National College of Ireland

Configuration Manual

MSc Research Project
MSc in Cybersecurity

Ashwathy Ajaykumar Marath
Student ID: x23166371

School of Computing
National College of Ireland

Supervisor: Prof. Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Ashwathy Ajaykumar Marath		
Student ID:	X23166371		
Programme:	MSc In Cybersecurity	**Year:**	2024-2025
Module:	MSc Research Project		
Lecturer:	Prof. Arghir Nicolae Moldovan		
Submission Due Date:	12-12-2024		
Project Title:	Comparing the Capabilities of Ensemble Learning and SAST tools for Effective Code-Based Vulnerability Detection		
Word Count:	876	**Page Count:**	4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Ashwathy Ajaykumar Marath

Date: 12-12-2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Ashwathy Ajaykumar Marath
Student ID: x23166371

1 Libraries Imported

Data preparation, modeling, and assessment were achieved using major Python packages in the study. Pandas prepared and massaged the data and Matplotlib and Seaborn used it to visualize and explore the data. In machine learning models and features extraction (TF-IDF), scikit-learn was critical, as was in evaluation metrics (classification reports and ROC-AUC scores). SMOTE balanced the data using balanced-learn (Kothandapani, 2021). The supervisors also used the importing ensemble models of XGBoost in the predicting process, LightGBM, and Cat Boost enhanced the prediction results. By using counter for class distribution analysis. These libraries effectively used different algorithms and evaluated various datasets, satisfying the study objectives.

[1]:
```
import json
import pandas as pd
```

[2]:
```
# File path
file_path = "diversevul_20230702.json"

# Read JSON in chunks
data_list = []
with open(file_path, 'r') as f:
    for line in f:  # Reading the JSON line by line
        try:
            data_list.append(json.loads(line))  # Parse each line
        except json.JSONDecodeError as e:
            print(f"Error parsing line: {e}")
```

[3]:
```
# Convert list of dictionaries to DataFrame
df = pd.DataFrame(data_list)

# Display basic information and first few rows
print(df.info())
print(df.head())
```

Figure 1: Libraries imported

2 Methods used

This comparative research employed both quantitative and qualitative techniques. We performed an analysis of literature from 2016 to 2024 of vulnerability detection trends, challenges, and pragmatic progress through a qualitative technique. Quantitative analysis of SAST tool assessment and ensemble learning techniques was done using both VUDENC and

DiverseVul datasets. Random Forest, XG Boost, Light GBM and Cat Boost algorithms were applied for the ensemble models. Their imbalanced class problem was solved using SMOTE.

To compare the capabilities of SAST tools with ensemble models and evaluate their advantages and disadvantages as well as their potential vulnerability detection rate Bandit and SonarQube were used.

3 SonarQube

- **Download SonarQube:** Download the latest version (10.6.0.92116) from the official SonarQube website.
- **Install Java:** Ensure Java 17 or later is installed and properly configured in the system.
- **Extract SonarQube:** Unzip the downloaded package into a desired directory.
- **Configure Database:** Update the sonar.properties file to configure the database connection (e.g., MySQL, PostgreSQL).
- **Start SonarQube:** Navigate to the SonarQube directory and run the StartSonar.bat (Windows) or ./bin/sonar.sh start (Linux/Mac).
- **Access Dashboard:** Open http://localhost:9000 in a browser, log in with default credentials (admin/admin), and change the password.

Vulnerabilities has been analyzed for VUDENC using SonarQube. It generates large security, maintainability, and reliability reports with severity-level vulnerabilities and recommendations. This paper compared SonarQube to ensembles of learning methods and found that compared to ensemble learning, SonarQube excelled in the areas of low severity concerns and rule-based vulnerability detection while pointing out that its weakness in searching for complex patterns (Shatnawi *et al.*, 2024).

4 Bandit

Install Python: Ensure Python 3.x is installed on your system and added to the system path.

Install Bandit: Open a terminal or command prompt and run the following command:

pip install bandit

Verify Installation: Confirm the installation by running:

bandit --version

This should display the installed Bandit version.

Run Bandit: Analyze a Python project by navigating to its directory and executing:

bandit -r

Bandit analysed VUDENC for security issues related to Python scripts. While looking for usual mistakes such as wrong input processing, unsafe arrangements, and flaws, it generated many security reports. Severity and confidence-based approaches were compared with the ensemble learning techniques (Shatnawi *et al.*, 2024). Bandit pointed high-criticality issues, but the complex structure and continuously evolving risks were challenging to triage and close.

5 Software and Hardware Specifications

Software used for this paper are Anaconda 2.6.3, Jupyter Notebook for Model Evaluation and SAST tool 10.6.0.92116 and Bandit. The hardware specifications include i5 processor, Ram 8GB, 64bit OS Windows 11.

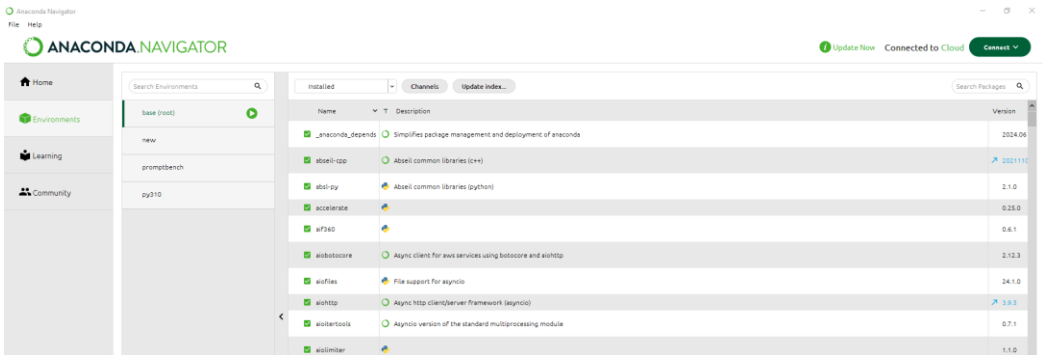

Figure 2: Anaconda Navigator

References

Kothandapani, H.P., 2021. A benchmarking and comparative analysis of python libraries for data cleaning: Evaluating accuracy, processing efficiency, and usability across diverse datasets. *Eigenpub Review of Science and Technology*, *5*(1), pp.16-33.

Shatnawi, A.S., Al-Duwairi, B. and Ala'A, S., 2024. Comprehensive Empirical Study of Python JWT Libraries. *Procedia Computer Science*, *238*, pp.827-832.