Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Reproducing a Machine Learning-Driven Access Control and Intrusion Detection Framework for Hybrid Cloud Environments

From Anonymity-Preserving Encryption to Adversarially Robust Detection

Fatima Muhammad Dawud
Department of Computer Engineering
University of Maiduguri,
Maiduguri, Nigeria.
Department of Software Engineering, Al-Ansar University
Maiduguri, Maiduguri, Nigeria

Abdulmumini Imam Ibrahim
Department of Computer Science
Ramat Polytechnic Maiduguri
Maiduguri, Nigeria.

***Abstract*—Hybrid cloud environments combine the elasticity of public infrastructure with the control of private deployments, but they also widen the surface through which data confidentiality, user privacy, and system availability can be attacked. This paper reports an independent reproduction of a four-part machine learning-driven security framework for hybrid clouds originally proposed by Mulimani et al. The reproduced components are an anonymity-preserving, collusion-resistant symmetric encryption scheme; a two-level supervised classification framework for role-based access control, together with an attribute-based access control architecture built on recursive feature elimination; a dynamic intrusion detection system combining Linear Discriminant Analysis feature reduction, an AdaBoost ensemble, and class-by-class K-means clustering; and an adversarial robustness analysis using FGSM and DeepFool attacks with same-family and cross-adversarial training. On the NSL-KDD dataset, the reproduction obtains 98.10 percent accuracy for Naive Bayes binary classification, 99.26 percent for multinomial logistic regression attack-category classification, 93.16 percent for the AdaBoost ensemble, and per-class F1 scores of 0.76 to 0.88 for hierarchical clustering. Undefended FGSM and DeepFool attacks reduce classifier accuracy from 91 to 99 percent to as low as 16 percent. Same-family adversarial training largely restores accuracy, whereas cross-adversarial training reveals strongly direction-dependent robustness, with the Support Vector Machine generalizing most consistently. The paper also documents the limitations of the reproduction, including reliance on a single dated dataset, a synthetic role attribute, and single-split evaluation.**

Keywords*—Hybrid cloud security; access control; attribute-based encryption; role-based access control; intrusion detection system; NSL-KDD; ensemble learning; adversarial machine learning; adversarial training; FGSM; DeepFool

I. INTRODUCTION

Cloud computing's on-demand, elastically scalable resource model spanning Infrastructure-, Platform-, and Software-as-a-Service delivery and public, private, hybrid, and community deployment [1], [2] has made it the default substrate for outsourced storage and computation. Hybrid clouds, which keep sensitive workloads on a private cloud while offloading auxiliary processing to a public cloud, are particularly attractive for organizations that need both cost efficiency and control [2]. However, cloud adoption remains constrained by security and privacy concerns: users lose direct control over data stored on semi-trusted third-party servers, and the Cloud Security Alliance's threat taxonomy ranks data breaches and weak identity, credential, and access management as the two leading threats to cloud deployments [3]. As cloud-hosted attack surfaces expand, machine learning (ML) has become a standard tool for the pattern-recognition tasks of anomaly classification, feature selection, and clustering-based grouping that traditional rule-based security mechanisms handle poorly at cloud scale [4].

This paper reproduces the integrated security framework for hybrid cloud environments proposed by Mulimani et al. [34]–[39], which is organized around four complementary objectives, each addressing a different layer of the cloud security stack: (1) preserving user anonymity and data confidentiality via hashed, attribute-derived encryption keys; (2) enforcing coarse- and fine-grained access control using supervised ML on role- and attribute-based access control data; (3) building a dynamic, multi-technique intrusion detection system; and (4) evaluating and improving that IDS's robustness to adversarial evasion. Each component was re-implemented following the original descriptions, and the results reported here are those obtained in this reproduction. Sections IV–VII present each component in turn; Section VIII discusses how they compose into a single security prototype.

A. Contributions

- An independent reproduction of the per-transaction anonymity scheme of [34], in which the cloud server, cloud users, and attribute authority cannot combine keys to de-anonymize a data owner or data consumer, implemented with symmetric (AES) encryption and hashed attribute/file keys.
- A reproduction of the two-level supervised-learning access-control framework of [35], which couples role-based access control (RBAC) role attributes with network traffic features to classify activity as normal or anomalous and then attribute anomalies to DoS, Probe, U2R, or R2L categories, together with the attribute-

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

based access control (ABAC) architecture of [36] that uses Recursive Feature Elimination to build mandatory/optional-attribute access policies.

- A reproduction of the dynamic intrusion detection pipeline of [37], combining linear discriminant analysis feature reduction, five supervised classifiers, an AdaBoost ensemble of the three best performers, and a class-by-class K-means procedure to recover attacks that supervised and ensemble models miss.
- A reproduction of the adversarial machine learning evaluation of [38], [39] against FGSM and DeepFool evasion attacks, including same-family adversarial training and cross-adversarial training (train on one attack family, test on another).
- A consolidated presentation of the four reproduced components, with an explicit account of the limitations that bound the generality of the results.

II. RELATED WORK

A. Cloud Security Foundations and Access Control

Comprehensive surveys of cloud security threats, architecture, and stakeholder concerns are well established [3]-[6], and comparative analyses of mandatory, discretionary, role-based, and attribute-based access control consistently demonstrate ABAC best suited to the scale and dynamism of cloud environments, at the cost of greater policy-management complexity [7], [8]. Anonymity- and privacy-preserving cloud storage schemes have used blind signatures, hash chains, group signatures, and decentralized access control to prevent cloud servers and third parties from linking transactions to identities [9]-[11], which is the design lineage of the anonymity model of [34] reproduced in Section IV. More recent work continues to explore ML-assisted ABAC: privacy-preserving "dummy-attribute" ABAC models that obfuscate real attributes before cloud transmission [12] and hybrid ML-based attribute-authentication schemes for resource-constrained IoT/edge settings using Random-Forest anomaly scoring over login and behavioral attributes [13] both extend the attribute-driven access control direction pursued here.

B. Machine Learning for Intrusion Detection

A large literature applies supervised (Naïve Bayes, Logistic Regression, SVM, Decision Tree, and Neural Network) and unsupervised (k-means and DBSCAN) and ensemble techniques to network intrusion detection on benchmark datasets, most commonly NSL-KDD, KDD Cup '99, and UNSW-NB15 [14]-[17]. Feature selection and reduction (e.g., SelectKBest, Recursive Feature Elimination, and Linear Discriminant Analysis) are consistently reported to improve both accuracy and training/testing time by removing redundant or irrelevant fields [17]-[19], consistent with the feature-reduction results reported in Sections V–VI. However, NSL-KDD, while still widely used as a compact, well-characterized cross-generational benchmark, is now more than two decades removed from its underlying traffic-capture era; recent evaluations recommend pairing it with, or replacing it for headline results with, more contemporary datasets such as CICIDS2017, CSE-CIC-IDS2018, or UNSW-NB15, which better reflect modern attack patterns and traffic volumes [20]-[23]. Cross-dataset evaluations further show that models trained on one benchmark can lose a large share of their accuracy, in some settings approaching chance-level performance, when evaluated on another, underscoring that single-dataset NSL-KDD results should be read as a controlled proof of concept rather than as evidence of field-ready generalization [22], [23].

C. Adversarial Machine Learning for Intrusion Detection

Adversarial machine learning (AML), originally studied chiefly in image classification, has become an active sub-field of research on network intrusion detection, with two comprehensive 2023 surveys (He, Kim, and Asghar; Alotaibi and Rassam) establishing the now-standard taxonomy of white-/gray-/black-box threat models, evasion/poisoning/extraction attack types, and defensive countermeasures for ML-based NIDS [24], [25]. Attack generation methods used in this taxonomy, FGSM, PGD, the Basic Iterative Method, JSMA, DeepFool, and Carlini-Wagner, have each been shown capable of substantially degrading NIDS accuracy on NSL-KDD-family and other benchmark datasets, and adversarial training remains the most consistently effective single defense, though at some cost to clean-data accuracy [26]-[28]. Since 2023, the field has moved toward (a) more realistic, feature-space-constrained attacks that respect protocol validity rather than perturbing arbitrary features [29], (b) multi-stage defenses that combine adversarial training with complementary techniques such as input preprocessing and feature squeezing [30], [31], and (c) broader systematic reviews and surveys synthesizing threat models, defenses, and open challenges [32], [33]. Within this landscape, cross-adversarial training, training a model on one attack family and evaluating it on another, as proposed in [38], [39] and reproduced in Section VII, remains comparatively understudied relative to same-family adversarial training.

III. SYSTEM AND THREAT MODEL

The framework assumes a hybrid cloud consisting of four entity types: data owners, who create and upload files; data consumers, who request and download files; a cloud server, which stores files and mediates all transactions; and one or more attribute authorities, each responsible for a disjoint subset of user attributes and for issuing encryption/decryption keys. The cloud server is modeled as honest-but-curious: it correctly executes storage and retrieval operations but may attempt to infer sensitive information (user identity and access patterns) from the transactions it observes. Attackers are classified as internal (an authenticated user attempting an operation beyond their assigned role or attributes) or external (an unauthenticated party attempting to access resources). This threat model motivates the four-layer defense pursued in Sections IV–VII: anonymization defeats server-side inference even under honest-but-curious observation; ML-based RBAC/ABAC access control detects and blocks internal attackers who retain valid credentials; the dynamic IDS detects network-level attacks regardless of attacker origin; and adversarial-robustness analysis addresses attackers who additionally attempt to evade the IDS itself.

IV. USER ANONYMITY AND DATA SECURITY

A. Proposed Methodology

Following the scheme of [34], to keep the cloud server oblivious to which user performs a given read, write, or delete operation, each user's private key is constructed from two hashed components: an attribute key, derived from the user's

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

attribute set and issued once per user by the attribute authority, and a file-specific key, generated fresh for every upload/download transaction as a hash of the username, user ID, and a randomly chosen partition (of size at most a configurable threshold H) of the file being transacted. Because the file-specific component changes with each transaction while the attribute component remains stable for a single user, the combined key varies for every transaction observed by the cloud server, which prevents the server from linking multiple transactions to the same user. Files are encrypted with a symmetric algorithm (AES) prior to upload; only a data consumer who obtains the correct combination of attribute key and file key from the attribute authority can decrypt the retrieved ciphertext. Because the cloud server, individual users, and the attribute authority each hold only a partial view of the key material and are prevented from combining their partial keys, the scheme is collusion-resistant: no pairwise or full collusion among these entities is sufficient to recover a user's identity or file contents.

B. Formal Operations

The scheme is specified by four algorithms executed by the attribute authority and cloud entities: Setup(security parameter) -> (PK, MK) generates a public key and master key; KeyGen(PK, MK, A_u) -> SK_u issues a user u's private key from their attribute set A_u; Encrypt(PK, M) -> CT produces ciphertext from a public key and message; and Decrypt(PK, SK_u, CT) -> M recovers the plaintext given a valid private key. In implementation, the data owner requests the public key, encrypts the file locally, and uploads the ciphertext; the data consumer requests the private key (the combined attribute key and file key), downloads the ciphertext, and decrypts it locally.

C. Results and Discussion

The scheme was implemented with AES symmetric encryption and Base64 encoding for key transport; sample-generated public and private keys confirmed correct key derivation and transaction-unique key generation across repeated upload/download cycles. Because every transaction is presented to the cloud server as belonging to a distinct, unlinkable identity, the server cannot build a usage profile for any individual user even though it faithfully stores and serves the (encrypted) data. This component reproduces the scheme originally proposed in [34].

V. MACHINE LEARNING-BASED ACCESS CONTROL MECHANISMS

A. Two-Level Role-Based Access Control Framework

Because centrally administered RBAC does not scale to the number of users and roles typical of a dynamic cloud environment (leading to role- and permission explosion), the framework of [35] adopts a two-level supervised-learning approach over network traffic data, annotated with a synthetic role attribute (admin, superuser, data owner, etc.), to identify which role performed an anomalous action. In Level 1, a feature-selection step (SelectKBest) reduces the NSL-KDD feature set from 44 to the 23 highest-scoring features, which are combined with the role attribute into a 24-feature custom dataset; supervised classifiers (Naïve Bayes and Logistic Regression) then label each record as normal or anomalous. In Level 2, records flagged as anomalous in Level 1 are further classified into DoS, Probe, U2R, or R2L using the same two classifiers, with logistic regression reconfigured for multinomial (softmax/lbfgs) multi-class output.

Table I reports the resulting performance. Naïve Bayes performs best for Level 1 binary classification (98.10% accuracy), while multinomial logistic regression performs best for Level 2 attack-category classification (99.26% accuracy).

TABLE I. PERFORMANCE OF LEVEL 1 (NORMAL/ANOMALY) AND LEVEL 2 (ATTACK-CATEGORY) CLASSIFICATION IN THE TWO-LEVEL RBAC FRAMEWORK

Algorithm	Level	Accuracy	Precision	Recall	F-score
Naïve Bayes	1 (binary)	0.9810	1.0000	0.9646	0.9810
Naïve Bayes	2 (multi-class)	0.7357	0.4029	0.6204	0.4005
Logistic Regression	1 (binary)	0.9416	0.9307	0.9623	0.9411
Logistic Regression	2 (multi-class)	0.9926	0.9913	0.8621	0.9059

Table II presents the averaged performance across both levels for each classifier, showing logistic regression as the stronger overall performer once both binary and multi-class stages are combined.

TABLE II. AVERAGE PERFORMANCE ACROSS BOTH LEVELS OF THE PROPOSED TWO-LEVEL RBAC FRAMEWORK

Algorithm	Accuracy	Precision	Recall	F-Score
Naïve Bayes	0.8584	0.7015	0.7925	0.6908
Logistic Regression	0.9671	0.9610	0.9122	0.9235

B. Attribute-Based Access Control Architecture

Because Level 1/2 classification identifies only the offending role, not the specific user, fine-grained control requires attribute-based access control (ABAC). In the ABAC architecture of [36], an attribute authority applies Recursive Feature Elimination (RFE) to rank all available user/resource attributes, from which the data owner selects a set of mandatory attributes and k of n lower-ranked optional attributes to build an access-control policy using AND logical connectives (e.g., (protocol_type = 'tcp' AND role = 'Admin') OR (service = 'auth')). A data consumer is granted access to a file only if their attributes satisfy the associated policy; the mandatory-attribute/k-of-n-optional-attribute structure lets the data owner tune the strictness of access without hand-authoring a full Boolean policy over the entire attribute space.

This component reproduces the classification framework of [35]; the ABAC architecture follows the design of [36].

VI. DYNAMIC INTRUSION DETECTION SYSTEM

A. Proposed Methodology

Traditional signature-based and static anomaly-based IDS approaches struggle against a continuously expanding attack space. The dynamic IDS of [37], as reproduced here, reduces the NSL-KDD feature set from 44 to 21 features using Linear Discriminant Analysis (LDA), which maximizes between-class variance relative to within-class variance and ranks features via Fisher's criterion. The reduced feature set is used to train five supervised classifiers (logistic regression, Naïve Bayes, neural network, decision tree, and SVM) for multi-class attack classification; the three best-performing classifiers are then combined with an AdaBoost ensemble to further improve

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

accuracy. Because supervised and ensemble models can still miss some attacks, a class-by-class unsupervised stage applies K-means (k=2) hierarchically, first separating DoS from not-DoS; then U2R from not-U2R within the not-DoS group; then R2L from not-R2L; and finally Probe from Normal rather than a single flat k=4 clustering, which is shown to substantially improve per-class detection.

B. Results: Feature Reduction Effect

TABLE III. CLASSIFIER PERFORMANCE METRICS BEFORE (44 FEATURES) AND AFTER (21 FEATURES, LDA-SELECTED) FEATURE REDUCTION

Classifier	Precision (44 ft.)	Recall (44 ft.)	F-Score (44 ft.)	Precision (21 ft.)	Recall (21 ft.)	F-Score (21 ft.)
Naïve Bayes	0.81	0.83	0.79	0.95	0.91	0.81
Logistic Regression	0.86	0.79	0.65	0.89	0.92	0.91
Neural Network	0.89	0.73	0.81	0.78	0.69	0.72
SVM	0.74	0.86	0.91	0.84	0.81	0.85
Decision Tree	0.91	0.85	0.93	0.93	0.78	0.82

Table IV compares overall accuracy before and after feature reduction; logistic regression, naïve Bayes, decision tree, and SVM all improve, while neural network accuracy declines slightly, suggesting the reduced feature set trades some of the representational capacity a neural model can otherwise exploit for the interpretability/efficiency gains that benefit the other four classifiers.

TABLE IV. ACCURACY COMPARISON BEFORE AND AFTER LDA FEATURE REDUCTION

Classifier	Accuracy before Reduction (%)	Accuracy after Reduction (%)
Naïve Bayes	74.52	92.66
Logistic Regression	80.68	93.54
Neural Network	83.82	81.57
SVM	65.26	79.39
Decision Tree	84.36	91.65

C. Results: Ensemble and Class-by-Class Unsupervised Detection

The AdaBoost ensemble of Naïve Bayes, Logistic Regression, and Decision Tree (Table V) outperforms every individual classifier's overall accuracy in Table IV, confirming the expected benefit of combining diverse weak learners.

TABLE V. PERFORMANCE OF THE ADABOOST ENSEMBLE COMBINING NAÏVE BAYES, LOGISTIC REGRESSION, AND DECISION TREE

Classifier	Accuracy (%)	Precision	Recall	F-Score
AdaBoost Ensemble (NB + LR + DT)	93.16	0.85	0.90	0.88

Flat k=4 K-means clustering (Table VI) performs markedly worse than supervised or ensemble methods, particularly on U2R and probes. The class-by-class, hierarchical k=2 K-means procedure (Table VII) recovers substantially better per-class performance, confirming that reformulating multi-class unsupervised detection as a sequence of binary separations is the more effective unsupervised strategy for this feature space.

TABLE VI. FLAT (K=4) K-MEANS CLUSTERING PERFORMANCE PER ATTACK CLASS

Attack Class	Accuracy (%)	Precision	Recall	F-score
DoS	64.22	0.65	0.45	0.12
U2R	55.39	0.59	0.39	0.35
R2L	73.41	0.69	0.56	0.44
Probe	62.36	0.58	0.22	0.36

TABLE VII. CLASS-BY-CLASS HIERARCHICAL (K=2 PER LEVEL) K-MEANS CLUSTERING PERFORMANCE FOR EACH ATTACK CLASS

Attack Class	Precision	Recall	F-score
DoS	0.90	0.88	0.79
U2R	0.93	0.96	0.88
R2L	0.98	0.88	0.79
Probe	0.96	0.88	0.76

This component reproduces the approach of [37].

VII. ADVERSARIAL MACHINE LEARNING FOR INTRUSION DETECTION

A. Proposed Methodology

The dynamic IDS of Section VI assumes clean training and test data. In practice, an adversary aware of the deployed classifier can craft small, targeted perturbations to evade detection. Following [38], [39], we evaluate the robustness of Naïve Bayes, Logistic Regression, SVM, and a deep neural network (five hidden layers, 2,688 neurons total, ReLU activations, softmax output, categorical cross-entropy loss, and Adam optimizer) trained on clean NSL-KDD data, then attacked using the Fast Gradient Sign Method (FGSM) and DeepFool, generated via the IBM Adversarial Robustness Toolbox (ART). Four experimental conditions are evaluated: (a) clean train / clean test (baseline); (b) clean train / adversarial test (adversarial machine learning, i.e., undefended evasion); (c) adversarial train / adversarial test, same attack family (adversarial training, i.e., same-family defense); and (d) adversarial train on one attack family / adversarial test on the other (cross-adversarial training, i.e., generalized defense).

B. Attack Parameterization

FGSM was run untargeted with epsilon = 0.3; DeepFool was run untargeted with nb_grads = 10 and overshoot = 0.0001 (Table VIII). Sweeping the FGSM epsilon parameter (Table IX) shows accuracy collapsing from 97.17% at epsilon = 0 to 16-25% for any epsilon >= 0.1, indicating that the deep model is highly sensitive even to small perturbation budgets. DeepFool's runtime, in contrast, falls sharply as the overshoot parameter increases (Table X), since a larger overshoot lets the search terminate in fewer iterations.

TABLE VIII. PARAMETER SETTINGS FOR THE FGSM AND DEEPFOOL ATTACK GENERATION METHODS

Attack Method	nb_grads	Iterations	Targeted	Epsilon / Overshoot
FGSM	-	-	No	0.3
DeepFool	10	100	No	0.0001

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

TABLE IX. DEEP NEURAL NETWORK ACCURACY UNDER FGSM ATTACKS FOR VARYING EPSILON

Epsilon	Deep Model Accuracy (%)
0.0	97.17
0.1	23.80
0.2	25.16
0.3	22.67
0.4	16.20
0.5	17.33
0.6	22.52
0.7	22.33
0.8	21.80
0.9	22.36

TABLE X. DEEPFOOL GENERATION TIME FOR VARYING OVERSHOOT PARAMETERS

Overshoot (epsilon)	Time Taken (ms)
0.0001	1031.65
0.001	552.23
0.01	501.95
0.1	433.92
0.2	279.06

C. Results: Undefended Evasion (Adversarial Machine Learning)

Table XI compares clean-data accuracy against accuracy under undefended FGSM and DeepFool attacks. Every classifier loses the large majority of its clean-data accuracy: SVM is the most resilient (68.36% under FGSM and 29.74% under DeepFool versus a clean-data ceiling of 99.15%), while Naïve Bayes and the deep neural network are the least resilient to FGSM (16.36% and 16.37%) and the deep neural network is the least resilient to DeepFool (20.41%).

TABLE XI. ACCURACY OF ML/DL MODELS ON CLEAN VS. UNDEFENDED ADVERSARIAL (DEEPFOOL, FGSM) TEST DATA

Classifier	Clean Data (%)	DeepFool (%)	FGSM (%)
Naïve Bayes	91.15	22.43	16.36
Logistic Regression	97.03	22.43	25.92
Support Vector Machine	99.15	29.74	68.36
Deep Neural Network	97.44	20.41	16.37

D. Results: Same-Family Adversarial Training

Training on adversarial examples from the same attack family used at test time (Table XII) restores most classifiers to near-clean-data accuracy, with Logistic Regression, SVM, and DNN all recovering to 92-97% under both attack families; Naïve Bayes recovers only partially (77.6-83.6%), consistent with its comparatively simple decision boundary offering less capacity to absorb adversarial training.

TABLE XII. ACCURACY COMPARISON ACROSS CLEAN-DATA BASELINE, UNDEFENDED ADVERSARIAL EVALUATION (AML), AND SAME-FAMILY ADVERSARIAL TRAINING

Classifier	Clean (%)	AML DeepFool (%)	AML FGSM (%)	Adv. Training DeepFool (%)	Adv. Training FGSM (%)
Naïve Bayes	91.15	22.43	16.36	83.64	77.56
Logistic Regression	97.03	22.43	25.92	96.99	97.13
Support Vector Machine	99.15	29.74	68.36	92.20	97.14
Deep Neural Network	97.44	20.41	16.37	95.50	97.07

E. Results: Cross-Adversarial Training

Cross-adversarial training, training on one attack family and testing on the other, probes whether a defense generalizes beyond the specific attack it was hardened against. Training on FGSM and testing on DeepFool (Table XIII) yields moderate, fairly uniform accuracy (69.9-84.3%) across all four classifiers. Training on DeepFool and testing on FGSM (Table XIV) is far more asymmetric: SVM generalizes well (88.44%) and Naïve Bayes moderately (77.56%), but Logistic Regression (29.56%) and the deep neural network (32.14%) generalize poorly, indicating that DeepFool hardening does not transfer to FGSM robustness for gradient-sensitive models to the same degree it does for margin-based SVM.

TABLE XIII. MODELS TRAINED ON FGSM ADVERSARIAL DATA, TESTED ON DEEPFOOL ADVERSARIAL DATA

Classifier	Accuracy (%)	Precision	Recall	F1-Score
Naïve Bayes	83.65	0.87	0.82	0.82
Logistic Regression	84.34	0.86	0.83	0.83
Support Vector Machine	69.87	0.80	0.67	0.65
Deep Neural Network	84.13	0.87	0.83	0.83

TABLE XIV. MODELS TRAINED ON DEEPFOOL ADVERSARIAL DATA, TESTED ON FGSM ADVERSARIAL DATA

Classifier	Accuracy (%)	Precision	Recall	F1-Score
Naïve Bayes	77.56	0.83	0.76	0.75
Logistic Regression	29.56	0.21	0.31	0.24
Support Vector Machine	88.44	0.89	0.87	0.88
Deep Neural Network	32.14	0.22	0.34	0.25

Overall, SVM emerges as the most consistently robust classifier across both directions of cross-adversarial training, while Logistic Regression and the DNN are strongly direction-dependent: they remain fairly robust when trained on FGSM and tested on DeepFool, but generalize poorly in the reverse direction. This directional asymmetry indicates that same-family adversarial training alone is an insufficient robustness guarantee. This component reproduces the cross-adversarial training methodology of [38], [39].

VIII. INTEGRATED DISCUSSION: TOWARDS A UNIFIED SECURITY PROTOTYPE

The four components address distinct but complementary layers of the hybrid cloud threat model articulated in Section III. Anonymity-preserving encryption (Section IV) protects confidentiality and privacy against an honest-but-curious cloud

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

server, independent of any attack on the network layer. ML-based RBAC/ABAC (Section V) constrains what an authenticated-but-malicious insider can do and provides fine-grained, attribute-driven authorization beyond what static roles allow. The dynamic IDS (Section VI) detects network-level attacks, including those from authenticated insiders who evade access-control constraints, using a feature-reduced, ensemble-strengthened, class-by-class-recovered pipeline. Finally, the adversarial-robustness analysis (Section VII) closes the loop by testing whether the IDS itself can be evaded and by how much adversarial and cross-adversarial training can restore its reliability. Used together, these four components constitute a defense-in-depth prototype: even if one layer is bypassed (e.g., an attacker obtains valid credentials, defeating the anonymity and access-control layers), the IDS and its adversarial-hardening provide an independent detection backstop. A production deployment would need to integrate the four components into a single pipeline with shared telemetry and a consistent evaluation dataset, which Section IX identifies as a priority for future work.

IX. LIMITATIONS

The reproduction has several limitations that bound the generality of its conclusions.

- *Single, dated benchmark dataset:* all components that use ML for detection or classification (Sections V–VII) rely exclusively on NSL-KDD, which is derived from 1998–99 DARPA traffic captures. Validation on contemporary datasets such as CICIDS2017, CSE-CIC-IDS2018, or UNSW-NB15 [16], [17], [22], [23] is needed to establish generalization to modern attack patterns and traffic volumes.
- *Synthetic role attribute:* the role feature used in the two-level RBAC framework (Section V-A) is randomly assigned rather than drawn from a real organizational RBAC deployment, so the reported role attribution reflects a controlled simulation rather than genuine role–activity correlations.
- *Statistical validation:* all reported metrics are single-run point estimates on a fixed train/test split. k-fold cross-validation with mean and standard deviation, together with significance testing between classifiers, would strengthen the comparisons.
- *Hyperparameter reporting:* the SVM kernel, the decision tree depth and pruning settings, the Section VI neural network architecture, and the per-layer unit distribution of the Section VII deep neural network are not fully reported, which limits exact re-reproduction.
- *Attack coverage:* only FGSM and DeepFool were evaluated, both under white-box assumptions. Stronger iterative attacks (PGD, Basic Iterative Method, Carlini-Wagner, and JSMA) and black-box methods were not considered.
- *Defense baselines:* the adversarial-training results are compared only internally (clean, undefended, same-family, and cross-adversarial conditions), not against recent published defenses such as [30], [31].
- *Fragmented pipeline:* the four components are evaluated as separate experiments. This paper therefore presents four reproduced components toward an integrated system rather than an end-to-end evaluation of that system.
- *Cloud-specific evaluation:* the anonymity scheme (Section IV) and the ABAC architecture (Section V-B) are validated by construction and illustrative examples rather than by a formal security analysis or quantitative overhead benchmarks.

X. CONCLUSION AND FUTURE WORK

This paper reproduced the four-part machine learning-driven security framework for hybrid cloud environments proposed in [34]–[39]: an anonymity-preserving, collusion-resistant encryption scheme; a two-level ML-based RBAC framework paired with an RFE-driven ABAC architecture; a dynamic IDS that combines LDA feature reduction, ensemble learning, and class-by-class unsupervised recovery; and an adversarial machine learning evaluation, including cross-adversarial training, of the IDS's robustness against FGSM and DeepFool evasion. Across the reproduced components, the main empirical findings are that (i) feature reduction and ensemble learning meaningfully improve NSL-KDD-based intrusion detection, (ii) a hierarchical class-by-class unsupervised procedure substantially outperforms flat multi-class clustering for recovering missed attacks, and (iii) adversarial robustness does not transfer uniformly across attack families or classifiers: SVM generalizes best under cross-adversarial training, while Logistic Regression and the deep neural network are strongly direction-dependent. Future work should prioritize (1) validation on a contemporary traffic dataset (CICIDS2017/2018, UNSW-NB15) alongside NSL-KDD; (2) cross-validated, statistically tested results; (3) an expanded, partly black-box adversarial attack suite; (4) a quantitative security and performance evaluation of the anonymity and ABAC components; and (5) integration of the four components into a single, end-to-end evaluated pipeline.

DATA AVAILABILITY STATEMENT

The NSL-KDD dataset used in Sections V–VII is publicly available from the Canadian Institute for Cybersecurity, University of New Brunswick [14]. The reproduction code is available from the corresponding author upon reasonable request.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST Special Publication 800-145, Sep. 2011.

[2] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proc. 5th Int. Joint Conf. INC, IMS and IDC (NCM)*, 2009, pp. 44–51.

[3] Cloud Security Alliance Top Threats Working Group, "The Treacherous 12: Cloud computing top threats in 2016," Cloud Security Alliance, Feb. 2016.

[4] N. Gonzalez et al., "A quantitative analysis of current security concerns and solutions for cloud computing," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 1, Art. no. 11, 2012.

[5] M. Almorsy, J. Grundy, and I. Müller, "An analysis of the cloud computing security problem," in *Proc. APSEC 2010 Cloud Workshop*, Sydney, Australia, 2010.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

[6] F. Sabahi, "Cloud computing security threats and responses," in *Proc. IEEE 3rd Int. Conf. Communication Software and Networks (ICCSN)*, 2011, pp. 245–249.

[7] K. Punithasurya and S. Jeba Priya, "Analysis of different access control mechanism in cloud," *International Journal of Applied Information Systems*, vol. 4, no. 2, pp. 34–39, Sep. 2012, doi: 10.5120/ijais12-450660.

[8] V. C. Hu et al., "Guide to attribute based access control (ABAC) definition and considerations," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST Special Publication 800-162, Jan. 2014 (updated Feb. 2019).

[9] J. Xiong, Z. Yao, J. Ma, X. Liu, Q. Li, and J. Ma, "PRIAM: Privacy preserving identity and access management scheme in cloud," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 1, pp. 282–304, 2014, doi: 10.3837/tiis.2014.01.017.

[10] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 384–394, 2014.

[11] L. Malina and J. Hajny, "Efficient security solution for privacy-preserving cloud services," in *Proc. 36th Int. Conf. Telecommunications and Signal Processing (TSP)*, 2013, pp. 23–27.

[12] M. Baby, I. Memon, F. A. Alvi, U. Rajput, and M. Nabi, "Machine learning-driven security and privacy analysis of a Dummy-ABAC model for cloud computing," *Computers*, vol. 14, no. 10, Art. no. 420, 2025, doi: 10.3390/computers14100420.

[13] J. Saleem, U. Raza, M. Hammoudeh, and W. Holderbaum, "Machine learning-enhanced attribute-based authentication for secure IoT access control," *Sensors*, vol. 25, no. 9, Art. no. 2779, 2025, doi: 10.3390/s25092779.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications (CISDA)*, 2009, pp. 1–6.

[15] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives," in *Proc. IEEE 3rd Int. Conf. Computing, Communication and Security (ICCCS)*, 2018, pp. 1–8.

[16] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116.

[17] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. Military Communications and Information Systems Conf. (MilCIS)*, 2015, pp. 1–6.

[18] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[19] H. M. Harb and A. S. Desuky, "Selecting optimal subset of features for intrusion detection systems," *Advances in Computational Sciences and Technology*, vol. 4, no. 2, pp. 179–192, 2011.

[20] T. H. Sow and A. Mehdi, "Enhancing IDS performance through a comparative analysis of random forest, XGBoost, and deep neural networks," *Machine Learning with Applications*, vol. 22, Art. no. 100738, 2025.

[21] S. Debnath and R. Das, "A hybrid CNN-LSTM intrusion detection framework for cybersecurity in smart renewable energy grids," arXiv:2606.25200, Jun. 2026.

[22] C. Xin and K. Xu, "Cross-dataset Transformer-IDS with calibration and AUC optimization (evaluated on NSL-KDD, UNSW-NB15, CIC-IDS2017)," *Journal of Cyber Security*, vol. 7, no. 1, pp. 483–503, 2025, doi: 10.32604/jcs.2025.071627.

[23] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model generalization for intrusion detection: Unsupervised machine learning techniques," *Journal of Network and Systems Management*, vol. 30, no. 1, Art. no. 12, 2022, doi: 10.1007/s10922-021-09615-7.

[24] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.

[25] A. Alotaibi and M. A. Rassam, "Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense," *Future Internet*, vol. 15, no. 2, Art. no. 62, 2023.

[26] I. Debicha, T. Debatty, J.-M. Dricot, and W. Mees, "Adversarial training for deep learning-based intrusion detection systems," in *Proc. 16th Int. Conf. Systems (ICONS)*, 2021, pp. 45–49.

[27] W. D. Xiong, K. L. Luo, and R. Li, "AIDTF: Adversarial training framework for network intrusion detection," *Computers & Security*, vol. 128, Art. no. 103141, 2023, doi: 10.1016/j.cose.2023.103141.

[28] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367–38384, 2018.

[29] J. Vitorino, I. Praça, and E. Maia, "SoK: Realistic adversarial attacks and defenses for intelligent network intrusion detection," *Computers & Security*, vol. 134, Art. no. 103433, 2023, doi: 10.1016/j.cose.2023.103433.

[30] K. Roshan and A. Zafar, "Boosting robustness of network intrusion detection systems: A novel two phase defense strategy against untargeted white-box optimization adversarial attack," *Expert Systems with Applications*, vol. 249, Art. no. 123567, 2024.

[31] T. T. Nguyen, U. H. Tran, and H. N. Nguyen, "Adversarial attack and defense in AI-powered intrusion detection," *Journal of Computer Science and Cybernetics*, vol. 41, no. 4, pp. 387–402, 2025, doi: 10.15625/1813-9663/22884.

[32] T. Ndayipfukamiye, J. Ding, D. S. Sarwatt, A. G. Philipo, and H. Ning, "Adversarial defense in cybersecurity: A systematic review of GANs for threat detection and mitigation," arXiv:2509.20411, 2025.

[33] S. Ennaji, F. De Gaspari, D. Hitaj, A. K/Bidi, and L. V. Mancini, "Adversarial challenges in network intrusion detection systems: Research insights and future prospects," arXiv:2409.18736, 2024.

[34] M. Mulimani, S. Allagi, and R. Rachh, "Preserving anonymity in cloud environment," in *Proc. IEEE Int. Conf. Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, Dec. 2018, pp. 371–375, doi: 10.1109/CTEMS.2018.8769219.

[35] M. Mulimani and R. Rachh, "A framework of machine learning-based intrusion detection system using classification and feature selection," *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 6720–6732, 2020.

[36] M. Mulimani, R. Rachh, and S. Allagi, "Attribute-based access control using machine learning techniques," *International Journal of Computer Applications*, submitted for publication.

[37] M. Mulimani, R. Rachh, and S. Allagi, "A hybrid machine learning-based approach for dynamic identification of network attacks in cloud environment," *International Journal of Engineering Trends and Technology*, vol. 70, no. 3, pp. 37–47, 2022, doi: 10.14445/22315381/IJETT-V70I3P205.

[38] M. Mulimani, R. Rachh, and S. Kavatagi, "A comparative approach: Machine learning and adversarial learning for intrusion detection," in *Proc. 3rd Int. Conf. Paradigms of Communication, Computing and Data Sciences (PCCDS 2022)*, Algorithms for Intelligent Systems. Singapore: Springer, 2023, doi: 10.1007/978-981-19-8742-7_39.

[39] M. Mulimani, R. Rachh, and S. Kavatagi, "CAT-RIDS: Cross adversarial training for robust intrusion detection system," *International Journal on Advanced Science, Engineering and Information Technology*, submitted for publication.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)