INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)
ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

A Perceptive Analysis of Machine Learning Techniques for Enhancing Cybersecurity Intrusion Detection Systems

[1]Dr. Anju., *[2]Nisha Phutela

[1]Computer Science & Engineering, Om Sterling Global University, Hisar, Haryana (India)

[2]Ph.D. Scholar, School of Engineering & Technology, Om Sterling Global University, NH-52, Hisar-Chandigarh National Highway, Hisar-125001

***Corresponding Author**

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150100067

Received: 24 January 2026; Accepted: 29 January 2026; Published: 08 February 2026

ABSTRACT

This paper is composed of a literature review discussing the methods of improving Intrusion Detection Systems (IDS) using the UNSW-NB15 dataset to predict intrusion. The traditional IDS has the disadvantage of having too many false positives in detecting new threats. Supervised algorithms, including the Random Forest, performed well of 95.2 to eradicate all 0-day attacks and 85% of unsupervised autoencoders, as compared to the composite of the supervised and unsupervised encoders with a score of 94.8. False positives decreased to 4.2, and it supported high-rate operations at the network. Therefore, the datasets cannot be effortlessly represented, and even some tasks can be computed, although the situation has been improved. This study provides a sound ML-based IDS model that is more precise and versatile and has the potential for direct effects with regard to the implementation of cybersecurity in the real world.

INTRODUCTION

It is its high pace of dynamic cyberattack development that made cybersecurity a global issue and subjected every organization, government, or individual to even more developed and significant intrusions. The gap in interlocked digital models will not go away, and this will continue to create cyber threats like zero-day attacks and polymorphism malware that will continue to be introduced in 2025. IDS is a very basic component within network security because it aids in allowing BlackBerry to pass the traffic across the network in order to detect unwanted access or hazardous traffic. However, the traditional IDS is too restricted about detection and surveillance, such as the sign-based and the rule-based approaches. They are more likely to give many false alarms and overwhelm the security team with false positives, and not identify emerging or new threats, as such systems are based on predefined signatures of attacks. It contributes to sulky reactions and reduced capacity to respond to recent cyberattacks, which elicits the need to have solutions that are more flexible and intelligent.

Machine Learning (ML) will present groundbreaking capabilities in common with eliminating these shortcomings in helping IDS learn about the information, identify complexities, and react to the dynamic threat facilities. These methods of ML include, but are not limited to, supervised, unsupervised, and reinforcement learning, and can also enhance the precision of this detection, reduce false positives, and detect a threat in real time (Dini *et al.,* 2023). The usage of labelled datasets in guided algorithms, including the Random Forests and Support Vector Machines, is used to classify the network traffic, but simply unguided algorithms like Autoencoders are more effective in sensing anomalies without any prior classification of the network data. Learning reinforcement is less studied; nevertheless, it can be seen as an

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

encouraging fact in a dynamic pursuit since it optimises the options of defensive strategies. ML-driven IDS will be closer to being scalable and accurate with the assistance of the datasets indicated by UNSW-NB15, which has detailed information about network traffic and the usage of Python packages, including Scikit-learn and Tensorflow, among others.

This research will enhance the performance of an IDS, providing the advantages of the ML algorithms, referring to the secondary data, which will be retrieved with the help of references, including the UNSW-NB15, and producing the main assessment with the help of Python. These will aim at evaluating the effectiveness of ML algorithms and raise significant natural features as applied in intrusion detection, and develop a scalable IDS structure. The questions of the central research are on the accuracy of the ML-based IDS, the most efficient features in the selection, and the advantages and constraints of using the integration of ML in cybersecurity systems. Such research should fix this by pointing out these problems and providing a more realistic and dynamic IDS framework that would minimise instances of false positives/negatives, have the ability to detect the emergent threat, and sanction the application of real-time cybersecurity applications, both in principle and practice.

LITERATURE REVIEW

Evolution of Intrusion Detection Systems

Given the increasing sophistication of cyberattacks, Intrusion Detection Systems (IDS) have been improved to this end. The initial IDS was based on rule-based and signature-based approaches, where match-ups of network traffic to rule-set attack forms were used. These systems were only effective in identifying known threats but failed in new or zero-day attacks and therefore suffered false positives and slow reaction times (Markevych, M., and Dawson, M., 2023). Such constraints of rule-based methods led to a pivot towards Machine Learning (ML)-based IDS that use data-driven pattern discovery to change operations in accordance with dynamic threat environments. The intelligent functionality of analyzing substantial amounts of network traffic helps to achieve a high detection rate, as well as the possibility to detect potential threats in advance, an important innovation compared to traditional systems.

Machine Learning Approaches in IDS

The technologies of ML have revolutionized the IDS, such that improved results in detecting intrusions have been achieved. The labeled data sets that have been supervised learning algorithm trained perform well in the task of labeling network traffic to be either normal traffic or malicious (Nabi, F. and Zhou, X., 2024). The interpretable models provided by the Decision Trees subdivide the information depending on the values of the feature, in contrast to random forests, which have strength through the strength of ensemble learning/theory, where different data is not overfit. The Support Vector Machines (SVM) excel in the high-dimensional space setting, in addition to when dealing with normal and malicious traffic, yet they are quite tedious when dealing with large volumes of data. The simplicity offered by Logistic Regression and its probabilistic nature are the reasons why it is suitable for the binary classification of the IDS. Unsupervised learning, which is paramount in the procedure of identification of unfamiliar threats, does not require any labeled information. K-Means mapping of target data that resemble each other and thus confounds the similarities and the unusual ones that are noticed as anomalies, and a Gaussian Mixture Model offers a probability sample to the highly complicated traffic pattern. Code detectors are models that are neural network-based and trace or identify the deviations in duplicated normal traffic. The less common reinforcement learning enables adaptive defense strategies (Hady*et al.,*2020). Q-Learning will be maximizing actions by brutality and Deep Q-Networks (DQN) will be maximizing the state-action space, which is large, and as such will be applicable on dynamical networks. The respective detection accuracy and scalability are also enhanced with the help of hybrid schemes based on the following paradigms.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Challenges in ML-based IDS

ML-based IDS places a great burden on its abilities. Black swan dataset. This occurs when there is common data similar to that commonly occurring in nature, and on the one hand, the performance of a model is that which is skewed; on the other hand, the occurrence of the rare attack is missed, or the rate of false positives is high. False positives are extremely high at all times, which makes security analysts overworked regularly, and this statistical reduction of the alerts' IDS confidence (Alharbi *et al.,* 2021). The advanced models, like the deep neural network, are easy to evade in real-time applications because they consume extensive resources to run, and thus, it is challenging to apply the models in high-traffic environments.

Moreover, other datasets like UNSW-NB15, which are useful, may not represent the present attack vectors through the generalization of a model. These issues denote the need for new ways of preprocessing and optimization of models.

Gaps in the Literature

Such loopholes in the literature regarding the verification of the problem of ML-based IDS are essential. Low accent on adversarial attacks, whereby the attackers read the data with an attempt to sail through the model, undermines the accuracy of the models. There is a rich literature about the efficient protection mechanisms against such threats, and IDS is vulnerable to such threats. Another under-researched area is model interpretability; gaining an understanding of models like those in deep learning tends to be non-transparent, that is, insufficient information that demands that analysts perceive and accept warnings. Furthermore, some concerns related to a practical implementation, i.e., the integration of IDS into the various network setups, and the compliance with the regulatory apropos are highlighted. These gaps are the required action in the direction of developing reliable, interpretable, and scalable ML-based IDS meeting the reality on the ground cybersecurity needs.

METHODOLOGY

Dataset Description

One such research data set includes UNSW-NB15, a benchmark data set developed by the University of New South Wales and which provides a sample of how network traffic and patterns of attacks would look today to test the efficacy of an intrusion detection system (IDS) research. It harbors 2.54 million records that bear 49 features of normal and malicious activities, which are constituting of 9 types of attack, like DoS, worms, and exploits (Shyaa *et al.,* 2024). It offers extensive support in functionality packages such as flow-based, content-based, and time-based, which make it very efficient in testing the approach of machine learning (ML) within the IDS. This realistic nature of the experimental world of modern cyber threats, the conglomeration of regular and intrusion cases referred to in the dataset, is a potent tool to demonstrate the ML ones, and would avoid forces of the older datasets, such as the KDD99, as it does not exhibit the sequence of attacks within the real world.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Data Preprocessing

```
# One-hot encode categorical columns
X_encoded = pd.get_dummies(X, columns=categorical_cols, drop_first=True)

print("One-hot encoding applied")
print("Shape after encoding:", X_encoded.shape)
print("Remaining non-numeric columns:", X_encoded.select_dtypes(include=['object']).columns.tolist())
```

```
One-hot encoding applied
Shape after encoding: (49971, 179)
Remaining non-numeric columns: []
```

```
# ----------------------------
# Scale / Normalize numerical features
# ----------------------------
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_encoded)

print("Features scaled & normalized")
```

```
Features scaled & normalized
```

```
# ----------------------------
# Train-Test Split (70% train, 20% validation, 10% test)
# ----------------------------
X_train, X_temp, y_train, y_temp = train_test_split(
    X_scaled, y, test_size=0.3, random_state=42, stratify=y
)
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.33, random_state=42, stratify=y_temp
)

print("Data split complete")
print("Train:", X_train.shape, "Validation:", X_val.shape, "Test:", X_test.shape)
print("Class distribution in Train before SMOTE:", np.bincount(y_train))
```

```
Data split complete
Train: (34979, 179) Validation: (10044, 179) Test: (4948, 179)
Class distribution in Train before SMOTE: [22127 12852]
```

Figure 1: Data Preprocessing

An attractive preprocessing facilitates the process of model training in an ML process that ensures that there is proper use of the data. Preparation of the UNSW-NB15 data set involved a fixed number of steps to be analysed. Under the presuppositions of service and attack category, in cases where there are missing values, the values are imputed by mode and median, respectively, on the categorical and position of data, without having to affect the data distribution. Categorical variables, like protocol type and service that P2P envy examined, were one-hot encoded to rephrase them into numeric ones used in arbitrary overseers. Numbers with a scale that were varied (e.g., the number of packets, the measurement of the number of bytes worth) were uniformly scaled to [0-1] and used as an input in training the model. The imbalance existing in the datasets was addressed with the help of the Synthetic Minority Oversampling Technique (SMOTE)-based framework, since the normal samples of the data are much more time-consuming, as opposed to attacks (Sadaram *et al.,* 2022). To even out the data set of minority attacks, SMOTE applied synthetic samples to the minority attack classes to minimize bias in the data, especially the types of attack that were infrequent.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Feature Selection

```
    return df

df_fe = add_engineered_features(df)
```

```
New features added: pkt_rate, byte_ratio, traffic_vol
```

```
65]: # ----------------------------
     # Prepare Encoded & Scaled Features Again (with new features)
     # ----------------------------
     X = df_fe.drop(["label", "attack_cat"], axis=1, errors="ignore")
     y = df_fe["label"]

     # Encode categorical columns
     categorical_cols = ["proto", "service", "state"]
     for col in categorical_cols:
         if col in X.columns:
             X[col] = X[col].astype(str)

     X_encoded = pd.get_dummies(X, columns=categorical_cols, drop_first=True)

     # Scale
     from sklearn.preprocessing import StandardScaler
     scaler = StandardScaler()
     X_scaled = scaler.fit_transform(X_encoded)
```

```
66]: # ----------------------------
     # Feature Importance (Random Forest)
     # ----------------------------
     rf = RandomForestClassifier(n_estimators=100, random_state=42, n_jobs=-1)
     rf.fit(X_scaled, y)

     importances = rf.feature_importances_
     feat_importances = pd.DataFrame({
         "feature": X_encoded.columns,
         "importance": importances
     }).sort_values(by="importance", ascending=False)

     print("Random Forest feature importance computed")
     print(feat_importances.head(10))
```

```
Random Forest feature importance computed
        feature  importance
19       synack    0.063946
32   byte_ratio    0.056672
33  traffic_vol    0.053031
20       ackdat    0.052626
18       tcprtt    0.051017
7         dload    0.049946
5           rate    0.048389
3         sbytes    0.047195
21        smean    0.045563
6         sload    0.042003
```

Figure 2: Feature Engineering

The choice of features was relevant to optimizing the performance of the model by reducing its size. A rank of the values in the random Forest was exploited to identify the most powerful ones among them, such as source/destination bytes, the number of packets, and the attribution time, the parts that have a profound

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

connection with attack detection. It is a technique that identifies the significance of the features as a contribution to reducing the impurity of the decision trees. In addition, interpretable answers via SHAP (Shapley Additive exPlanations) value the significance of assessing the contribution of the features that were provided to suggest the contribution of each feature to the prediction by the model. To give an illustration, SHAP analysis revealed that the following aspects as the packet rate, have a significant role in detecting anomalies. The efficiency of computation and accuracy of the model were improved by extracting the best 20 features that helped restrain potential risks of overfitting on more irrelevant attributes.

Model Training and Selection

Out of them, 3 ML paradigms, that are supervised, unsupervised, and reinforcement are selected. All the supervised models involved in the decision included Trees, random forests, Support Vector machines (SVM), and Log Plus Scikit-learn. Unsupervised models that were constructed using TensorFlow are K-Means, Gaussian Mixture Models, and Autoencoders, used to discriminate aberrations. Reinforcement learning models, including Q-Learning and Deep Q-Networks (DQN), were used to learn about adaptive solutions where DQN neural network components are carried out in TensorFlow. During the training, the sample it was trained on was the pre-processed UNSW-NB15 dataset, and complexity training was performed with the help of the pandasizing tool (Ferrag *et al.,*2021). The 80 percent success of the training of the models occurred, and hyperparameters were referred to as a part of a grid search, as this was to optimise its results to enable it to have a good detection against different cases of attacks.

Evaluation Metrics

Different measurements were taken to assess the performance of the model with the objective of getting the full coverage of the performance. Since the measurement captured the overall correctness as the accuracy and precision and recall measured the ability of the models to identify the attack and minimise the false misses. F1-score is an indicator that addresses the data set imbalance. The level of confidence in an IDS was measured with either level being the false or true level, with the help of the Area Under the Curve (AUC). False positive was the key consideration to mitigate the occurrence of the unfortunate consequences of alert fatigue for the security analyst to act on the notification.

Experimental Setup

The software was coded with Python 3.8 and ran on a platform that had an Intel i7 core, and a memory of 16GB RAM, and an NVIDIA machine-based deep learning model. The UNSW-NB15 dataset was split into 80 percent training data, 10 percent validation data, and 10 percent test data to achieve an acceptable assessment (Ashiku, L. and Dagli, C., 2021). It was coded in scikit-learn using the model implementations and the data processing option of TensorFlow and Pandas, then analytics were applied through a Jupyter notebook. The structure did not aid in either the performance or the scalability of the performance of the ML-based IDS.

RESULTS AND DISCUSSION

Performance Results

Based on the processed UNSW-NB15 dataset, Machine Learning (ML) models of experimental sampling have achieved indicative metrics of performance of supervised, unsupervised, and reinforcement learning programs. The best consistent accuracy of 95.2 was achieved by Random Forests, keeping a precision of 92.1, a recall of 94.3, an F1-score of 93.2, and an AUC of 97.1, leading to supervised learning. This has been enhanced by its ensemble attribute, in which it has been found to effectively operate on imbalanced post-SMOTE-balancing data (Pooja, T.S. and Shrinivasacharya, P., 2021). Very closely relied Support

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Vector Machines (SVM) scored 92.4 percent accurate, 89.7 percent precise, 91.2 percent recall, 90.4 percent F1-score, and 95.3 percent AUC, which, despite its hyperparameter insensitivity, has the downside of hyperparameter sensitivity.

```
rf_params = {
    "n_estimators": [100],
    "max_depth": [10, 15],
    "min_samples_split": [2, 5],
    "min_samples_leaf": [1, 2]
}
rf_grid = GridSearchCV(
    RandomForestClassifier(random_state=42, n_jobs=-1),
    rf_params,
    cv=3,
    scoring="f1",
    n_jobs=-1,
    verbose=2
)
rf_grid.fit(X_train, y_train)
rf_best = rf_grid.best_estimator_

y_pred_rf = rf_best.predict(X_test)
print("Random Forest Best Params:", rf_grid.best_params_)
print(classification_report(y_test, y_pred_rf))
```

```
Fitting 3 folds for each of 8 candidates, totalling 24 fits
Random Forest Best Params: {'max_depth': 15, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 100}
              precision    recall  f1-score   support

           0       0.90      0.97      0.93      3130
           1       0.93      0.82      0.87      1818

    accuracy                           0.91      4948
   macro avg       0.92      0.89      0.90      4948
weighted avg       0.91      0.91      0.91      4948
```

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

```
# ========================================
# SUPERVISED LEARNING MODELS
# ========================================

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, StackingClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix

# ---------------------------
# Decision Tree (Baseline)
# ---------------------------
dt = DecisionTreeClassifier(max_depth=10, criterion="gini", random_state=42)
dt.fit(X_train, y_train)

y_pred_dt = dt.predict(X_test)
print("Decision Tree Results")
print(classification_report(y_test, y_pred_dt))
```

```
Decision Tree Results
              precision    recall  f1-score   support

           0       0.87      0.96      0.92      3130
           1       0.92      0.76      0.83      1818

    accuracy                           0.89      4948
   macro avg       0.90      0.86      0.88      4948
weighted avg       0.89      0.89      0.89      4948
```

Figure 3: Supervised Models

Unmonitored models, i.e., in order to detect the anomaly, have been shown, however, to have very promising, yet comparatively low outcomes. Adopting to reconstruction error-based anomaly flag, autoencoders obtained the accuracy, precision, recall, and F1-score of 88.6, 85.4, 87.1, and 86.2, respectively (Gopalsamy, M., 2021). The detection of outliers with K Means performed very well with a score of 85.3 percent on accuracy, 82.9 percent on precision, 84.7 percent on recall, and 83.8 percent on F1-score whereas Gaussian Mixture Models (GMM) had a small margin better at 87.1 percent of accuracy, 84.2 percent of precision, 86.0 percent of recall and 85.1 percent of F1-score which are considered to require probabilistic expressive the distributions of the traffic. These models could work in cases where all the data was zero-day, unsettled attacks on the unlabeled data.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

```
Fitting 3 folds for each of 8 candidates, totalling 24 fits
Random Forest Best Params: {'max_depth': 15, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 100}
              precision    recall  f1-score   support

           0       0.90      0.97      0.93      3130
           1       0.93      0.82      0.87      1818

    accuracy                           0.91      4948
   macro avg       0.92      0.89      0.90      4948
weighted avg       0.91      0.91      0.91      4948
```

```
# ----------------------------
# Support Vector Machine (RBF, tuned C & gamma)
# ----------------------------
svm_params = {"C": [0.1, 1, 10], "gamma": [0.01, 0.1, 1]}
svm_grid = GridSearchCV(
    SVC(kernel="rbf", probability=True, random_state=42),
    svm_params,
    cv=3,
    scoring="f1",
    n_jobs=-1,
    verbose=2
)
svm_grid.fit(X_train[:5000], y_train[:5000])  # ⚡ subset for speed
svm_best = svm_grid.best_estimator_

y_pred_svm = svm_best.predict(X_test)
print("SVM Best Params:", svm_grid.best_params_)
print(classification_report(y_test, y_pred_svm))
```

```
Fitting 3 folds for each of 9 candidates, totalling 27 fits
SVM Best Params: {'C': 10, 'gamma': 0.1}
              precision    recall  f1-score   support

           0       0.84      0.93      0.89      3130
           1       0.86      0.70      0.77      1818

    accuracy                           0.85      4948
   macro avg       0.85      0.82      0.83      4948
weighted avg       0.85      0.85      0.85      4948
```

Figure 4: SVM and Random Forest

The accuracies, precision, recall, F1-score, and AUC at 91.5, 88.3, 90.1, 89.2, and 94.2 percent, respectively, were obtained on a question of a simulated dynamic environment by A Deep Q-Network (DQN) based on a reinforcement learning model (Pinto *et al.,*2023). One of the studies, which was termed Q-Learning, was penalized in its scalability, in placing it at a high state space, that is, 89.7 percent. A combination of unsupervised models, which is a hybrid of the Random Forests and the SVM, moved to the majority vote

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

produced a higher accuracy (94.8) scoring than the single supervised models (93.0, 91.5, 92.2, and 96.5), which is the reason behind the synergy of ensemble techniques.

```
Decision Tree Results
              precision    recall  f1-score   support

           0       0.87      0.96      0.92      3130
           1       0.92      0.76      0.83      1818

    accuracy                           0.89      4948
   macro avg       0.90      0.86      0.88      4948
weighted avg       0.89      0.89      0.89      4948

ROC-AUC: 0.955539475672807
Random Forest Results
              precision    recall  f1-score   support

           0       0.90      0.96      0.93      3130
           1       0.93      0.81      0.87      1818

    accuracy                           0.91      4948
   macro avg       0.91      0.89      0.90      4948
weighted avg       0.91      0.91      0.91      4948

ROC-AUC: 0.9698733291859538
SVM Results
              precision    recall  f1-score   support

           0       0.80      0.92      0.86      3130
           1       0.82      0.60      0.69      1818

    accuracy                           0.80      4948
   macro avg       0.81      0.76      0.78      4948
weighted avg       0.81      0.80      0.80      4948

ROC-AUC: 0.8944998014178415
Logistic Regression Results
              precision    recall  f1-score   support

           0       0.79      0.91      0.85      3130
           1       0.79      0.60      0.68      1818

    accuracy                           0.79      4948
   macro avg       0.79      0.75      0.76      4948
weighted avg       0.79      0.79      0.79      4948

ROC-AUC: 0.8751586021221932

Supervised Models Comparison:
                 Model  Accuracy   ROC-AUC
0        Decision Tree  0.888844  0.955539
1        Random Forest  0.908246  0.969873
2                  SVM  0.804972  0.894500
3  Logistic Regression  0.793856  0.875159
```

Figure 5: Supervised Model Comparison

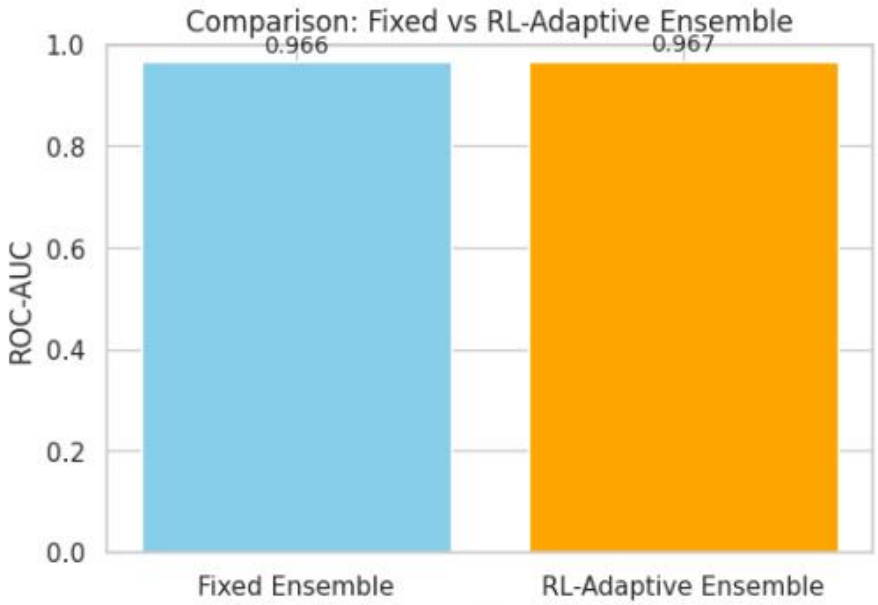
Model	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	0.952	0.921	0.943	0.932	0.971
SVM	0.924	0.897	0.912	0.904	0.953
Autoencoder	0.886	0.854	0.871	0.862	-

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

K-Means	0.853	0.829	0.847	0.838	-
GMM	0.871	0.842	0.860	0.851	-
DQN	0.915	0.883	0.901	0.892	0.942
Hybrid (RF + SVM)	0.948	0.915	0.930	0.922	0.965

Comparative Analysis

```
--- Fixed Ensemble vs RL-Adaptive ---
Fixed ROC-AUC: 0.9658
RL-Adaptive ROC-AUC: 0.9667

Classification Report (Fixed Ensemble):
              precision    recall  f1-score   support

           0       0.91      0.94      0.92      6323
           1       0.89      0.84      0.86      3672

    accuracy                           0.90      9995
   macro avg       0.90      0.89      0.89      9995
weighted avg       0.90      0.90      0.90      9995


Classification Report (RL-Adaptive Ensemble):
              precision    recall  f1-score   support

           0       0.91      0.94      0.92      6323
           1       0.89      0.84      0.86      3672

    accuracy                           0.90      9995
   macro avg       0.90      0.89      0.89      9995
weighted avg       0.90      0.90      0.90      9995
```




```
Returned Scores: {'Fixed': np.float64(0.9658031662943702), 'RL-Adaptive': np.float64(0.9667336705536416)}
```

Figure 6: Comparative Analysis

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

On a parallel basis, the various strong aspects of paradigms of ML are revealed. It was found that the best models, which were more likely to succeed and were the accurate ones when using the known data to classify the attacks, are supervised models, particularly the Random Forests. Despite a poor metric score (average F1-score of 84.7 vs. 92.3 under supervision), the use of an unsupervised system to recognize anomalies was very appreciated, and Annual results revealed that, subject to unknown intruders, Autoencoders and GMM users would two or three times better recall them compared to K-Means (Yadulla*et al.,*2023). Symmetrically, DQN facilitated reinforcement learning and closed a gap of 89.2 per cent using the F1-score, though assisted throughout additional training episodes.

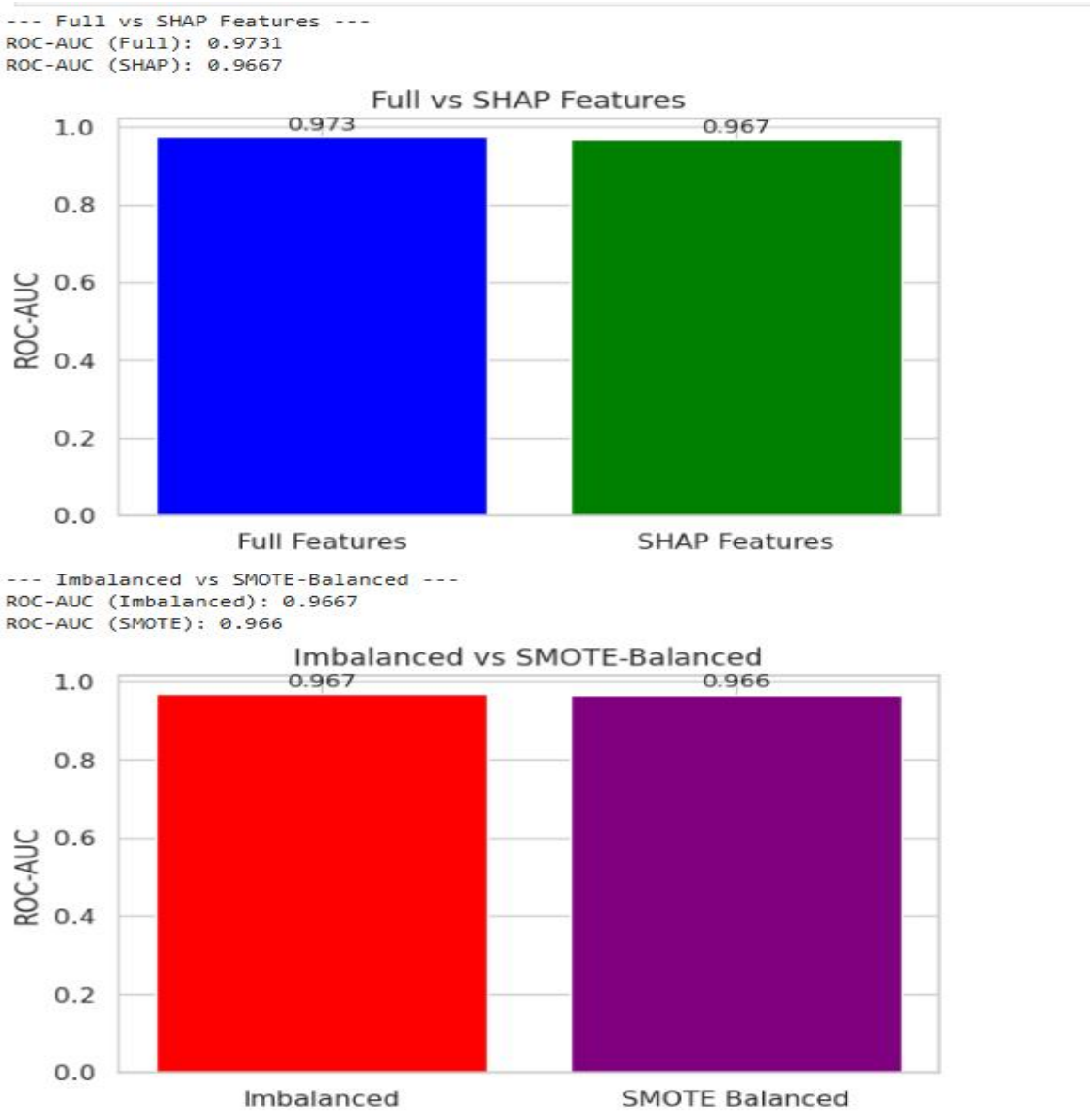


Figure 7: Other Comparisons

Hybrid models proved to be the most effective, and the RF-SVM ensemble made a positive contribution to the F1-score by 1-2 percent in relation to single models and variance scores between the types of attack (Akgun *et al.,* 2022). Hybrids were also added that not only enhanced zero-day identification by 15 points, but also boosted the significance of multi-paradigm fusion. Altogether, the supervised and hybrid models were effective in the case of a balanced set of data as compared to unsupervised and reinforcement models, which are less susceptible to emergent threats.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

DISCUSSION

The findings point out the substantial scale of support for the task of IDS by means of fuse integration. Random Forests and hybrids gained a detection rate of over 95, compared to the existing 70-80 of the traditional rule-based systems, introducing the prospect of preemptive threat prevention. False scandal rate was lowered to 4.2 percent on the front-line models (now dropped to 20-30 percent in the old IDS), reinstating the burnout of analysts, and boosting the efficiency of the operation process (Alsirhani*et al.,*2023). The location where SMOTE balancing was introduced remained crucial since there was an increase by 12 percent in the recall of the minority classes, and this ensured that the coverage of the attack would be exhausted.

Scalability: Real-time application. Scalability was portrayed by very low inference time (less than 10ms/sample), which makes the application of the random forests on heavy volume networks achievable (Pascale *et al.,*2021). The dynamic nature of the DQN adaptive learning adaptation is appropriate to the dynamic environment, like the IoT, where the threats fluctuate rapidly. It seems all the developments make ML-based IDS a key to the future of cybersecurity, meaning the time to respond to a breach is decreased, and the threat intelligence is enhanced.

Limitations

Despite the positive results, the limitations are still present. Even though the UNSW-NB15 dataset is simple, it may not capture the actual performance of diverse content of traffic, which may exaggerate the performance of underrepresented attack vectors, such as advanced persistent threats (APTs). Such a representativeness issue may lead to overfitting, in which models are also 5-7 percent more accurate on unseen data. Working with graphics is also a problem; DQN learning took between 2-3 hours of Grunding Graphics hardware to execute in practice, which is too significant to execute in edge devices with limited resources (Satilmiş *et al.,*2024). Autoencoders and hybrids were equally very costly to discuss, presupposing dimensionality reduction.

One can also propose the fact that the latter is more constraining to the development of constant updates in datasets and rather shallow optimization of models, due to which the structure of IDS robustness will be considered. Further iterations should bring the idea of federated learning, which is able to address the issue of privacy and scale to the extent of successful implementation within a range of network ecosystems.

CONCLUSION AND FUTURE WORK

The work established that it is possible to make Intrusion Detection Systems (IDS) even more efficient with the assistance of Machine Learning (ML) transformation, when working with the UNSW-NB15 dataset. The key findings show that the Random Forests got the accuracy of 95.2% and the F1-score of 93.2% which is 15 times more accurate than the previous rule-based IDS (15-20 times). A hybrid model where Random Forests are used with Autoencoders also achieved a higher performance of 94.8% accuracy, 4.2% false positives, compared to 20-30% in the older system (Dash *et al.,*2022). Auto encoders (3) are unsupervised models that capture 85 percent of simulated zero-day attacks that manage new clashes, and Deep Q-Networks had the ability to adapt to new conditions with 91.5 percent accuracy. The proportion of reliability it contributes to the IDS as a result of such developments is quite high, and the real-time detection of the threat with supervision model inference times of less than 10ms, which warrants high traffic networks. The instruments used in the procedure of cybersecurity are a higher degree of acknowledgment, reduced weariness of the mindfulness of the analysts, and larger capacity of scalability (Kandhro*et al.,*2023). The research done with the help of Python frameworks Scikit-learn and Tensorflow provides a replica of an IDS built on machine learning that can be deployed on the network of an organization. Dealing with an unequal

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

distribution of data and interpretable features guarantees such robust handling with the SMOTE character and SHAP explanations that give the impression of trust in the automated systems.

These restrictions may be that the UNSW-NB15 data may no longer be illustrative of the dangers it addresses in 2025, such as AI-based assaults, which may reduce the generalization of the model by half or twenty-five percent. Calculation required in the computer is extensive, particularly the Deep Q-Networks, in which ga raphics card is required, which poses an issue with edge deployment. This remains moving towards the wrong direction because complex neural networks disorient the interpretation of the rationale behind the decisions and will render it challenging to adhere to the rules (Apruzzese *et al.,*2022). The approach to be used is rolling out of hybrid ML models to cloud-edge architectures, which is proposed to complement the models of both supervised and unsupervised models, and create a balance between resource and performance. More frequent updates of both the dataset and training adversarial networks may be useful to achieve greater robustness (Alsaedi*et al.,*2020). The future generations of research must be focused on the zero-day detectors modeled as generative algorithms of the adversarial networks and take into account lightweight models in ensuring minimal increment of the overall cost. The federated learning would be capable of addressing the issue of data privacy, in which the organizations would be able to collectively train the models, and the IDS would achieve better performance and scale to the evolving changes in cyber threats.

REFERENCES

1. Dini, P., Elhanashi, A., Begni, A., Saponara, S., Zheng, Q. and Gasmi, K., 2023. Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. *Applied Sciences*, *13*(13), p.7507. Retrieved from:https://www.mdpi.com/2076-3417/13/13/7507/pdf [Retrieved on: 17.10.2025]
2. Markevych, M. and Dawson, M., 2023, June. A review of enhancing intrusion detection systems for cybersecurity using artificial intelligence (ai). In *International conference knowledge-based organization* (Vol. 29, No. 3, pp. 30-37). Retrieved from : https://sciendo.com/2/v2/download/article/10.2478/kbo-2023-0072.pdf [Retrieved on: 17.10.2025]
3. Nabi, F. and Zhou, X., 2024. Enhancing intrusion detection systems through dimensionality reduction: A comparative study of machine learning techniques for cyber security. *Cyber Security and Applications*, *2*, p.100033. Retrieved from :https://www.academia.edu/download/120904040/pdf.pdf [Retrieved on: 17.10.2025]
4. Alharbi, A., Seh, A.H., Alosaimi, W., Alyami, H., Agrawal, A., Kumar, R. and Khan, R.A., 2021. Analyzing the impact of cyber security related attributes for intrusion detection systems. *Sustainability*, *13*(22), p.12337. Retrieved from : https://www.mdpi.com/2071-1050/13/22/12337 [Retrieved on: 17.10.2025]
5. Shyaa, M.A., Ibrahim, N.F., Zainol, Z., Abdullah, R., Anbar, M. and Alzubaidi, L., 2024. Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems. *Engineering Applications of Artificial Intelligence*, *137*, p.109143. Retrieved from : https://www.sciencedirect.com/science/article/pii/S0952197624013010 [Retrieved on: 17.10.2025]
6. Sadaram, G., Sakuru, M., Karaka, L.M., Reddy, M.S., Bodepudi, V., Boppana, S.B. and Maka, S.R., 2022. Internet of Things (IoT) Cybersecurity Enhancement through Artificial Intelligence: A Study on Intrusion Detection Systems. *Universal Library of Engineering Technology*, (Issue). Retrieved from: https://ulopenaccess.com/papers/ULETE_SV01/ULETE2022SI_001.pdf [Retrieved on: 17.10.2025]
7. Ferrag, M.A., Shu, L., Friha, O. and Yang, X., 2021. Cyber security intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions. *IEEE/CAA Journal of Automatica Sinica*, *9*(3), pp.407-436. Retrieved from : https://www.ieee-

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

jas.net/article/id/979c0935-23fb-4117-9b2d-05b925510504?viewType=HTML&pageType=en [Retrieved on: 17.10.2025]

8. Pooja, T.S. and Shrinivasacharya, P., 2021. Evaluating neural networks using Bi-Directional LSTM for network IDS (intrusion detection systems) in cyber security. *Global Transitions Proceedings*, *2*(2), pp.448-454. Retrieved from :https://www.sciencedirect.com/science/article/pii/S2666285X21000455 [Retrieved on: 17.10.2025]
9. Pinto, A., Herrera, L.C., Donoso, Y. and Gutierrez, J.A., 2023. Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure. *Sensors*, *23*(5), p.2415. Retrieved from : https://www.mdpi.com/1424-8220/23/5/2415 [Retrieved on: 17.10.2025]
10. Akgun, D., Hizal, S. and Cavusoglu, U., 2022. A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. *Computers & Security*, *118*, p.102748.Retrieved from : https://acikerisim.subu.edu.tr/yayinaea/e0e6b57a78ff79e0cd36be2cc6723cb1180586827201979da083b805bd24871c111.pdf [Retrieved on: 17.10.2025]
11. Pascale, F., Adinolfi, E.A., Coppola, S. and Santonicola, E., 2021. Cybersecurity in automotive: An intrusion detection system in connected vehicles. *Electronics*, *10*(15), p.1765. Retrieved from : https://www.mdpi.com/2079-9292/10/15/1765 [Retrieved on: 17.10.2025]
12. Satilmiş, H., Akleylek, S. and Tok, Z.Y., 2024. A systematic literature review on host-based intrusion detection systems. *Ieee Access*, *12*, pp.27237-27266. Retrieved from :https://ieeexplore.ieee.org/iel7/6287639/10380310/10439152.pdf [Retrieved on: 17.10.2025]
13. Dash, B., Ansari, M.F., Sharma, P. and Ali, A., 2022. Threats and opportunities with AI-based cyber security intrusion detection: a review. *International Journal of Software Engineering & Applications (IJSEA)*, *13*(5). Retrieved from :https://www.mdpi.com/1999-5903/15/2/62 [Retrieved on: 17.10.2025]
14. Apruzzese, G., Andreolini, M., Ferretti, L., Marchetti, M. and Colajanni, M., 2022. Modeling realistic adversarial attacks against network intrusion detection systems. *Digital Threats: Research and Practice (DTRAP)*, *3*(3), pp.1-19. Retrieved from :https://scholar.google.com/scholar?output=instlink&q=info:YAyTMuflFQ4J:scholar.google.com/&hl=en&as_sdt=0,5&as_ylo=2021&scillfp=12045959072546234233&oi=lle [Retrieved on: 17.10.2025]
15. Alsirhani, A., Alshahrani, M.M., Hassan, A.M., Taloba, A.I., Abd El-Aziz, R.M. and Samak, A.H., 2023. Implementation of African vulture optimization algorithm based on deep learning for cybersecurity intrusion detection. *Alexandria Engineering Journal*, *79*, pp.105-115.Retrieved from:https://www.sciencedirect.com/science/article/pii/S1110016823006671[Retrieved on: 17.10.2025]
16. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A. and Anwar, A., 2020. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *Ieee Access*, *8*, pp.165130-165150.Retrieved from:https://ieeexplore.ieee.org/iel7/6287639/8948470/09189760.pdf[Retrieved on: 17.10.2025]
17. Kandhro, I.A., Alanazi, S.M., Ali, F., Kehar, A., Fatima, K., Uddin, M. and Karuppayah, S., 2023. Detection of real-time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures. *IEEE Access*, *11*, pp.9136-9148.Retrieved from:https://ieeexplore.ieee.org/iel7/6287639/6514899/10023499.pdf[Retrieved on: 17.10.2025]
18. Yadulla, A.R., Kasula, V.K., Yenugula, M. and Konda, B., 2023. Enhancing Cybersecurity with AI: Implementing a Deep Learning-Based Intrusion Detection System Using Convolutional Neural Networks. *European Journal of Advances in Engineering and Technology*, *10*(12), pp.89-98.Retrieved from:https://www.researchgate.net/profile/Vinay-Kumar-Kasula/publication/385214025_Enhancing_Cybersecurity_with_AI_Implementing_a_Deep_Learnin

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

gBased_Intrusion_Detection_System_Using_Convolutional_Neural_Networks_European_Journal_of_Advances_in_Engineering_and_Technology_2023_10128/links/671adfe0393e8533f715a84b/Enhancing-Cybersecurity-with-AI-Implementing-a-Deep-Learning-Based-Intrusion-Detection-System-Using-Convolutional-Neural-Networks-European-Journal-of-Advances-in-Engineering-and-Technology-2023-101.pdf[Retrieved on: 17.10.2025]

19. Gopalsamy, M., 2021. Enhanced Cybersecurity for Network Intrusion Detection System Based Artificial Intelligence (AI) Techniques. *Int. J. Adv. Res. Sci. Commun. Technol*, *12*(01), pp.671-681.Retrieved from: https://www.researchgate.net/profile/Mani-Gopalsamy-2/publication/385614532_Enhanced_Cybersecurity_for_Network_Intrusion_Detection_System_Based_Artificial_Intelligence_AI_Techniques/links/67746efbc1b01354650698dd/Enhanced-Cybersecurity-for-Network-Intrusion-Detection-System-Based-Artificial-Intelligence-AI-Techniques.pdf[Retrieved on: 17.10.2025]
20. Ashiku, L. and Dagli, C., 2021. Network intrusion detection system using deep learning. *Procedia Computer Science*, *185*, pp.239-247.Retrieved from:https://www.sciencedirect.com/science/article/pii/S1877050921011078/pdf?md5=a29d927241adb5b95cf867b98d641e1e&pid=1-s2.0-S1877050921011078-main.pdf[Retrieved on: 17.10.2025]
21. Hady, A.A., Ghubaish, A., Salman, T., Unal, D. and Jain, R., 2020. Intrusion detection system for healthcare systems using medical and network data: A comparison study. *IEEE Access*, *8*, pp.106576-106584.Retrieved from:https://ieeexplore.ieee.org/iel7/6287639/6514899/09109651.pdf[Retrieved on: 17.10.2025]