Internship Report: Evaluating Machine Learning Models for Cybersecurity Intrusion Detection Using the NSL-KDD Dataset

Mohammed Bekkouche
LabRI-SBA Laboratory, Ecole Supérieure en Informatique (ESI-SBA), Sidi Bel Abbès, Algeria
Internship at ReDCAD Laboratory, Sfax University, Tunisia

12 December 2025 – 22 December 2025

Funded by: Ecole Supérieure en Informatique (ESI-SBA), Algeria

Abstract

This internship focused on evaluating machine learning models for intrusion detection to improve cybersecurity, using the NSL-KDD dataset, a widely used benchmark for distinguishing malicious traffic (DoS, Probe, U2R, R2L) from normal network traffic. The study aims to enhance research capacity and foster international cooperation from **ESI-SBA** (Algeria) with **Sfax University**, the **ReDCAD Laboratory**, the **High Institute of Computer Science and Multimedia at Sfax University**, **Digital Research Center at Sfax**, and **National School of Engineering at Sfax University** (Tunisia).

Contents

1 Introduction

An Intrusion Detection System (IDS) is a security mechanism that monitors network and/or system activities to identify malicious behavior. It analyzes network traffic and system logs to detect potential threats and, when suspicious activity is detected, generates alerts for administrators or a central security system without directly blocking the attack [9]. This behavior distinguishes an IDS from an Intrusion Prevention System (IPS), which can automatically take actions to block or mitigate detected threats.

Since the 1980s, IDS technologies have evolved significantly to address the growing challenges of network security. Despite these advancements, many existing IDS solutions remain limited in their ability to accurately detect diverse and complex attack types. Therefore, precise and reliable intrusion detection systems are essential to enhance the security of networked environments.

Researchers have investigated the use of artificial intelligence techniques, including machine learning and deep learning, to address the security challenges of IDSs. These methods are capable of identifying significant patterns and relationships within large-scale data [8].

The aim of this internship was to evaluate the performance of various machine learning models for network intrusion detection in the context of cybersecurity. The work focused on analyzing different categories of attacks, including Denial of Service (DoS), Probing, User-to-Root (U2R), and Remote-to-Local (R2L), and on designing experiments to detect these attacks using the NSL-KDD dataset.

2 Internship Aims

The main aims of this internship are:

a) Review the state-of-the-art in intrusion detection systems (IDS) and machine learning techniques.

b) Preprocess and analyze the NSL-KDD dataset.

c) Implement and evaluate machine learning models (e.g., Decision Trees, SVM, Random Forests, Neural Networks) for intrusion detection.

d) Compare the performance metrics of models for different attack categories.

e) Provide recommendations for future research and potential enhancements in IDS.

3 IDS Types

Any illegal access to network and system data that undermine its availability, integrity, and confidentiality is referred as an intrusion. An IDS is a software that continuously surveils the network and computer system and identify any possible intrusion. The IDS is classfied according to its deployment and detection method, as illustrated in the figure 1 [9].

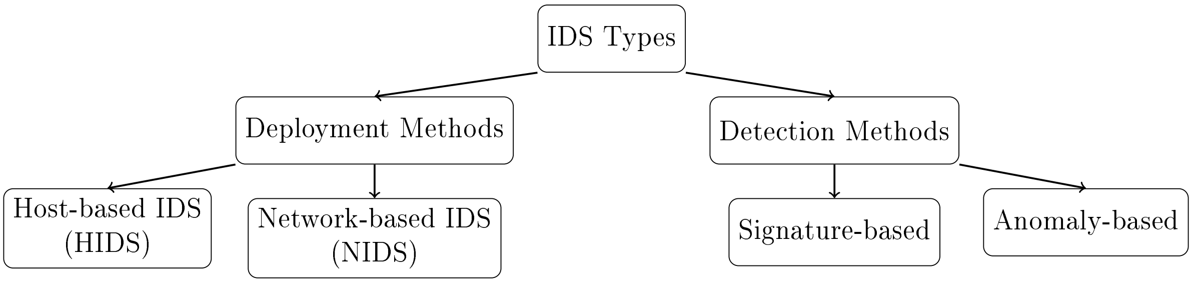

Figure 1: IDS classification by deployment and detection methods

3.1 Deployment-based IDS

IDSs can be categorized based on the types of events they monitor and their deployment methods. The main deployment-based IDS types are:

1. **Network-based IDS (NIDS)** monitors network traffic within a specific network segment and examines activity to identify suspicious behavior. It is generally deployed at the boundary between networks to detect attacks targeting multiple hosts.

2. **Host-based IDS (HIDS)** monitors events on a single host and analyzes them for malicious activity. HIDS are typically deployed on critical hosts, such as servers containing sensitive information or publicly accessible servers.

3.2 Detection-based IDS

IDS can also be classified based on the method used to detect intrusions. The main detection-based IDS types are:

1. **Signature-based detection:** A signature is a pattern that corresponds to a known threat. In signature-based detection, observed events are compared with signature patterns, and if a match occurs, the event is classified as a threat. This method is effective for detecting known threats but is less efficient in identifying unknown or novel attacks.

2. **Anomaly-based detection:** This method compares observed events with descriptions of normal behavior and identifies significant irregularities. Events that differ substantially from expected patterns are flagged as potential threats.

4 Methodology

4.1 Dataset Description

The NSL-KDD dataset [2] is a refined version of the KDD Cup 1999 dataset. It contains labeled network traffic records classified as normal or as different attack types. The NSL-KDD dataset is widely used for evaluating NIDSs and for testing machine learning methods in cybersecurity applications. Table 1 summarizes the four main attack categories along with the number of subtypes for each.

Table 1: Number of Subtypes for Each Attack Type in NSL-KDD

Attack Type	**Number of Subtypes**
Denial of Service (DoS)	10
Probing (Probe)	6
User-to-Root (U2R)	7
Remote-to-Local (R2L)	15

Each record also includes features such as duration, protocol type, service, flags, bytes transmitted, and other statistical measures.

DoS Attacks

Denial-of-Service (DoS) attacks aim to make a system or service unavailable. They achieve this by sending large numbers of requests, consuming system resources, or exploiting protocol vulnerabilities that can cause the target to freeze, crash, or stop responding.
Example (Neptune): An attacker sends a large number of TCP SYN packets to a web server while masking the true source IP addresses. As a result, the server's connection table reaches its limit, preventing legitimate users from accessing the website.

Probe Attacks

Probe attacks are network intrusions intended to gather information about a target network. Unlike attacks that immediately disrupt services, probe attacks focus on reconnaissance to determine active hosts, open ports and services, system configurations, and potential vulnerabilities.
Example (Nmap Port Scan): An attacker scans a corporate network to determine which hosts are responsive and which ports are open. As a result, the attacker builds a map of active hosts and exposed services, which can later support further intrusion steps.

R2L Attacks

Remote-to-Local (R2L) attacks occur when an attacker communicates from an external machine and attempts to obtain local access or elevated privileges on a target system

without possessing a legitimate account.
Example (FTP Write Exploit): An attacker connects to an FTP server that allows unauthenticated write access. The attacker uploads a malicious script, which executes with the server's permissions, allowing access to files that are normally restricted.

U2R Attacks

User-to-Root (U2R) attacks happen when an intruder already has normal user-level access on a system and attempts to escalate privileges to administrator-level (root) control.
Example (Memory-corruption exploit in a vulnerable program): A normal user runs a program that processes input incorrectly. The program stores user input in a fixed-size memory region but does not verify the length or structure of the received data. As a result, internal execution data are altered, causing the program to execute commands with administrator-level permissions, since it runs with elevated privileges. This allows the attacker to move from standard user permissions to root-level control.

Table 2 provides an summary of the number of records per category in both the training and test sets of the NSL-KDD dataset. This breakdown highlights the class distribution and the imbalance across attack types, especially for U2R and R2L attacks.

Table 2: NSL-KDD Dataset: Number of Records per Category in Training and Test Sets

Category	**Training Set**	**Test Set**
Normal	67,343	9,711
DoS	45,927	7,167
Probe	11,656	2,421
U2R	119	67
R2L	4,173	3,178

4.2 Data Preprocessing

- Preprocessing data, including handling missing values, encoding categorical features, and normalizing/scaling numerical features.
- Selecting relevant features based on correlation analysis and feature importance.

4.3 Machine Learning Models

This subsection describes the machine learning models used in this internship for intrusion detection:

- **Decision Tree (DT) Classifier** [3]: DT is a rule-based classifier that splits the feature space into decision regions based on conditions on feature values, making it interpretable and suitable for sparse data. It was applied to classify normal and attack traffic in the NSL-KDD dataset.
- **Support Vector Machines (SVM)** [4]: SVM is a boundary-based classifier that finds the optimal hyperplane to separate normal and anomalous sequences in the feature space. It was used to detect various types of network attacks.

- **Random Forest (RF)** [5]: RF is an ensemble learning method that builds multiple decision trees and aggregates their predictions, typically through majority voting. In this internship, it was used to improve the robustness of intrusion detection across different attack types.
- **XGBoost (XB)** [6]: XB is a gradient-boosted decision tree algorithm that builds trees sequentially, where each new tree corrects the errors of the previous ones. It was employed to enhance detection accuracy and handle complex patterns in the NSL-KDD dataset.
- **Principal Component Analysis (PCA)** [10]: PCA is trained on normal NSL-KDD traffic to learn typical network behaior. New samples are projected and reconstructed from the PCA space, and the reconstruction error is computed. Normal traffic has low error, while attack traffic has high error. A threshold on this error is used to classify connections as normal or attack.

4.4 Evaluation Metrics

The performance of the intrusion detection models is evaluated using the following metrics, derived from the confusion matrix:

- **Accuracy** measures the proportion of correctly classified instances:
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
- **Precision** indicates the proportion of detected attacks that are truly attacks:
$$\text{Precision} = \frac{TP}{TP + FP}$$
- **Recall** (also known as Detection Rate) measures the proportion of actual attacks that are correctly identified:
$$\text{Recall} = \frac{TP}{TP + FN}$$
- **F1-score** is the harmonic mean of Precision and Recall:
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
- **False Positive Rate (FPR)** measures the proportion of normal traffic incorrectly classified as attacks:
$$\text{FPR} = \frac{FP}{FP + TN}$$

5 Results and Discussion

This section presents the experimental results obtained during the evaluation of machine learning models for intrusion detection using the NSL-KDD dataset. The evaluation was performed under multiple classification scenarios: binary classification (Attack vs Normal) and multi-class classification (DoS, Probe, R2L, U2R, Normal).

5.1 Binary Classification: Attack vs Normal

Table 3 summarizes the performance of five machine learning models in separating malicious traffic from normal traffic. Metrics reported include Accuracy, Precision, Recall, F1-score, and False Positive Rate (FPR).

Table 3: Binary Classification Performance (Attack vs Normal)

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**FPR**
DT	0.77	0.97	0.62	0.76	**2.81%**
SVM	0.69	0.89	0.53	0.66	9.04%
RF	0.77	0.97	0.62	0.75	2.85%
XB	**0.79**	**0.97**	**0.65**	**0.78**	2.84%
PCA	0.75	0.96	0.59	0.73	3.00%

XB achieved the highest accuracy and F1-score, while SVM showed a higher false positive rate, indicating it misclassified more normal traffic as attacks. DT, RF, and PCA performed similarly in terms of recall and precision, demonstrating their effectiveness for detecting known attack patterns.

5.2 Multi-Class Classification: DoS, Probe, R2L, U2R, Normal

Table 4 shows the per-class recall for the five main categories. This evaluation highlights how well models detect each attack type and handles class imbalance, particularly for rare classes such as U2R and R2L.

Table 4: Per-Class Recall for 5-Class Classification

Model	**Normal**	**DoS**	**Probe**	**R2L**	**U2R**
DT	**0.97**	084	0.57	0.02	0.04
SVM	0.92	0.67	0.14	0.00	0.00
RF	**0.97**	0.80	**0.66**	0.04	0.03
XB	**0.97**	**0.86**	**0.66**	**0.06**	**0.09**

DoS attacks are detected with high recall across all models, due to their distinctive traffic patterns. Probe attacks show moderate detection rates; their network patterns can resemble normal activity, making them slightly more difficult to detect. R2L and U2R attacks are the most challenging to identify due to their rarity and subtle behavior, XB achieves slightly higher recall for these attacks, indicating its ability to capture complex patterns.

5.3 Discussion

Ensemble methods show superior performance compared to single classifiers. For the multi-class scenario, both RF and XB achieved better results than DT and SVM. In the binary classification, XB achieved higher performance than DT and SVM, while RF performed similarly to DT and PCA and better than SVM.

5.4 Selective Prediction Strategy for DT, SVM, RF, and XB

To improve classification accuracy and prediction reliability for DT, SVM, RF, and XB models, a selective prediction strategy is adopted. In this setting, the classifier issues a prediction when its confidence exceeds a predefined threshold; otherwise, it abstains. This mechanism allows the model to focus on samples for which it is sufficiently confident, while limiting unreliable decisions.

The aim is to improve prediction trustworthiness by accepting just high-confidence samples, while keeping the proportion of rejected samples under control. In our experiments, the threshold is selected to maximize accuracy under the constraint of a minimum coverage of 70%. This setting is particularly relevant for intrusion detection scenarios, where incorrect decisions, especially failing to detect attacks, can have serious consequences. Allowing the model to abstain when confidence is low provides an additional safety margin.

The best confidence threshold is determined using a validation subset corresponding to 10% of the test data, with known labels. For each candidate threshold, two quantities are computed:

- Accuracy on the accepted predictions.
- Coverage, defined as the fraction of samples whose confidence exceeds the threshold.

The selected threshold achieves the highest accuracy while satisfying the coverage constraint. This strategy ensures that the predictions produced by the model are highly dependable, while still covering a substantial portion of the data.

5.5 PCA with Selective Prediction

PCA is applied as a reconstruction-based approach for intrusion detection by learning patterns of normal network traffic from the training data. The reconstruction error serves as an anomaly score, where higher values indicate abnormal behavior.

To improve decision reliability, a selective prediction strategy is introduced. Two thresholds are defined: a lower threshold representing high-confidence normal traffic and an upper threshold representing high-confidence attacks. Predictions are made for samples with reconstruction errors below the lower threshold or above the upper threshold, while samples with errors between these thresholds are rejected.

In the selective prediction setup, labels from training set are just used to set the confidence thresholds, not to train the PCA model. PCA still does not see attack labels during training, it just learns normal traffic patterns.

Performance is assessed on accepted predictions using accuracy and coverage metrics. This strategy improves the reliability of intrusion detection by focusing on high-confidence decisions and limiting uncertain classifications.

5.6 Binary Classification Results under the Selective Prediction Strategy

Table 5 presents the results for binary classification (Attack vs Normal) obtained using the selective prediction strategy. In this setting, predictions are produced just when the model confidence exceeds the selected threshold, while low-confidence samples are rejected.

Table 5: Binary intrusion detection results under the selective prediction strategy

Model	**Accuracy with 100% coverage**	**Accuracy**	**Coverage**	**Best Threshold**
DT	0.772	**0.804**	**0.920**	0.998
SVM	0.693	0.810	0.700	0.878
RF	0.771	0.771	1.000	0.100
XB	0.791	**0.895**	**0.776**	0.999
PCA	0.754	**0.865**	**0.891**	/

Compared to the baseline scenario with full coverage, selective prediction leads to higher accuracy for most models. The DT benefits from this strategy, with accuracy increasing from 0.772 to 0.804 while maintaining a high coverage of 92%, indicating that misclassifications are largely concentrated among low-confidence predictions. The SVM exhibits a marked improvement, achieving an accuracy of 0.810 at the minimum required coverage of 70%, which confirms the relevance of confidence-based rejection for this model.

RF shows identical accuracy before and after applying selective prediction, with full coverage, suggesting that the confidence scores produced by this model do not effectively separate correct and incorrect predictions. In contrast, XB achieves the largest improvement under the selective prediction strategy, with accuracy rising from 0.791 to 0.895 while retaining more than 77% coverage, highlighting its strong confidence discrimination capability in the binary intrusion detection task.

The PCA-based approach also benefits significantly from the selective prediction strategy. Its accuracy increases from 0.754 to 0.865 at 89.1% coverage, demonstrating that reconstruction error provides a reliable confidence measure for distinguishing between normal and attack traffic.

5.7 Multi-Class Classification Results under the Selective Prediction Strategy

Table 6 summarizes the multi-class classification results when the selective prediction strategy is applied. This task is more challenging due to the presence of multiple attack categories and class imbalance.

Table 6: Multi-class intrusion detection results under the selective prediction strategy

Model	**Accuracy with 100% coverage**	**Accuracy**	**Coverage**	**Best Threshold**
DT	0.752	**0.854**	**0.819**	0.998
SVM	0.624	0.684	0.801	0.649
RF	0.750	0.750	1.000	0.100
XB	0.772	**0.886**	**0.805**	0.999

The DT shows a substantial improvement under selective prediction, with accuracy increasing from 0.752 to 0.854 at 81.9% coverage. The SVM achieves a moderate increase in accuracy, indicating that rejecting uncertain predictions partially mitigates its limitations in the multi-class setting.

Similar to the binary scenario, RF remains unaffected by selective prediction, preserving full coverage and unchanged accuracy. XB demonstrates the strongest response to this strategy, with accuracy increasing from 0.772 to 0.886 while maintaining over 80% coverage. This confirms that selective prediction is particularly effective when paired with models capable of generating reliable confidence estimates across multiple classes.

5.8 Discussion for the Selective Prediction Strategy

The experimental results clearly indicate that the selective prediction strategy enhances classification reliability by restricting decisions to high-confidence samples. This approach proves particularly beneficial in intrusion detection scenarios, where incorrect predictions, such as missed attacks, can lead to severe security risks.

Models that produce well-separated confidence scores, particularly XB, benefit the most from selective prediction in both binary and multi-class settings. In contrast, RF shows limited sensitivity to confidence-based rejection, suggesting that its probability estimates may require calibration to fully utilize this strategy.

6 Conclusion

This intership focused on evaluating the performance of various machine learning models for ntework intrusion detection using the NSL-KDD dataset. The important findings are summarized as follows:

- **Effectiveness of different machine learning models:** Ensemble methods, particularly XB, demonstrated superior performance in both binary and multi-class classification scenarios. DT and RF also performed well, while SVM showed lower performance, especially in detecting less frequent attack types.

- **Challenges encountered in detecting certain attack types:** Attacks such as U2R and R2L were more difficult to detect due to their rarity and similarity to normal traffic. Class imbalance significantly affected model performance, highlighting the need for proper handling of minority classes in intrusion detection datasets.

- **Results with the selective prediction strategy:** Applying selective prediction improved the reliability of model predictions by allowing the classifier to abstain on low-confidence samples. XB benefited the most, achieving higher accuracy while maintaining substantial coverage. This approach proved particularly effective in mitigating misclassifications for critical attack types in both binary and multi-class scenarios. Additionally, the results for PCA in binary intrusion detection show that, when combined with selective prediction, it can effectively reduce misclassifications of critical attack samples while maintaining high coverage, demonstrating its competitive performance despite being an unsupervised, reconstruction-based method.

- **Recommendations for improving IDS performance and future work:** Future work includes exploring advanced feature engineering techniques, deep learning models, and calibration methods to further improve the detection of rare attack types. Evaluating the proposed approaches on real-world network traffic would provide valuable insights into deployment feasibility and system robustness. In addition, the use of explainable artificial intelligence tools such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) is considered to measure feature importance in attack prediction. This direction aims to detect attacks in network traffic while also providing clear explanations that can help security experts better understand and trust the model predictions.

These findings demonstrate the potential of machine learning techniques, combined with selective prediction, to improve cybersecurity through reliable intrusion detection while identifying avenues for further research and improvement.

7 Acknowledgments

This internship was funded by the **Ecole Supérieure en Informatique (ESI-SBA), Algeria**, with the aim of strengthening research capacity and promoting international cooperation with the **Sfax University**, **ReDCAD Laboratory**, the **High Institute of Computer Science and Multimedia at Sfax**, **Digital Research Center at Sfax**, and **National School of Engineering at Sfax (Tunisia)**.

I would like to express my sincere gratitude to all members of the **High Institute of Computer Science and Multimedia of Sfax**, in particular **Dr. Ismail Bouassida**, for giving me the opportunity to present a seminar to students of this institution on the application of machine learning techniques for log-based anomaly detection to improve system security, as well as for their warm welcome, support, and valuable scientific discussions during the internship. I also warmly thank all members of the **ReDCAD Laboratory**, **Digital Research Center at Sfax**, and **National School of Engineering at Sfax** for their support and constructive scientific discussions throughout the internship.

References

[1] Roesch, M. (1999, November). Snort: Lightweight intrusion detection for networks. In Lisa (Vol. 99, No. 1, pp. 229-238).

[2] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). Ieee.

[3] Chen, M., Zheng, A. X., Lloyd, J., Jordan, M. I., & Brewer, E. (2004, May). Failure diagnosis using decision trees. In International Conference on Autonomic Computing, 2004. Proceedings. (pp. 36-43). IEEE.

[4] Liang, Y., Zhang, Y., Xiong, H., & Sahoo, R. (2007, October). Failure prediction in ibm bluegene/l event logs. In Seventh IEEE International Conference on Data Mining (ICDM 2007) (pp. 583-588). IEEE.

[5] Anton, S. D. D., Sinha, S., & Schotten, H. D. (2019, September). Anomaly-based intrusion detection in industrial data with SVM and random forests. In 2019 International conference on software, telecommunications and computer networks (SoftCOM) (pp. 1-6). IEEE.

[6] Henriques, J., Caldeira, F., Cruz, T., & Simões, P. (2020). Combining k-means and xgboost models for anomaly detection using log datasets. Electronics, 9(7), 1164.

[7] Abdulganiyu, O. H., Tchakoucht, T. A., & Saheed, Y. K. (2024). RETRACTED ARTICLE: Towards an efficient model for network intrusion detection system (IDS): systematic literature review. Wireless networks, 30(1), 453-482.

[8] Saied, M., Guirguis, S., & Madbouly, M. (2024). Review of artificial intelligence for enhancing intrusion detection in the internet of things. Engineering Applications of Artificial Intelligence, 127, 107231.

[9] Chinnasamy, R., Subramanian, M., Easwaramoorthy, S. V., & Cho, J. (2025). Deep learning-driven methods for network-based intrusion detection systems: A systematic review. ICT Express.

[10] Bekkouche, M., Benslimane, S. M. (2025). Improving Anomaly Detection in the HDFS Dataset with Novel Machine Learning Models and Techniques. Computer Science Journal of Moldova, 99(3).