

# ML Project Proposal

Proactive Detection of School Cybersecurity  
Attacks using Supervised Learning

Prepared by :

**Angeli De Los Reyes**



# Contents

1. Problem Statement
2. Proposed Solution
3. Background
3. Machine Learning Solution Exploration
4. Summary of Reviewed ML Solutions
5. Proposed ML Solution
6. Evaluation of Machine Learning Models
7. Machine Learning Project Flow
8. References



## 1. Problem Statement

Queensland University faces increasing risks of cyberattacks targeting its digital infrastructure, reflecting a broader trend in the educational sector. Recent reports from the Australian Cyber Security Centre (ACSC) indicate a 17% rise in cyberattacks on the education sector from 2021 to 2022, with an average of 3,934 attacks per week in July 2022 (Brown, 2023). Additionally, 60% of educational institutions reported experiencing ransomware attacks in 2021, up from 44% in 2020, often resulting in recovery times exceeding three months (Henebery, 2023). To proactively detect and mitigate these threats, there is a need for a robust, supervised machine learning (ML) solution that leverages labeled network traffic data to identify malicious activities accurately.

## 2. Proposed Solution

Develop and deploy a proactive machine learning-powered Intrusion Detection System utilizing supervised learning techniques. This system will:

- **Proactively detect known and emerging cyber threats:** Analyze network traffic data to identify malicious activities (e.g., web attacks, infiltration, DDoS attacks) and classify network flows into benign or attack categories, utilizing a labeled dataset ([NF-CSE-CIC-IDS2018.csv](#))
- **Improve threat intelligence:** Gather insights into specific cyberthreats targeting the education sector in Australia.
- **Minimize downtime and operational disruption:** Reduce the impact of cyberattacks by enabling faster incident response and recovery.
- **Generate real-time alerts:** Trigger timely notifications to the school's IT team, enabling rapid response and mitigation of attacks.

## 3. Background

The Australian Cyber Security Centre (ACSC) has reported a significant increase in cyberattacks targeting the education sector, emphasizing the need for robust cybersecurity measures<sup>1</sup>. In light of these growing threats, various Network Intrusion Detection System (NIDS) approaches have been developed by experts and researchers over the years to counteract the complexity of cyberattacks.

The table below outlines these types, highlighting the pros and cons of each, and why ML-based NIDS are increasingly preferred:

**Table 1:** Comparison of the different types of NIDS<sup>2</sup>

Type of NIDS	Description	Advantages	Disadvantages	Preferred?
<b>Signature-based NIDS</b>	Detects intrusions by comparing network traffic to a database of known attack patterns (signatures).	- High accuracy for known threats - Low false positive rate	- Cannot detect new or unknown attacks - Requires frequent updates	Less preferred due to inability to detect new threats.
<b>Anomaly-based NIDS</b>	Identifies deviations from normal behavior patterns to detect anomalies or potential threats.	- Can detect unknown or novel attacks - Adaptive to new threat patterns	- Higher false positive rate - Requires extensive profiling of normal behavior	Useful for detecting unknown threats, but requires careful tuning.

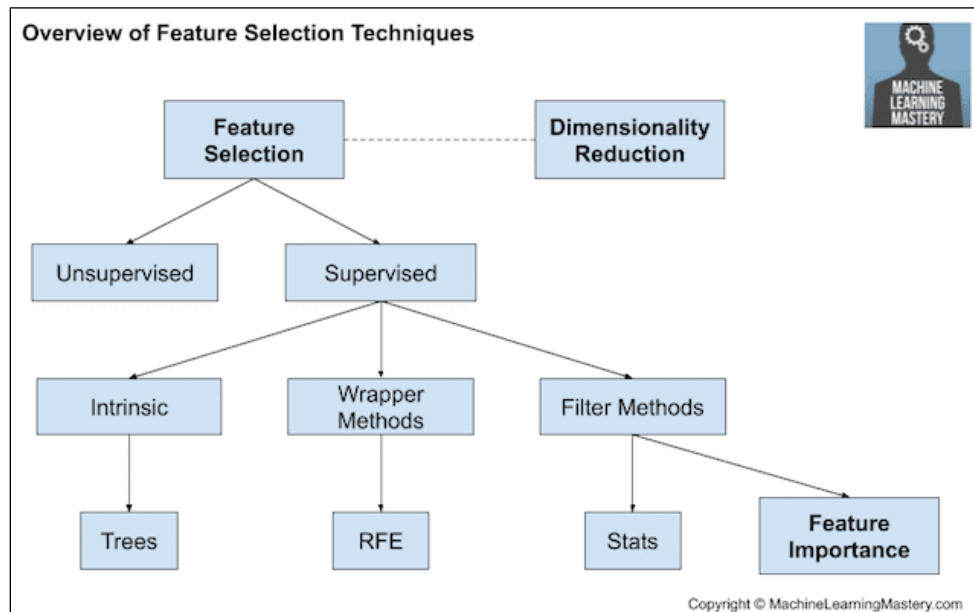
<sup>1</sup> Australian Cyber Security Centre. (2021). ACSC Annual Cyber Threat Report.

<sup>2</sup> ChatGPT, response to "What are the different types of NIDS and how are they different", OpenAI, January 10, 2025.

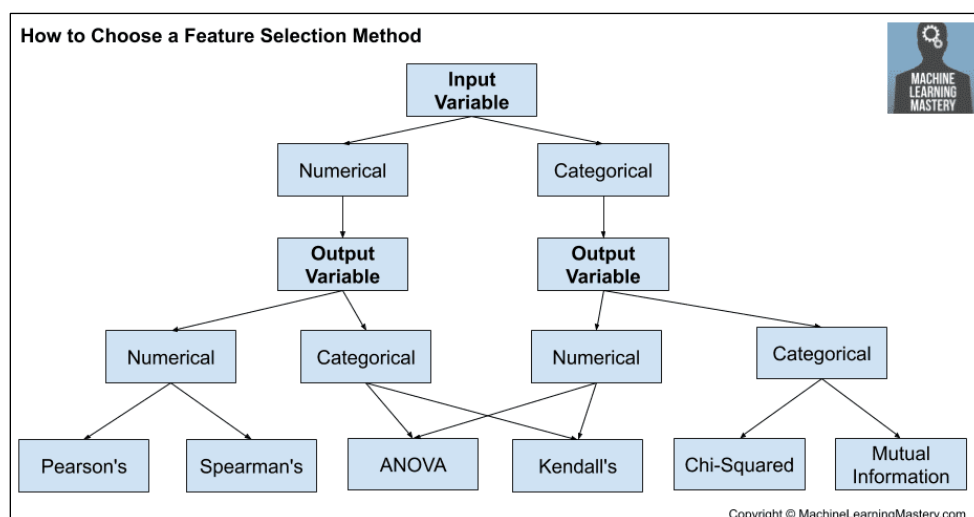


While these classifiers often incorporate embedded feature selection during the model training process, we can explore manual feature engineering and selection (Figures 2 & 3) based on domain expertise, such as removing low-variance or highly correlated features. Additionally, we will utilize the embedded feature selection methods, like feature importance in Random Forests or GBM, to further refine our feature set. We then perform an iterative refinement process guided by both manual and automated methods to enhance model performance and provide valuable insights.

**Figure 2:** Overview of Feature Selection Techniques<sup>4</sup>




**Figure 3:** Univariate Statistical Measures for Filter-Based Feature Selection Methods<sup>5</sup>




<sup>4</sup> Brownlee, J. (2020). How to Choose a Feature Selection Method For Machine Learning. Retrieved from Machine Learning Mastery: <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>

<sup>5</sup> Brownlee, J. (2020). How to Choose a Feature Selection Method For Machine Learning. Retrieved from Machine Learning Mastery: <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>



Next, we will delve into more advanced machine learning techniques, drawing on research-backed studies specifically focused on Intrusion Detection Systems. Below, we examine the top three machine learning solutions that best align with our objectives.

### Solution 1: Enhanced Random Forest (RF) with SMOTE

- **Relevant Literature:** Intrusion detection system combined enhanced random forest with SMOTE algorithm (Wu, et al., 2022)
  - **Description:** The proposed method is designed to improve detection accuracy in network intrusion systems by addressing the data imbalance issue that arises when minority class samples are underrepresented. The model integrates:
    - **SMOTE** for balancing datasets by synthesizing minority class samples.
    - **Enhanced Random Forest** optimized for decision tree similarity and classification performance.
  - **Implementation:**
    - **Data Preprocessing:**
      - Attributes of the dataset are normalized, and non-numerical features are converted to numerical or binary values.
      - The NSL-KDD dataset is used as the benchmark for testing the system.
    - **Dataset Balancing with K-Means and SMOTE:**
      - K-means clustering identifies and minimizes outlier samples.
      - SMOTE synthesizes new minority class samples, enhancing dataset balance and improving feature richness.
    - **Enhanced Random Forest:**
      - Decision trees with high performance are selected based on AUC (Area Under the Curve) scores.
      - Decision trees with low similarity are prioritized to ensure classifier diversity.
      - A similarity matrix adjusts predictions using rules based on detected attack types and their similarities.
    - **Optimization and Detection:**
      - The similarity matrix is generated for refining classification decisions.
      - The system applies majority voting and correction mechanisms to ensure accurate detection.
    - **Performance Testing:**
      - The model achieves high classification accuracy, precision, recall, and F1 scores compared to traditional methods.
  - **Advantages:**
    - **Improved Accuracy:** Achieved **99.72% training accuracy** and **78.47% testing accuracy** on the NSL-KDD dataset.
    - **Effective Imbalance Handling:** K-means and SMOTE significantly improve the detection of minority attack types.
    - **Robustness:** Enhanced random forest minimizes overfitting and improves model generalization.
    - **Flexibility:** The similarity matrix allows correction for misclassifications based on pre-defined rules.
  - **Disadvantages:**
    - **Computational Overhead:** K-means clustering and SMOTE add complexity to preprocessing.
    - **Limited Minority Detection:** Despite improvements, U2R and R2L classes exhibit lower detection rates compared to majority classes.
    - **Dataset Dependence:** Results are heavily reliant on the characteristics of the NSL-KDD dataset, which might limit generalizability.
- 

## Solution 2: Filter-Based Feature Selection with XGBoost

- **Relevant Literature:** Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset (Kasongo & Sun, 2020)
- **Description:**
  - This method aims to reduce the high dimensionality of the dataset and improve IDS performance, particularly for multiclass and binary classification problems. The **XGBoost** algorithm identifies the most relevant features, optimizing the training and prediction process for ML models like ANN, Decision Trees, and SVM.
  - The study achieved the highest accuracy rate of **90.85%** using the **Decision Tree (DT)** classifier in the **binary classification task** with the reduced feature set (19 features) selected via XGBoost.
- **Implementation:**
  - **Dataset Preprocessing:**
    - The UNSW-NB15 dataset (42 features) is used. It includes various attack types such as DoS, Shellcode, and Worms.
    - Non-numeric features are transformed, and normalization (Min-Max scaling) is applied to ensure balanced input for ML models.
  - **Feature Selection with XGBoost:**
    - XGBoost computes feature importance scores based on gradient boosting.
    - A subset of 19 features (from the original 42) is selected to optimize model training.
  - **ML Model Training:**
    - Multiple ML models, including **Artificial Neural Networks (ANN)**, **k-Nearest Neighbor (kNN)**, **Decision Trees (DT)**, **Logistic Regression (LR)**, and **Support Vector Machines (SVM)**, are tested.
    - Models are trained on the reduced feature set and evaluated on binary and multiclass classification tasks.
  - **Performance Metrics:**
    - Metrics like Accuracy (AC), Precision, Recall, and F1-Score are used to measure performance.
    - Results are compared across the full feature set (42 features) and reduced feature set (19 features).
- **Advantages:**
  - **Feature Reduction:**
    - Decreases model complexity and training time.
    - Improves generalization by removing irrelevant features.
  - **Performance Improvement:**
    - Using the reduced feature set, models show higher test accuracy, particularly for binary classification (e.g., DT accuracy increases from 88.13% to 90.85%).
  - **Adaptability:**
    - The XGBoost feature selection integrates well with various ML models, enhancing their predictive power.
- **Disadvantages:**
  - **Minority Class Challenge:**
    - The method underperforms for rare attack types (e.g., Backdoor, Shellcode, Worms), as these are minority classes in the dataset.
  - **Computational Cost:**
    - XGBoost's feature selection and tree-based operations require significant computational resources.
  - **Dependence on Dataset:**
    - Results may not generalize well beyond the UNSW-NB15 dataset without further adaptation.



### Solution 3: Support Vector Machine (SVM) with Radial Basis Function (RBF) Kernel

- **Relevant Literature:** Research on Application of Support Vector Machine in Intrusion Detection (Su, et al., 2021)
- **Description:**
  - The study uses **Support Vector Machines (SVM)** which transforms low-dimensional data into higher-dimensional spaces to identify optimal hyperplanes for classification.
  - The study ultimately chose the **Radial Basis Function (RBF)** kernel for the Support Vector Machine (SVM) classifier to achieve an average detection accuracy above 95%.
- **Implementation:**
  - **Dataset:**
    - The **KDDCUP99 dataset** (10% sample) was used, containing 41 features with diverse types of attacks, such as DOS, R2L, U2R, and Probing.
    - All attacks were grouped into a single "abnormal" class, and the system performed binary classification to detect normal vs. abnormal states.
  - **Preprocessing:**
    - **Normalization:** Scaled feature values to prevent large values from overshadowing smaller ones, reducing computational bias.
    - **Text-to-Numeric Conversion:** Textual attributes were converted into numerical values. For example, normal records were labeled as +1, and attack records as -1.
  - **Kernel Function Selection:**
    - The **Radial Basis Function (RBF)** kernel was chosen due to its robust classification capability and sensitivity to parameters.
    - The parameters (e.g., penalty factor C and kernel width  $\sigma^2$ ) were optimized using cross-validation.
  - **Training and Testing:**
    - SVM models were trained using the **LIBSVM** library and training was conducted on subsets of the dataset, and models were validated and tested on separate data.
    - Detection accuracy was calculated as the proportion of correctly classified instances in the test dataset.
- **Advantages:**
  - **High Accuracy:**
    - Achieved an average detection accuracy of **95%** on the test dataset.
  - **Efficiency:**
    - SVM's ability to handle small datasets and high-dimensional features reduced training and detection time.
  - **Adaptability:**
    - The model doesn't require all normal and abnormal data to achieve good performance, allowing for real-time updates and upgrades.
  - **Scalability:**
    - The system can be extended by combining SVM with other techniques (e.g., PCA) or using multiple SVM classifiers for different attack types.
- **Disadvantages:**
  - **Kernel Sensitivity:**
    - Performance heavily depends on the kernel function and parameter tuning, requiring significant experimentation.
  - **Imbalanced Data:**
    - While the model handled the dataset well overall, imbalanced classes (e.g., minority attack types like U2R) may still affect accuracy.
  - **Binary Classification:**
    - The system focused on binary classification (normal vs. abnormal), limiting its granularity in identifying specific attack types.



#### 4. Summary of Reviewed ML Solutions

Criteria	Enhanced Random Forest (ERF) with SMOTE & K-means	XGBoost Feature Selection with 5 ML classifiers (SVM, KNN, LR, ANN, DT)	Support Vector Machine (SVM) with RBF Kernel
Dataset Used	NSL-KDD (41 features)	UNSW-NB15 (42 features)	KDDCUP99 (41 features)
Accuracy	99.72% (training), 78.47% (testing)	90.85% (binary classification using DT), 77.51% (multiclass classification using ANN)	~95% (binary classification)
Strengths	<ul style="list-style-type: none"><li>- Effectively handles class imbalance with SMOTE and K-means for outlier detection.</li><li>- RF Tree selection improves diversity and performance.</li><li>- Achieves high training accuracy.</li></ul>	<ul style="list-style-type: none"><li>- Reduces feature dimensionality, improving efficiency.</li><li>- Strong generalization across binary and multiclass tasks.</li><li>- Supports integration with various ML models.</li></ul>	<ul style="list-style-type: none"><li>- Handles small, high-dimensional, and heterogeneous datasets well.</li><li>- Fast training and detection.</li><li>- Suitable for real-time applications.</li></ul>
Weaknesses	<ul style="list-style-type: none"><li>- Testing accuracy significantly lower than training, indicating overfitting.</li><li>- Computational overhead due to preprocessing (SMOTE) and similarity adjustments.</li></ul>	<ul style="list-style-type: none"><li>- Limited performance for minority classes due to data imbalance.</li><li>- Requires careful feature engineering and parameter tuning.</li></ul>	<ul style="list-style-type: none"><li>- Performance is sensitive to kernel selection and hyperparameters.</li><li>- Requires careful tuning in especially when the input dimension is greater than the number of examples</li></ul>
Optimization Approach	<ul style="list-style-type: none"><li>- SMOTE balances data.</li><li>- Decision trees filtered by performance and similarity.</li><li>- Feature compression</li></ul>	<ul style="list-style-type: none"><li>- XGBoost optimizes model by reducing features and focusing on high-importance attributes.</li></ul>	<ul style="list-style-type: none"><li>- Radial Basis Function (RBF) kernel and parameters (<math>C</math>, <math>\sigma^2</math>).</li></ul>
Best Use Case	<ul style="list-style-type: none"><li>- Severely imbalanced datasets requiring interpretability.</li><li>- Scenarios needing rule-based ensemble methods.</li></ul>	<ul style="list-style-type: none"><li>- High-dimensional datasets needing scalability and dimensionality reduction.</li><li>- Tasks requiring both binary and multiclass classification.</li></ul>	<ul style="list-style-type: none"><li>- Small datasets with heterogeneity.</li><li>- Real-time intrusion detection.</li></ul>

#### 5. Proposed ML Solution


To enhance the effectiveness of the IDS, the following combination of methods is recommended:

- **XGBoost for Feature Selection:** Efficiently identifies the most relevant features by providing robust feature importance scores.
- **SMOTE for Imbalance Handling:** Generates synthetic examples for the minority class to balance the dataset and improve model learning.
- **K-means for Clustering:** Groups similar data points to uncover underlying patterns and enhance classification performance.
- **Random Forest (RF) for Classification:** Offers robust classification with high accuracy and resilience against overfitting, leveraging the refined feature set.

This combination aims to maximize classification accuracy, generalization, and robustness, leveraging the strengths of each method for a comprehensive intrusion detection system.

#### 6. Evaluation of Machine Learning Models


Once the appropriate machine learning method has been determined, the next critical step is to evaluate the model's performance using relevant classification metrics. For our intrusion detection system, we will focus on the following performance criteria:

- 
1. **Accuracy:** Represents the number of correct predictions made by a model, divided by the total number of the dataset samples.
  2. **Precision:** Quantifies the reliability of a model's positive predictions, obtained by dividing the number of correct positive predictions by the total number of correct positive predictions made by the model.
  3. **Recall (Sensitivity):** Represents the number of correct positive predictions made by a model, divided by the total number of positives.
  4. **F1-Score:** A metric that combines the precision and recall metrics to assess a model's accuracy on a dataset. It is defined as the harmonic mean of precision and recall.
  5. **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):** A performance metric for classification problems. It represents the separability of classes, with a higher AUC indicating better distinction between malicious and non-malicious packets. The AUC is derived from the ROC curve, which shows the model's accuracy at different thresholds.

These metrics will help ensure that the model is not only accurate but also effective in identifying different attack types with minimal false positives and negatives.

## 7. Machine Learning Project Flow

Here's a summary of the machine learning pipeline, detailing the key steps from data preprocessing to model deployment and monitoring:

1. **Data Collection:** Gather and consolidate data from various sources, ensuring it is comprehensive and relevant for the classification task.
  2. **Data Preprocessing:** Cleanse the data to handle missing values, outliers, and noise. Normalize or standardize features as needed to ensure uniformity.
  3. **Feature Engineering and Selection:** Use domain knowledge and statistical methods (e.g., ANOVA, chi-squared tests) to select and engineer features that contribute most significantly to model performance.
  4. **Model Selection:** Utilize the ML Algorithm Selection Flowchart to choose the most suitable machine learning algorithms.
  5. **Model Training:** Train selected models on the preprocessed and feature-engineered dataset, using appropriate cross-validation techniques to avoid overfitting.
  6. **Model Evaluation:** Assess the models using performance metrics like Accuracy, Precision, Recall, F1-score, and AUC-ROC to determine the best-performing model.
  7. **Hyperparameter Tuning:** Fine-tune the chosen model's parameters to optimize performance further.
  8. **Model Deployment:** Implement the trained and tuned model in a production environment, ensuring it integrates seamlessly with existing systems.
  9. **Model Monitoring and Maintenance:** Continuously monitor the model's performance in the real world, retraining or updating it as needed to adapt to new data patterns and maintain accuracy.
- 



## References

Austalian Cyber Security Centre, 2021. *ACSC Annual Cyber Threat Report*, s.l.: s.n.

Barrios, A., 2022. *Machine Learning Algorithms Cheat Sheet*. [Online]

Available at: <https://www.accel.ai/anthology/2022/1/24/machine-learning-algorithms-cheat-sheet>

Brown, J., 2023. *educationdaily*. [Online]

Available at: <https://educationdaily.au/teachers/cybercrime-is-on-the-rise-is-your-schools-data-at-risk-3106/>

Henebery, B., 2023. *The Educator Australia*. [Online]

Available at: <https://www.theeducatoronline.com/k12/news/school-cybersecurity-in-2023-whats-your-incident-response-plan/282047>

Kasongo, S. & Sun, Y., 2020. Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. *Journal of Big Data*, p. 7:105.

Su, L., Bai, W., Zhu, Z. & He, X., 2021. Research on Application of Support Vector Machine in Intrusion Detection. *Journal of Physics: Conference Series*, p. 2037.

Wu, T. et al., 2022. Intrusion detection system combined enhanced random forest with SMOTE algorithm. *EURASIP Journal on Advances in Signal Processing*, p. 39.

