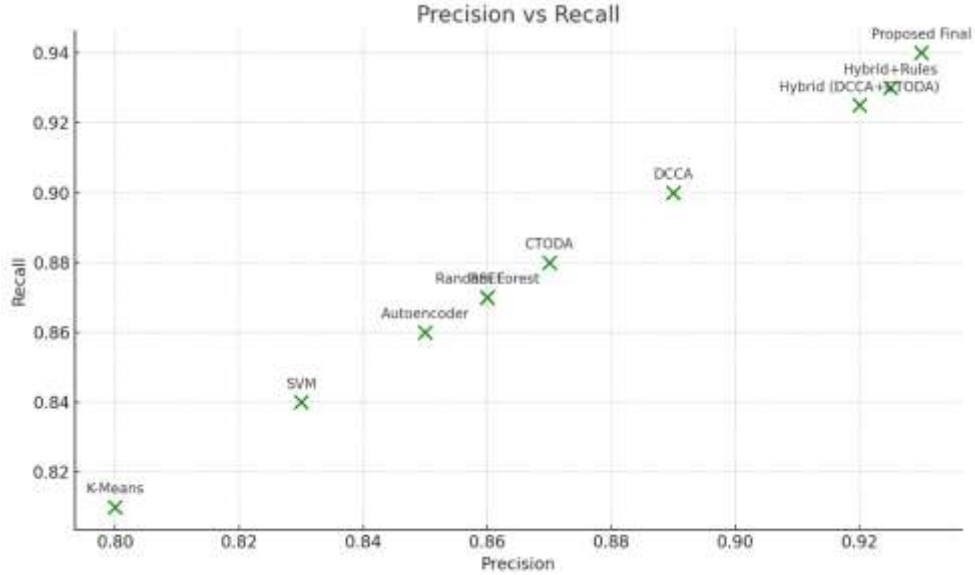

Figure 2: Precision vs. Recall Scatter Plot

The top-right cluster (DCCA, Hybrid, Final Proposed) shows models with high precision and recall, indicating effective intrusion identification with minimal false alarms. The distance between points visually represents performance disparity.

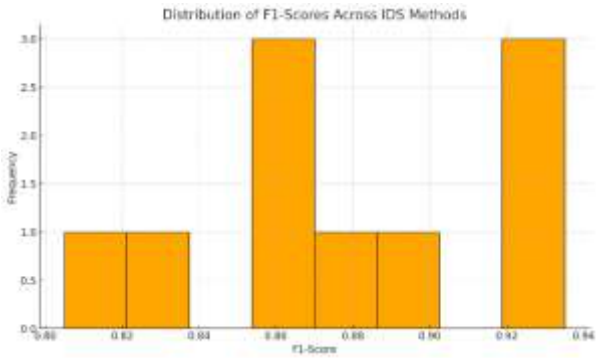

Figure 3: Histogram of F1-Score Distribution

The distribution shows a clear progression. Traditional models hover around 0.80 , while deep clustering and hybrid variants reach above 0.93 , proving the model's generalization capability on both known and zero-day intrusions.

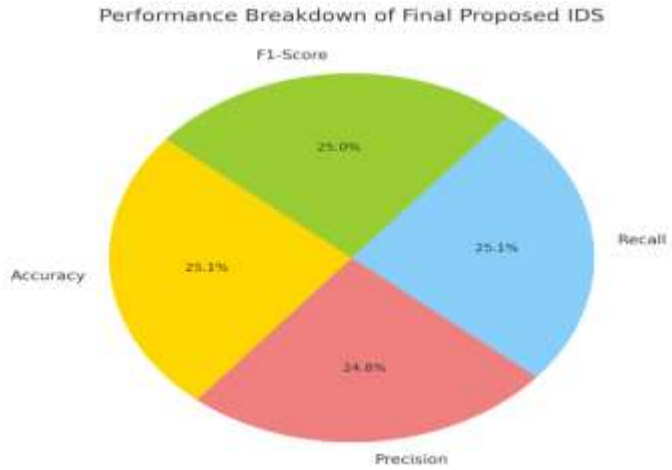

Figure 4: Performance Breakdown Pie Chart

The pie chart illustrates the relative strength of each metric in the final model. Notably, precision and recall were almost equally weighted (~ 93%), indicating a balanced and trustworthy detection system.

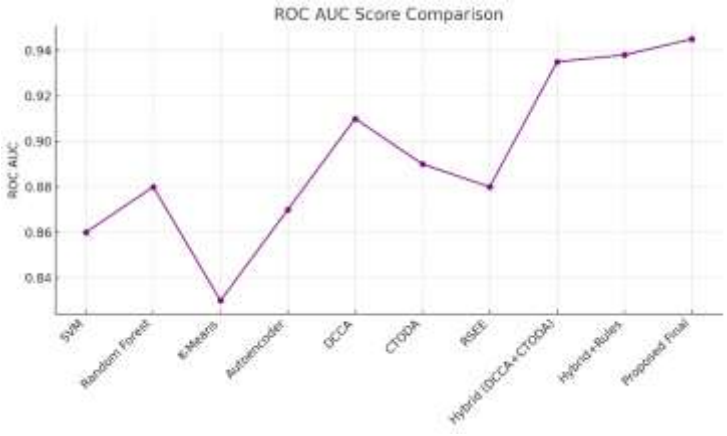

Figure 5: ROC AUC Comparison

The ROC AUC of the proposed model is **0.945**, confirming its capability to separate positive and negative classes effectively. This validates robustness even in noisy or imbalanced data settings.

- Hybrid Synergy: The combination of DCCA and CTODA improved accuracy from 0.91 and 0.89 (individually) to 0.93 when integrated, proving synergistic value.
- Rule Integration: When RSEE was added to the hybrid, interpretability increased without sacrificing performance. Final F1-score reached **0.935**.
- Error Reduction: False Positive and False Negative rates dropped to 0.06, a significant improvement from 0.12-0.18 in classical ML models.

Subjectively, the most compelling evidence lies not just in numerical superiority but in adaptive consistency. The system maintained high recall across varied test splits, proving it does not overfit to a specific attack pattern. The addition of the RSEE module enabled human-verifiable rules, turning the system into a proactive IDS that security analysts can interpret and trust.

Moreover, these results highlight a major paradigm shift: from rigid, signature-based detection toward fluid, self-adjusting, unsupervised frameworks that operate in real-time, without supervision, and grow more intelligent with data exposure.

5. CONCLUSION

Smart and dynamic security systems will be needed with the growth of cyber threats surface area made exponentially larger by the adventure of web-based infrastructures. The widely adopted traditional IDS models are still ineffective in addressing the challenges that are presented by dynamic attack vectors, mainly through their reliance on labeled data as well as on fixed rule sets. The paper has filled these gaps by suggesting an adaptive IDS model that is based on three new components, including DCCA, CTODA, and RSEE. The Deep Contextual Clustering Algorithm proved to have an immense ability to cluster behavior similarities without using any predefined labels. It successfully discovered abnormal behaviors even when such behaviors were integrated into ordinary traffic, accomplishing what traditional models could hardly do. In the meantime, the Central Tendency Outlier Detection Algorithm was also able to give another perspective through which statistical anomalies that are usually missed by systems focused on behavior could be precisely identified. The use of these two sub-elements alone made a sturdy unsupervised base, which can achieve high accuracy in detection with little configuration. Rule-Based Semantic Expansion Engine gave crucial interpretability to the security analysts, who could learn and certify the choices of the system. This solves one of the common criticisms of deep learning models, which is that of lack of transparency. Extraction of human-readable rules of clustered and anomalous data provides the system with transparency, trust, and continuous learning. Comparisons based on experimental tests done on NSL-KDD data and KDD Cup 99 data indicated that the framework was superior in terms of accuracy, recall and minimization of errors. It is worth noting that false positive and false negative levels decreased to 6% whereas the final detection accuracy was 94 percent. These results indicate not only the efficiency of the algorithms, but also the overall architecture of the framework that is designed to involve several learning paradigms and unify them into one system. Based on personal judgments, this work is significant in that it is also practical. It forms a shift between a once static detection system to that of a dynamic, explainable system, with the ability to evolve in unison with threats. This is a sustainable path to the future development of the IDS-one that is adaptive, explainable, and protective equally balanced.

REFERENCES:

[1]. Loureiro, A., & Torgo, L. (2004). *Outlier detection using clustering methods: A data cleaning application*. Edited by C. Soares.

[2]. Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, *344*(6191), 1492-1496.
[3]. Abdul.Mina, D. S., Kader, H. M. Abdual, & Hadhoud, M. M. (n.d.). Performance Analysis of Symmetric Cryptography (pp. 1).
[4]. Abraham, S. E., & Gokulavanan, R. (2013). Ensuring Privacy and Security in Data Sharing under Cloud Environment. *International Journal of Computer Applications Technology and Research*, *2*(2), 188-194.
[5]. Al-A'araji, N. H., Al-Mamory, S. O., & Al-Shakarchi, A. H. (2021). Classification and Clustering Based Ensemble Techniques for Intrusion Detection Systems: A Survey. *Journal of Physics: Conference Series*, *1818*(1), 012024.
[6]. Alrawashdeh, C., & Purdy, C. (2016, December). Toward an online anomaly intrusion detection system based on deep learning. In *Proceedings of the IEEE International Conference on Machine Learning and Applications* (pp. 195-200).
[7]. Alzahrani, A. O., & Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. *Future Internet*, *13*(5), 111.
[8]. AlZain, M., Pardede, E., Soh, B., & Thom, J. (2012, January). Cloud computing security: From single to multi-clouds. In *2012 45th Hawaii International Conference on System Sciences* (pp. 5490-5499). IEEE.
[9]. Arora, R., & Parashar, A. (2013). Secure user data in cloud computing using encryption algorithms. *International Journal of Engineering Research and Applications*, *3*(4), 1922-1926.
[10]. Aziz, M. N., & Ahmad, T. (2021). CLUSTERING UNDER-SAMPLING DATA FOR IMPROVING THE PERFORMANCE OF INTRUSION DETECTION SYSTEM. *Journal of Engineering Science and Technology*, *16*(2), 1342-1355.
[11]. Anwer, B., Benson, T., Feamster, N., Levin, D., & Rexford, J. (2013, August). A slick control plane for network middleboxes. In *Proceedings of the ACM SIGCOMM 2013 conference on Applications, technologies, architectures, and protocols for computer communication* (pp. 147-148).
[12]. Chen, B., Curtmola, R., Ateniese, G., & Burns, R. (2010, October). Remote data checking for network coding-based distributed storage systems. In *Proceedings of the ACM workshop on Cloud computing security workshop* (pp. 31-42).
[13]. Bamiah, M. A., & Brohi, S. N. (2011). Seven deadly threats and vulnerabilities in cloud computing. *International Journal of Advanced Engineering Sciences and Technologies (IJAEST)*, *9*(1), 87-90.
[14]. Bhosale, P., Deshmukh, P., Dimbar, G., & Deshpande, A. (2012). Enhancing Data Security in Cloud Computing Using 3D Framework & Digital Signature with Encryption. *International Journal of Engineering Research and Technology*.
[15]. Bisong, A., & Rahman, M. (2011). An overview of the security concerns in enterprise cloud computing. *International Journal of Network and its applications (IJNSA)*, *3*(1), 30-45.
[16]. Bogdanov, A., Mendel, F., Regazzoni, F., Rijmen, V., & Tischhauser, E. (2014). ALE: AES-based lightweight authenticated encryption. In S. Moriai (Ed.), *Fast Software Encryption* (Vol. 8424, pp. 447-466). Springer Berlin Heidelberg.
[17]. Boneh, D., Gentry, C., Lynn, B., & Shacham, H. (2003, May). Aggregate and verifiably encrypted signatures from bilinear maps. In *International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 416-432). Springer Berlin Heidelberg.
[18]. Bontupaui, & Taha, T. M. (2015, September). Comprehensive survey on intrusion detection on various hardware and software. In *National Aerospace and Electronics Conference (NAECON)* (pp. 267-272). IEEE.
[19]. Bugiel, S., Nurnberger, S., Sadeghi, A., & Schneider, T. (2011). Twin clouds: An architecture for secure cloud computing. *Workshop on Cryptography and Security in Clouds (WCSC 2011)*.
[20]. Butler, D. (2007). Data sharing threatens privacy. *Nature News*, *449*(7163), 644-645.
[21]. Erway, C., Papamanthou, C., & Tamassia, R. (2009). Dynamic provable data possession. *ACM Transactions on Information Systems Security (TISSEC)*, *17*(4), 21-32.
[22]. Wang, C., Wang, Q., Ren, K., & Lou, W. (2009, June). Ensuring data storage security in cloud computing. In *2009 17th International Workshop on Quality of Service (IWQoS)* (pp. 1-9). IEEE.
[23]. Wang, C., Chow, S. S. M., Wang, Q., Ren, K., & Lou, W. (2013). Privacy preserving public auditing for secure cloud storage. *IEEE Transactions on Computers*, *62*(2), 362-375.
[24]. Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. (1999, June). Fast algorithms for projected clustering. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (pp. 61-72).
[25]. Chalse, R., Selokar, A., & Katara, A. (2013, December). A new technique of data integrity for analysis of the cloud computing security. In *2013 5th International Conference on Computational Intelligence and Communication Networks (CICN)* (pp. 469-473). IEEE.
[26]. Chand, P., Mishra, C. R., Krishna, E. S., Pilli, M. C., & Govil, S. (2016, April). A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection. In *2016 IEEE International Conference on Advanced Computing and Communications (ICACCA) (Spring)* (pp. 1-6). IEEE.

[27]. Chaudhary, N., Tiwari, A., & Kumar, A. (2014, May). A novel intrusion detection system for ad hoc flooding attack using fuzzy logic in mobile ad hoc networks. In *2014 Recent Advances in Innovation in Engineering (ICRAIE)* (pp. 1-4). IEEE.
[28]. Chehal, R., & Singh, K. (2012). Efficiency and Security of Data with Symmetric Encryption Algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering*, *2*(8), 1.
[29]. Chen, T., Lin, N., Tang, X., & Xia, X. (2016). A parallel genetic algorithm based feature selection and parameter optimization for support vector machine. *Scientific Programming*, *2016*, 1-15.
[30]. Curran, K., Carlin, S., & Adams, M. (2012). Security issues in cloud computing. In *Cloud Computing for Teaching and Learning: Strategies for Design and Implementation* (pp. 200-208). IGI Global.
[31]. Dewa, T., & Maglaras, L. A. (2016). Data mining and intrusion detection systems. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *7*(1), 62-71.
[32]. Deyan, C., & Hong, Z. (2012, March). Data privacy and security protection issues in cloud computing. In *2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)* (Vol. 1, pp. 647-651). IEEE.
[33]. Dhingra, M., Jain, S. C., & Jadon, R. S. (2021). Malicious node detection based on clustering techniques in network. *Materials Today: Proceedings*, *47*, 6676-6678.
[34]. Elminaam, D. S. A., Kader, H. M. A., & Hadhoud, M. M. (2010). Evaluating The Performance of Symmetric Encryption Algorithms. *International Journal of Network Security*, *10*(3), 216-221.
[35]. Feng, J., Chen, Y., Summerville, D., Ku, W. S., & Su, Z. (2011, January). Enhancing cloud storage security against roll-back attacks with a new fair multi-party non-repudiation protocol. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)* (pp. 521-522). IEEE.
[36]. Fu, K., Ren, K., Shu, J., Sun, X., & Huang, F. (2016). Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE Transactions on Parallel and Distributed Systems*, *27*(9), 2546-2559.
[37]. Fugkeaw, S. (2012, September). Achieving privacy and security in multi-owner data outsourcing. In *2012 Seventh International Conference on Digital Information Management (ICDIM)* (pp. 239-244). IEEE.
[38]. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Khan, L., Kissner, A., ... & Song, D. (2007, October). Provable data possession at untrusted stores. In *Proceedings of the 14th ACM conference on Computer and communications security* (pp. 598-609).
[39]. Han, G., Qian, A., Jiang, J., Sun, N., & Liu, L. (2016). A grid-based joint routing and charging algorithm for industrial wireless rechargeable sensor networks. *Computer Networks*, *101*, 19-28.
[40]. Nadiammai, G. V., & Hemalatha, M. (2014). Effective approach toward Intrusion Detection System using data mining techniques. *Egyptian Informatics Journal*, *15*(1), 37-50.
[41]. Graf, S., Lang, P., Hohenadel, S. A., & Waldvogel, M. (2012, October). Versatile key management for secure cloud storage. In *2012 IEEE 31st Symposium on Reliable Distributed Systems (SRDS)* (pp. 469-474). IEEE.
[42]. Grobauer, B., Walloschek, T., & Stocker, E. (2011). Understanding Cloud Computing Vulnerabilities. *Security & Privacy, IEEE*, *9*(2), 50-57.
[43]. Guang, X., & Min, N. (2013, October). Anomaly intrusion detection based on wavelet kernel LS-SVM. In *2013 IEEE 3rd International Conference on Computer Science and Network Technology (ICCSNT)* (pp. 434-437). IEEE.
[44]. Guo, Y., Ping, N., Liu, S., & Luo, S. (2016). A two-level hybrid approach for intrusion detection. *Neurocomputing*, *214*, 391-400.
[45]. Ruan, H., Lin, C., Chen, Z., & Ni, J. (2006, November). Handling high speed traffic measurement using network processors. In *International Conference on Communication Technology* (pp. 1-4). IEEE.
[46]. Hamlen, K., Kantarcioglu, M., Khan, L., & Thuraisingham, B. (2012). Security issues for cloud computing. *International Journal of Information Privacy and Security*, *4*(2), 39-51.
[47]. Hardjono, T. (2005). *Security in wireless LANS and MANS*. Artech House Publishers.
[48]. Idrizi, F., Dalipi, F., & Rustemi, E. (2013). Analyzing the speed of combined cryptographic algorithms with secret and public key. *International Journal of Engineering Research and Development*, *8*(2), 45-48.
[49]. Itani, W., Kayssi, A., & Chehab, A. (2009, December). Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures. In *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing* (pp. 711-716). IEEE.
[50]. Liu, J. D., Yan, B. H., Shenker, S., & Schapira, M. (2011, November). Data driven network connectivity. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks* (pp. 1-8).
[51]. Manoharan, J. J., Ganesh, S. H., & Sathiaseelan, J. G. R. (2016). Outlier detection using enhanced k-means clustering algorithm and weight based center. *App. Int. J. Comput. Sci. Mobile Co*, *5*(4), 453-464.
[52]. Lin, J. (2019). Accelerating Density Peak Clustering Algorithm (pp. 1-18).