

A Comparative Evaluation of Standalone and Hybrid Machine Learning Techniques for Network Anomaly Detection Using NSL-KDD

Nuri Nazirah Binti Ismail
Faculty of Computer Science and
Information Technology,
Universiti Malaysia Sarawak,
94300 Kota Samarahan,
Sarawak, Malaysia
85272@siswa.unimas.my

Lau Sei Ping
Institute of Social Informatics and
Technological Innovations,
Universiti Malaysia Sarawak,
94300 Kota Samarahan,
Sarawak, Malaysia
splau@unimas.my

Tan Chong Eng
Institute of Social Informatics and
Technological Innovations,
Universiti Malaysia Sarawak,
94300 Kota Samarahan,
Sarawak, Malaysia
cetan@unimas.my

Abstract—Results of network anomaly detection in a controlled environment can be persuasive but often fail to perform with new, unseen network traffic. This study investigates four standalone machine learning models, namely Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), Neural Network (NN), and four hybrid models, which include SVM + RF, K-Means + RF, NN + RF, and K-Means + NN. These models are studied using both stratified 10-fold validation and KDDTest+ subset evaluation. This study investigates not only which model performs best but also assesses whether a hybrid model design provides a measurable advantage over standalone models. An identical processing procedure was applied to all models. The procedure includes one-hot encoding, normalization, and stratified 10-fold cross-validation, followed by final evaluation with the KDDTest+ subset. SVM + RF hybrid reported the highest accuracy of 84.08% and precision of 0.9716. In contrast, the standalone Neural Network outperformed the other models with a recall score of 0.7522 and an F1-score of 0.8425, which indicates better detection coverage performance. Relative to the standalone SVM baseline, SVM + RF achieved relative improvements of 1.01% in accuracy and 2.13% in precision. Standalone NN demonstrated a precision-recall trade-off with 1.36% higher recall over the top hybrid model. Overall, the findings show that hybrid models can perform better for the selected metrics, but do not guarantee consistent superiority. For deployment, model choice should be guided by the operational priority of the intrusion detection system, particularly the trade-off between reducing false alarms and detecting more attacks.

Keywords—artificial intelligence, machine learning, intrusion detection system, network anomaly detection, NSL-KDD

I. INTRODUCTION

As cybercrime and network-based attacks become more widespread and sophisticated, intrusion detection systems (IDSs) remain crucial for securing digital infrastructure. Anomaly detection improves network security by recognizing traffic behaviours that deviate from normal patterns, which may indicate malicious activity. While traditional signature-based IDSs are effective against known attacks, they are less capable when attack patterns change or are previously unseen [1].

Machine learning-based IDSs have therefore become prominent due to their ability to learn from labeled traffic data to identify attack patterns. Nevertheless, it is essential to evaluate the performance of these models, as high precision may not indicate improved performance on imbalanced security

datasets. This is because, in many situations, a classifier can identify a vast majority of normal data records, while it is unable to recognize the majority of attack data. Thus, a more holistic evaluation should be based on precision, recall, F1-score, and an interpretation of the confusion matrix.

Currently, many studies employ machine learning classifiers in IDSs, such as Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), and Neural Network (NN), while some have considered hybrid variants that combine clustering, feature generation, or ensemble classification. By training these models on labelled datasets, IDSs can recognize normal and anomalous traffic vectors and support automated cybersecurity decision-making.

This study uses the NSL-KDD dataset to benchmark standalone machine learning models and their hybrid variants. Compared with the original KDD Cup 99 dataset, NSL-KDD reduces duplicate records in the KDDTrain+ and KDDTest+ subsets, limiting frequency-based bias and forcing models to learn behavioral patterns rather than duplicated samples. Since network traffic classes are not evenly distributed, this paper reports multiple performance metrics rather than relying solely on accuracy.

The objective of this study is to conduct a standardized comparative evaluation of standalone and hybrid machine learning architectures for network anomaly detection. The main contributions are: (i) comparing standalone and hybrid models under an identical preprocessing procedure; (ii) reporting accuracy, precision, recall, F1-score, and confusion-matrix-based interpretation; (iii) highlighting the validation-to-test generalization gap on NSL-KDD; and (iv) identifying the practical trade-off between false-alarm suppression and attack detection sensitivity for IDS deployment.

II. RELATED WORK

Network anomaly detection is commonly divided into signature-based and anomaly-based detection. Signature-based methods rely on predefined attack patterns and can achieve low false positives for known threats, but they are limited against zero-day and evolving attacks [2]. Anomaly-based detection constructs a profile of normal behaviour and flags deviations as suspicious. This approach is more flexible for unknown attacks, but it may suffer from higher false-positive rates and greater modelling complexity [3].

Liu et al. [4] proposed a parallelized two-stage intrusion detection framework using Spark Resilient Distributed Datasets (RDDs). In the first phase, K-Means grouped preprocessed NSL-KDD and CIC-IDS2017 traffic into eight clusters, where small clusters were treated as anomalies and larger clusters were passed to the next stage. In the second phase, a Random Forest model performed binary classification. The hybrid approach achieved 92.9% accuracy on NSL-KDD and 99.87% accuracy on CIC-IDS2017, showing the potential of combining clustering with supervised classification.

Halbouni et al. [5] proposed a hybrid intrusion detection framework that sequentially integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) across three repeated stages to extract spatial and temporal traffic features. The CNN layers extracted critical traffic features, while the LSTM layers captured sequential dependencies. Their model achieved high detection accuracy on CIC-IDS2017, UNSW-NB15, and WSN-DS datasets and demonstrated the value of deep hybrid learning for intrusion detection.

Alsajri and Steiti [6] evaluated a hybrid intrusion detection framework combining a Support Vector Machine (SVM) with a Genetic Algorithm (GA) against a standalone SVM baseline. In this framework, GA was used to select informative feature subsets, while SVM performed classification in a high-dimensional feature space. Their findings showed that GA-SVM improved the true positive rate and reduced the false negative rate compared with the standalone SVM.

Prior studies show that hybrid IDS models can improve detection performance, but the extent of the improvement depends on how the hybrid components interact with the dataset structure. While clustering, feature selection, and ensemble learning have the potential to enhance model robustness, they may also add noise or computational complexity to the models. This can happen when the learned patterns or features produced during the intermediate processing phase fail to distinguish normal traffic from attack traffic. This paper examines this issue by benchmarking multiple standalone and hybrid models on NSL-KDD dataset using the same preprocessing procedures and metrics. The aim of the evaluation is to identify the best-performing model and to determine whether hybridization provides a meaningful advantage over simpler standalone models in terms of false-alarm reduction and attack-detection sensitivity.

III. METHODOLOGY

This section describes the experimental design used to benchmark the selected machine learning models for network anomaly detection. The experiments were designed with Python and Jupyter Notebooks in the Kaggle environment for fast, reproducible data preprocessing and iterative steps for model training, validation, and testing.

The models were benchmarked using an offline dataset; therefore, the results represent comparative offline detection performance rather than real-time operational IDS performance. The study assumes that the NSL-KDD dataset is appropriately labelled and the dataset remains a relevant resource for benchmarking anomaly detection models. Fig.1 illustrates the

experimental pipeline used in this study. After the dataset is collected and processed, either a standalone technique or a hybrid learning approach is selected. Depending on the selected model, the training stage may include feature extraction or selection, clustering, classification, and other model-specific procedures. The models are subsequently tuned, tested, evaluated, and analysed.

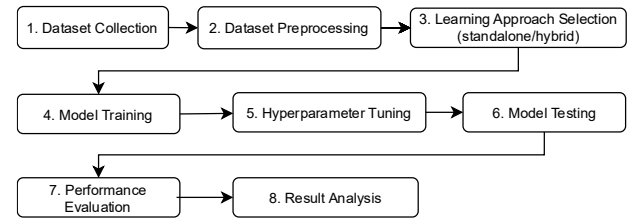


Fig. 1. Experimental workflow for evaluating standalone and hybrid models

A. Dataset Description

This study used the NSL-KDD dataset for performance benchmarking. This dataset has been widely used in intrusion detection research because it includes both normal and abnormal traffic, making it suitable for training and testing machine learning models. Two NSL-KDD subsets, namely KDDTrain+ and KDDTest+ were used for training and testing sessions, respectively. The NSL-KDD dataset contained 41 features that are further grouped into four categories: basic features, time-based traffic features, connection-based traffic features, and content features. Each record in the dataset consists of multiple features, including protocol type, service, and flag, which are essential for anomaly detection.

B. Data Preprocessing

Before training, the dataset underwent several preprocessing steps. This process ensures that the selected models receive clean and consistent input. This also helps to reduce noise and improve the classification accuracy of the selected models. Since the raw NSL-KDD dataset was in a header-less CSV format, it lacked built-in column titles. A predefined list of 43 attribute names was assigned to the dataset during loading. This ensures that each of the 41 technical features, the class label, and the difficulty score is correctly identified for preprocessing and analysis. To ensure the model relies on real-world network traffic features rather than artificial complexity cues, the difficulty score was removed during data cleaning.

To optimize performance, all of these various multi-class threat categories were consolidated and converted into a single unified attack class, while the normal records were retained as the normal class. Through binary label mapping, harmless network traffic is explicitly assigned a value of 0, while any malicious activity or systemic breach is assigned a value of 1. The NSL-KDD dataset contains three categorical features, namely *protocol_type*, *service*, and *flag*. To resolve this, One-Hot Encoding was applied to transform these qualitative labels into a binary format that a computer can read. This process involves creating a new binary column for each unique category within a feature, where a value of 1 (true) represents the presence of the attribute and 0 (false) represents its absence. To handle unseen network behaviours and different traffic distributions in the testing dataset (KDDTest+) compared to the

training dataset (KDDTrain+), standard independent encoding would normally produce a different number of columns for each file. Therefore, an explicit column alignment step was executed immediately after encoding to identify missing columns between the splits, fill them with zeros, and reorder the tables to ensure an identical structural shape that prevents critical feature mismatch errors during model testing.

Such a large disparity and scale in some features can cause certain algorithms to become biased toward features with higher magnitudes and completely ignore the lower values. To achieve a uniform data distribution, Min-Max scaling was implemented to normalize all numerical features into a standardized range between 0 and 1. By fitting the data into this specific range, the model is forced to treat each feature with equal importance during the training phase to ensure fairness.

To optimize model performance and achieve an optimal trade-off between bias and variance, a stratified 10-fold cross-validation strategy was implemented during training and validation. Under this framework, the KDDTrain+ is partitioned into 10 equal-sized folds, with each fold serving as the validation set while the remaining K-1 folds train the model. This validation mechanism ensures that every sample is utilized for both training and evaluation, enabling the models to learn all available attack patterns and effectively generalize to unseen network anomalies in the final testing phase. For the final evaluation, the trained models are tested exclusively using the KDDTest+ dataset. Isolating this testing dataset preserves evaluation integrity, ensuring unbiased assessment metrics when validating the performance of selected models against unseen malicious traffic.

C. Machine Learning Models

This study evaluates the performance of two categories of machine learning models, i.e., standalone and hybrid. The standalone models include Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), and Neural Network (NN). The hybrid models used a combination of SVM, K-Means, RF, and NN to evaluate whether model integration can improve anomaly detection capability. A brief description of each model is provided below:

- **Random Forest (RF):** RF is selected as a baseline supervised learning model because of its ability to handle high-dimensional and complex network traffic data. It builds multiple decision trees trained using bootstrap samples and randomly selected features. The final predictions are determined through majority voting, which reduces overfitting and improves data classification stability in noisy intrusion detection tasks.
- **Decision Tree (DT):** DT provides a simple and interpretable baseline by recursively dividing the feature space using criteria such as Gini impurity or entropy. Although DT is suitable for initial benchmarking, single-tree model is generally more prone to overfitting and less stable than ensemble-based models.
- **Support Vector Machine (SVM):** SVM performs binary classification by identifying a separating hyperplane between normal and anomalous traffic. Kernel functions can capture non-linear relationships in the NSL-KDD

dataset, making SVM a strong baseline for high-dimensional traffic features.

- **Neural Network (NN):** NN uses a feedforward Multilayer Perceptron (MLP) to capture complex non-linear relationships. The network consists of three hidden layers with 128, 64, and 32 neurons, which were activated with ReLU. It was trained using Adam optimizer and binary cross-entropy served as the loss function. This configuration provides a deep-learning baseline before hybrid integration.
- **SVM + RF:** The SVM first generates probability scores representing the likelihood of anomalous traffic. These scores are added as an additional feature and used as input to the RF classifier for final prediction.
- **K-Means + RF:** K-Means first groups network traffic according to their behavioral similarities. The resulting cluster assignments are added to the original feature set before RF performs the final classification.
- **NN + RF:** This hybrid was implemented using a soft-voting ensemble between a Random Forest classifier and a Multilayer Perceptron neural network. Both models generated predicted class probabilities, and the final prediction was determined from the highest combined probability.
- **K-Means + NN:** K-Means is first used to discover hidden groups in the dataset through clustering. The cluster assignments are then combined with the original features and supplied to the neural network, enabling the classifier to exploit both clustering information and learned feature representations.

D. Training and Testing Procedure

All models were trained using the same NSL-KDD preprocessing procedure, including binary label mapping, one-hot encoding, feature alignment, and Min-Max normalization. The KDDTrain+ set was used for stratified 10-fold validation and manual hyperparameter tuning. To ensure optimal detection performance and experimental consistency, model hyperparameters were selected through preliminary empirical tuning. Key structural parameters such as tree depth, number of estimators, hidden layer architecture, kernel functions, and cluster counts were systematically tested on the training validation splits. The hyperparameters for each model were then configured across 5 experimental iterations of the preliminary tuning, with the final configuration selected based on the highest performing evaluation metrics. The final model for each configuration was then evaluated on the independent KDDTest+ set without further tuning. This procedure ensures that the reported results reflect performance on unseen test data rather than cross-validation performance alone.

E. Evaluation Metrics

A confusion matrix was used to evaluate classification performance by comparing predicted labels with actual labels. For binary anomaly detection, the four outcomes are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The following metrics were derived from these values:

- Accuracy measures the proportion of correctly classified records:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

- Precision measures how many predicted attack records are actual attacks:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- Recall measures how many actual attack records are correctly detected:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- F1-score is the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Together, these metrics offer a holistic view of model performance on detection capability and error tendencies. In intrusion detection systems, they are essential for assessing how well a classifier distinguishes between normal and anomalous traffic.

IV. RESULTS AND DISCUSSION

A. Overall Performance Comparison

The final performance of the selected standalone and hybrid models that benchmarked with the KDDTest+ dataset are shown in Table I.

TABLE I. FINAL TEST PERFORMANCE OF STANDALONE AND HYBRID MODELS WITH KDDTEST+.

Model	Accuracy (%)	Precision	Recall	F1
RF	80.93	0.9701	0.6861	0.8038
DT	81.64	0.9667	0.7016	0.8131
SVM	83.24	0.9513	0.7436	0.8347
NN	83.99	0.9574	0.7522	0.8425
SVM + RF	84.08	0.9716	0.7421	0.8415
K-Means + RF	80.24	0.9692	0.6743	0.7953
NN + RF	80.31	0.9283	0.7087	0.8038
K-Means + NN	80.75	0.9570	0.6930	0.8039

The SVM + RF hybrid model recorded the highest overall accuracy of 84.08% and the highest precision of 0.9716. The best-performing model for recall and F1-score, however, was the stand-alone NN model with 0.7522 and 0.8425, respectively. These results indicate that hybrid models do not always outperform standalone models across all metrics compared. Instead, each model shows different strengths depending on whether the IDS prioritizes overall correctness, false-alarm reduction, or attack coverage.

B. Accuracy Analysis

As shown in Table I, classification accuracy for all models tends to converge within the 80-84% range. Although these

results are lower than those reported in prior NSL-KDD studies [7]-[9], these results are justifiable as the composition of the KDDTest+ consists of completely new and unseen subcategories of malicious records. During the training and validation phase, all the selected models reported exceptionally high stratified cross-validation accuracies of 98-99%. The accuracy drops from 98-99% during cross-validation to 80-84% on KDDTest+, demonstrating that the models performed much better on the training distribution than on unseen test data.

C. Precision Analysis

In an operational intrusion detection system (IDS), high detection precision is essential for minimizing false-positive alerts, which can cause unnecessary administrative burden and alert fatigue. Based on the results, RF-based models consistently achieved high precision scores, led by the SVM + RF hybrid model at 0.9716, standalone RF at 0.9701, and K-Means + RF at 0.9692. This performance is consistent with the ensemble voting properties of RF, which improve classification stability and reduce the influence of noisy or anomalous samples. Such high precision ensures that flagged threats by RF-based models are highly likely genuine threats, thus allowing network security teams to respond immediately without wasting effort on false alarms.

D. Recall Analysis

While precision governs false alarms, recall measures sensitivity of IDSs on missed exploits (false negatives). The results reveal that the standalone NN achieved the highest recall score at 0.7522, followed by the standalone SVM at 0.7436. These results indicate that the multi-layered weight adjustments in the NN and the continuous high-dimensional hyperplane mapping of the SVM can detect malicious behaviours which may go undetected under decision tree-based models. Since a missed attack is more hazardous than generating a false positive, high-recall models are preferred for maximum threat containment. Thus, a low recall score represents a catastrophic vulnerability in a live network environment, as undetected malicious traffic can bypass defences.

E. F1-Score Analysis

The F1-score reflects the balance between precision and recall. It also provides a better evaluation metric than accuracy alone, especially for imbalanced datasets, such as those presented by NSL-KDD. From Table I, the standalone NN achieved the highest F1-score of 0.8425, SVM + RF hybrid model followed with 0.8415. These results imply that maximizing accuracy alone does not necessarily optimize the system. For an environment that prioritizes a balance between attack sensitivity and low alert rates, NN demonstrates statistical superiority due to its flexible representation mapping. In cases where false-alarm suppression is an absolute need, then SVM + RF hybrid model can be a reasonable alternative.

F. Confusion Matrix Analysis

Fig. 2 presents the confusion matrices, which show the error distribution underlying the performance metrics reported in the previous sections. For intrusion detection, a false positive occurs when normal traffic is incorrectly classified as an attack, while a false negative occurs when attack traffic goes undetected. This difference is important as a model with high

accuracy may still be unsuitable for deployment if it produces excessive false alarms. The SVM + RF hybrid in Fig. 2(e) produced 278 false positives, which explains its strong precision result of 0.9716. This result indicates that only a small proportion of the traffic classified as attacks was actually normal, thus the model is suitable for environments where excessive false alarms may increase analyst workload and delay incident response. A similar precision-oriented result is also demonstrated by other RF-based models, such as RF and K-Means + RF, as shown in Fig. 2(a) and Fig. 2(f). Although these models achieve a low false alarm score, they also miss more attacks. This result makes these models a more conservative option rather than an aggressive alternative.

The NN in Fig. 2(d) shows a different detection profile in comparison to RF-based models. This model correctly detected the highest number of attack records and produced the lowest number of missed attacks. However, it also generated more false-positive alerts than the RF-based models. This detection-oriented performance may be valuable in security monitoring environments where missed attacks pose a greater risk than additional false alarms. Similarly, the SVM model shown in Fig. 2(c) exhibits the same trend, with broader attack coverage than

the more conservative RF-based models. Although SVM model offers better attack coverage, this performance involves a trade-off with a higher false-positive rate.

In general, the results shown in Fig. 2(a)–(h) confirm the main pattern reported in the metric results. The SVM + RF is more effective when priority is on false-alarm reduction. In contrast, NN and SVM are better alternatives when stronger attack detection coverage is required. Based on the results obtained, it is evident that there is no universally optimal model for every case. Therefore, no single model is preferable in all cases and the selection should consider operational requirements.

G. Standalone vs. Hybrid Model Discussion

One key aim of this study is to evaluate whether hybrid models consistently outperform standalone counterparts. The results show that this is not always the case. The SVM + RF model improved accuracy and precision, whereas K-Means-based hybrids did not show similar improvement. This result indicates that clustering may introduce additional noise when the cluster structure does not correspond well with the binary normal and attack classes.

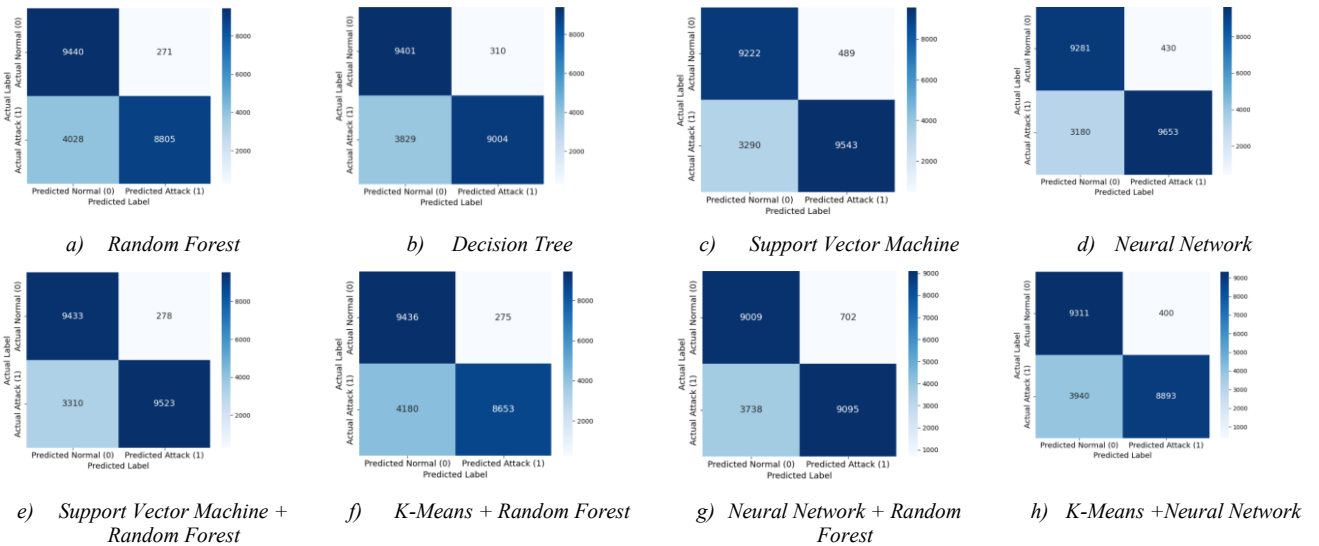


Fig. 2. Confusion matrices of the evaluated standalone and hybrid models on the KDDTest+ dataset.

TABLE II. VALIDATION-TEST ACCURACY DIFFERENCES AND MODEL INTERPRETATION

Model	Validation Accuracy (%)	Test Accuracy (%)	Diff. (%)	Interpretation
RF	99.68	80.93	18.75	Large validation–test performance gap, indicating limited generalization to unseen attack traffic.
DT	98.54	81.64	16.90	Significant generalization gap due to structural memorization during training session.
SVM	99.60	83.24	16.36	High training convergence but bounded testing accuracy due to rigid hyperplane margins.
NN	98.99	83.99	15.00	Most stable standalone model. Superior continuous weight optimization features.
SVM+RF	99.67	84.08	15.59	Top performer. Hybrid architecture marginally optimizes overall decision boundaries.
K-Means + RF	99.84	80.24	19.60	Unsupervised clustering may have introduced noise because the resulting clusters did not align well with the normal and attack classes.
NN + RF	99.86	80.31	19.55	Large validation–test performance gap. The soft-voting ensemble did not improve generalization on unseen KDDTest+ traffic.
K-Means +NN	99.53	80.75	18.78	Large validation–test performance gap. The K-Means clustering stage did not provide sufficient class separation for the neural classifier.

Therefore, under such conditions, simpler standalone models can remain competitive and may be more practical when computational cost and interpretability are important. The NN + RF hybrid revealed an additional finding. Although the combination of NN and RF was expected to improve predictive performance, simply merging their predictions together via a voting system did not provide a measurable advantage. The two models appeared to identify many of the same correctly classified instances, which limited the benefit of combining their predictions. As a result, the ensemble increased model complexity without providing a corresponding improvement in detection performance. This demonstrates that the integration of models does not inherently improve performance. If the individual model configurations fail to complement one another, the hybridization can degrade overall performance rather than enhance it.

H. Discussion on Training Validation and Testing Gap

Table II shows a clear drop in accuracy from the cross-validation results on KDDTrain+ to the final evaluation on KDDTest+. While most models achieved approximately 98-99% accuracy during cross-validation, their test accuracy decreased to about 80-84% on KDDTest+. This difference indicates that strong cross-validation performance on KDDTrain+ does not necessarily translate into robust generalization on KDDTest+. One possible reason is that KDDTest+ contains attack traffic that is absent or less well represented in the training data, which makes the final evaluation more challenging. The SVM + RF hybrid model achieved the highest final accuracy of 84.08%, suggesting that hybrid on probability-based feature augmentation can marginally stabilize classification decisions. In contrast, K-Means-based hybrids produced weaker results, indicating that centroid-based partitioning may not provide useful intermediate representations for this binary anomaly detection task. The differences between validation and test accuracies further suggest that cross-validation accuracy alone is not sufficient as the final model benchmark. Thus, evaluation on an independent test set is necessary to assess IDS model.

There are several limitations that are worth noting. First, this study relies on the historical NSL-KDD dataset rather than live, modern traffic profiles. This dataset is also limited to binary classification, manually tuned hyperparameter configurations, and an offline evaluation environment. These factors may have contributed to the generalization gap on the independent test set. Furthermore, the performance of the K-Means hybrid variants remains highly sensitive to initial spatial cluster quality. Consequently, the findings of this study may not fully reflect generalizable model performance in real-world deployments, but rather remain bound to the characteristics of the evaluated dataset. However, the result provides a reliable baseline indicator for model performance under fixed benchmark conditions. Nevertheless, this study contributes to machine learning-based cybersecurity by demonstrating that targeted model hybridization can improve anomaly-detection metrics, although its benefits depend on the model combination and IDS's operational objective. Future work will extend these evaluations to real-time detection environments, diverse large-

scale datasets, and decentralized IoT or cloud topologies to improve the operational robustness of IDSs.

V. CONCLUSION

This paper evaluated standalone and hybrid machine learning models for binary network anomaly detection using the NSL-KDD dataset. The SVM + RF hybrid achieved the best final accuracy and precision, making it suitable for IDS settings that prioritize false-alarm reduction. The standalone Neural Network achieved the best recall and F1-score, making it more suitable when attack detection sensitivity is more important. Overall, the findings show that hybrid models are not automatically superior; their effectiveness depends on how well the hybrid component enhances network traffic representation. Future work will extend the evaluation to other benchmark datasets such as CIC-IDS2017, CIC-IDS2018, and UNSW-NB15, and will move from binary detection to multiclass attack classification with per-attack-category analysis. Further work shall also apply systematic hyperparameter optimization, measure inference latency and memory overhead, and integrate explainable AI methods such as SHAP to make IDS decisions more interpretable for cybersecurity analysts.

ACKNOWLEDGMENT

The authors would like to acknowledge Universiti Malaysia Sarawak for the academic support, research environment, and facilities provided throughout the completion of this work.

REFERENCES

- [1] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure," *Sensors*, vol. 23, no. 5, Art. no. 2415, 2023, doi: 10.3390/s23052415.
- [2] J. Diaz-Verdejo, J. Muñoz-Calle, A. Estepa Alonso, R. Estepa Alonso, and G. Madinabeitia, "On the detection capabilities of signature-based intrusion detection systems in the context of web attacks," *Appl. Sci.*, vol. 12, no. 2, Art. no. 852, 2022, doi: 10.3390/app12020852.
- [3] K. Arshad *et al.*, "Deep reinforcement learning for anomaly detection: A systematic review," *IEEE Access*, vol. 10, pp. 124017–124035, 2022, doi: 10.1109/ACCESS.2022.3224023.
- [4] C. Liu, Z. Gu, and J. Wang, "A hybrid intrusion detection system based on scalable K-Means+ random forest and deep learning," *IEEE Access*, vol. 9, pp. 75729–75740, 2021, doi: 10.1109/ACCESS.2021.3082147.
- [5] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022, doi: 10.1109/ACCESS.2022.3206425.
- [6] A. Alsajri and A. Steiti, "Intrusion detection system based on machine learning algorithms: SVM and genetic algorithm," *Babylonian J. Mach. Learn.*, vol. 2024, pp. 15–29, 2024, doi: 10.58496/BJML/2024/002.
- [7] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD Cup 99 dataset employing SVMs," in *Proc. 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Dhaka, Bangladesh, 2014, pp. 1–6, doi: 10.1109/SKIMA.2014.7083539.
- [8] D. Kshirsagar and S. Kumar, "An efficient feature reduction method for the detection of DoS attack," *ICT Express*, vol. 7, no. 3, pp. 371–375, 2021, doi: 10.1016/j.icte.2020.12.006.
- [9] C. Thana-Aksaneekorn, S. Kosolsombat, and T. Luangwiriya, "Machine learning classification for intrusion detection systems using the NSL-KDD dataset," in *Proc. IEEE Int. Conf. Cybern. Innov. (ICCI)*, 2024, pp. 1–6, doi: 10.1109/ICCI60780.2024.10532265.