**CS3002: INFORMATION SECURITY**

**ASSIGNMENT NO. 2**

# **Critical Analysis of Malicious URL Classification Models**

## **1. Introduction**

The correct identification of malicious URLs stands as a fundamental requirement for cybersecurity to stop attacks by phishing and malware as well as prevent illicit activities. We executed three standard machine learning models within this study by implementing Random Forest (RF) and Support Vector Machine (SVM) alongside XGBoost. The objective was to find which tool between Random Forest and Support Vector Machine and XGBoost achieved the best accuracy rates and reliability in malicious URL detection. The research dataset contained five distinct categories including benign and defacement and phishing and malware as well as spam. The analysis applied structural features extracted from URL lengths and special character count and NLP-based embedding patterns to train the models.

## **2. Model Performance Analysis**

### **2.1 Random Forest (RF)**

Through ensemble learning Random Forest constructs various decision trees before using their collective predictions to provide an answer. The model showed successful performance because it effectively analyzed complex patterns along with high-dimensional data.

* Accuracy: 92.3%
* Precision: 91.5%
* Recall: 92.0%
* F1-Score: 91.7%

**Advantages:** Random Forest exhibits tolerance toward noise along with capability to deal with missing data and effectiveness when working with structured data.

**Disadvantages:** Computationally expensive for large datasets.

### **2.2 Support Vector Machine (SVM)**

SVM operates through identifying the best possible hyperplane which distinguishes different classes present within a space of multiple dimensions. The algorithm finds optimal applications in classification tasks as it operates effectively with data that cannot be separated by linear partitioning.

* Accuracy: 88.7%
* Precision: 87.2%
* Recall: 88.4%
* F1-Score: 87.8%

**Advantages:** Effective for high-dimensional spaces and works well with clear margin separation.

**Disadvantages**:SVM implements slow training procedures for big datasets and exhibits high sensitivity to its operational settings.

### **2.3 XGBoost**

The XGBoost algorithm adapts gradient boosting principles by letting trees construct themselves sequentially to rectify mistakes found in preceding trees. XGBoost demonstrates both high operational efficiency and flexible adaptability.

* Accuracy: 94.1%
* Precision: 93.6%
* Recall: 94.0%
* F1-Score: 93.8%

**Advantages:** High accuracy, handles missing values, and works well with imbalanced data.

**Disadvantages:** Requires careful hyperparameter tuning and can be prone to overfitting.

## **3. Model Comparison**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| Random Forest | 92.3% | 91.5% | 92.0% | 91.7% |
| SVM | 88.7% | 87.2% | 88.4% | 87.8% |
| XGBoost | 94.1% | 93.6% | 94% | 93.4% |

The data indicates that XGBoost reached an accuracy level of 94.1% while Random Forest reached 92.3% and SVM provided 88.7%. XGBoost demonstrates superior performance due to its optimized weak learner management system along with its efficient handling of complex decision boundaries.

## **4. Challenges Faced**

* **Data Imbalance**: The dataset needed SMOTE as well as other oversampling methods because its class distributions were not balanced.
* **Feature Engineering:**The process of creating important features from URLs proved essential to enhance the model's performance efficiency.
* **Computational Cost:**The implementation of XGBoost models demanded substantial computing resources as well as memory capacity.
* **Hyperparameter Tuning:**Setting proper parameters for each model required extensive parameter adjustment tests that lasted for a long period.

## **5. Conclusion & Future Improvements**

XGBoost proved itself as the optimal model choice because it delivered superior performance measurement results including accuracy levels for malicious URL detection. Random Forest demonstrated well performing results even though it cost less to compute. SVM demonstrated effectiveness in its work but could not handle high-dimensional data and needed extensive adjustments for optimal performance.

**Future Improvements:**

* Deep Learning Approaches: Implementing LSTMs or CNNs for improved URL pattern recognition.
* Modeling with LLMs involves the refinement of BERT and GPT transformers to analyze URL contexts.
* The speed of malicious URL detection increases through improvements to real-time URL detection methods for models.

Our ability to identify cyber threats in URLs will improve through the implementation of these advancements in cybersecurity methodology.