**Report of FedCSIS 2023 Challenge: Cybersecurity Threat Detection in the Behavior of IoT Devices**

**Problem Description**

The FedCSIS 2023 Challenge is designed to advance the field of cybersecurity by leveraging machine learning to identify and classify attack events within system activity logs. The challenge involves processing a large dataset of system events, each of which may or may not be part of a cyberattack. The goal is to preprocess this data, extract meaningful features, and develop a robust model to detect these attack events accurately.

**Data Features**

The dataset consists of multiple CSV files, each representing logs of system activities. The columns in the dataset include various attributes related to system calls, processes, user actions, and more. The key features in the data are as follows:

* **SYSCALL\_timestamp**: The timestamp of the system call event.
* **SYSCALL\_syscall**: The type of system call.
* **SYSCALL\_success**: Indicates if the system call was successful.
* **SYSCALL\_exit**: Exit code of the system call.
* **PROCESS\_comm**: Command name associated with the process.
* **PROCESS\_exe**: Executable path of the process.
* **PROCESS\_PATH**: Path related to the process.
* **CUSTOM\_openFiles**: Custom attribute for open files.
* **CUSTOM\_libs**: Custom attribute for libraries used.
* **PROCESS\_uid**: User ID associated with the process.
* **PROCESS\_gid**: Group ID associated with the process.
* **SYSCALL\_exit\_hint**: Hint for the system call exit status.
* **SYSCALL\_pid**: Process ID for the system call.
* **USER\_AUTH**: User authentication information.
* **CRED\_COUNT**: Credential count.
* **SYSTEM\_COUNT**: System count.
* **SERVICE\_COUNT**: Service count.
* **CUSTOM\_openSockets**: Custom attribute for open sockets.
* **USER\_ACTION\_op**: User action operation.
* **USER\_ACTION\_src**: Source of the user action.
* **USER\_ACTION\_res**: Result of the user action.
* **USER\_ACTION\_addr**: Address related to the user action.
* **PROCESS\_name**: Name of the process.
* **attack**: Label indicating if the event is part of an attack (1 for attack, 0 for normal).

**Preprocessing Steps Explanation:**

1. **Extracting Data:** This step involves unpacking the raw dataset from a ZIP file into a specified directory for further processing.
2. **Initial Data Exploration:** Here, we analyze the dataset's structure and content by determining the number of files and displaying the first few rows of the first file.
3. **Removing Single-Valued Columns:** Columns with only one unique value across all rows are removed as they do not contribute meaningful information for distinguishing between normal and attack events.
4. **Labeling Attack Files:** Each file is labeled to indicate whether it contains an attack event based on a predefined list of attack file names.
5. **Label Encoding:** Categorical columns are transformed into numerical values using label encoding to prepare them for machine learning algorithms.
6. **Handling Missing Values:** Missing values in categorical columns are filled with the string 'missing', while those in numerical columns are filled with the median value. Remaining missing values are filled with -1.
7. **Merging Data Files:** All processed and encoded files are merged into a single CSV file for easier analysis and modeling.
8. **Normalizing Data:** Data normalization is performed to ensure all features contribute equally to analysis by scaling them between 0 and 1.
9. **Handling Columns with Missing Values Post-Normalization:** Columns still containing missing values after normalization are removed from the dataset.

**Methods:**

This study aimed to assess the classification performance of four distinct machine learning models on a designated dataset. The dataset was initially subjected to preprocessing, wherein it was partitioned into feature variables (X) and the target variable (y). Subsequently, a standard train-test split strategy was employed, utilizing the train\_test\_split function from the scikit-learn library, with a test size of 20% and a fixed random state of 42.

Four diverse machine learning models were selected for evaluation:

1. XGBoost
2. Decision Tree
3. Random Forest
4. Logistic Regression

For each model, a set of common performance metrics was utilized to gauge its efficacy, encompassing:

* Accuracy
* Precision
* Recall
* F1 Score
* ROC AUC Score

Following training on the designated training set, predictions were generated for the test set. The performance metrics were subsequently computed based on the resultant predicted values vis-à-vis the actual target values.

**Results:**

The findings from the evaluation are encapsulated in the tabular representation below:

| **Model** | **Accuracy** | **F1 Score** | **Precision** | **ROC AUC** | **Recall** |
| --- | --- | --- | --- | --- | --- |
| Decision Tree | 0.997867 | 0.980950 | 0.990536 | 0.985495 | 0.971547 |
| Logistic Regression | 0.959593 | 0.491877 | 0.850826 | 0.671149 | 0.345933 |
| Random Forest | 0.996990 | 0.973074 | 0.984513 | 0.980495 | 0.961898 |
| XGBoost | 0.995235 | 0.956506 | 0.988328 | 0.963007 | 0.926669 |

* **Decision Tree:**

**Accuracy: 99.79%:** Decision trees can perform very well on certain datasets, especially when the data has clear decision boundaries.

**F1 Score: 98.10%:** The F1 score considers both precision and recall, making it a suitable metric for imbalanced datasets. Here, the high F1 score indicates a good balance between precision and recall.

**Precision: 99.05%:** Precision measures the proportion of true positives among all positive predictions. The high precision indicates that the model makes very few false positive predictions.

**ROC AUC: 98.55%:** ROC AUC measures the model's ability to distinguish between positive and negative classes. A high ROC AUC suggests that the model has good discrimination capability.

**Recall: 97.15%:** Recall measures the proportion of true positives that were correctly identified by the model. The high recall indicates that the model captures a large proportion of actual positive instances.

* **Logistic Regression:**

**Accuracy: 95.96%:** Logistic regression is a linear model that works well when the relationship between features and target variable is linear. However, it seems that the dataset might not be linearly separable, leading to slightly lower accuracy.

**F1 Score: 49.19%:** The F1 score is low, indicating poor balance between precision and recall. This could be due to the model's inability to correctly classify both positive and negative instances.

**Precision: 85.08%:** The precision is relatively high, suggesting that the model makes fewer false positive predictions compared to false negatives.

**ROC AUC: 67.11%:** The ROC AUC score is moderate, indicating that the model's ability to discriminate between positive and negative instances is not as strong as other models.

**Recall: 34.59%:** The recall is low, indicating that the model fails to capture a significant proportion of actual positive instances.

* **Random Forest:**

**Accuracy: 99.70%:** Random Forests are an ensemble of decision trees, which tend to perform well on various types of datasets.

**F1 Score: 97.31%:** The F1 score is high, indicating a good balance between precision and recall.

**Precision: 98.45%:** The precision is high, suggesting that the model makes very few false positive predictions.

**ROC AUC: 98.05%:** The ROC AUC score is high, indicating strong discrimination capability.

**Recall: 96.19%:** The recall is high, indicating that the model captures a large proportion of actual positive instances.

* **XGBoost:**

**Accuracy: 99.52%:** XGBoost is a powerful gradient boosting algorithm that often achieves high accuracy.

**F1 Score: 95.65%:** The F1 score is high, indicating a good balance between precision and recall.

**Precision: 98.83%:** The precision is high, suggesting that the model makes very few false positive predictions.

**ROC AUC: 96.30%:** The ROC AUC score is high, indicating strong discrimination capability.

**Recall: 92.67%:** The recall is high, indicating that the model captures a large proportion of actual positive instances.

Overall, the differences in performance among the models can be attributed to their underlying algorithms, the complexity of the dataset, and how well the models are able to capture the relationships between features and the target variable. Decision trees and ensemble methods like Random Forest and XGBoost tend to perform well on a wide range of datasets, while logistic regression may struggle with complex, nonlinear relationships.

**Conclusion:**

The present study embarked upon the evaluation of four distinct machine learning models—XGBoost, Decision Tree, Random Forest, and Logistic Regression—regarding their classification performance on a designated dataset. Through a meticulous methodological approach involving data preprocessing, train-test splitting, and metric-based assessment, notable insights into the efficacy of these models in a classification context were gleaned.

The findings of this investigation underscored the commendable performance of ensemble learning methods, specifically the Decision Tree and Random Forest algorithms. These models exhibited consistent and robust classification accuracy across diverse performance metrics, including Accuracy, Precision, Recall, F1 Score, and ROC AUC Score. The Decision Tree model, in particular, demonstrated remarkable prowess in accurately classifying instances within the dataset, indicating its suitability for classification tasks characterized by non-linear decision boundaries.

Conversely, the Logistic Regression model showcased comparatively diminished performance across various metrics, indicating potential limitations in handling complex classification tasks or capturing intricate relationships within the data. Despite its simplicity and interpretability, Logistic Regression may not suffice for datasets with non-linear or high-dimensional feature spaces.

The outcomes of this study not only contribute to the burgeoning field of machine learning methodology but also bear implications for practical applications across diverse domains. The discernment of model efficacy and performance nuances facilitates informed decision-making regarding model selection, deployment, and optimization in real-world scenarios. Moreover, the identification of ensemble methods as frontrunners underscores the significance of leveraging ensemble learning paradigms to harness collective intelligence and enhance predictive accuracy in classification endeavors.

Moving forward, avenues for further research abound, encompassing exploration into advanced ensemble learning techniques, fine-tuning of hyperparameters, and incorporation of domain-specific features or domainknowledge to bolster model performance. Additionally, longitudinal studies and cross-domain validation may offer deeper insights into model generalizability and robustness across varied datasets and application domains.

In summation, while the present study sheds light on the classification performance of diverse machine learning models, it also beckons the research community towards continued exploration, innovation, and refinement in the pursuit of advancing predictive modeling capabilities and addressing complex real-world challenges.

**Prepared By:**

Suat Deniz - Student ID: 2006102002

Berkay Caplık - Student ID: 2206102900

Abdulkadir Dağlar - Student ID: 2006102033