



Merging Intrusion Detection Systems

Our Team

Christopher Puig, Josue Camejo, Josue Naranjo, Dennis Martinez, Michael Simon



Background

The amount of zero-day attacks around the world has spiked in recent years. This increase has not just been zero-day attacks, but various forms of malware and viruses are being constantly updated and used by threat actors. In a perfect world, we can handle these attacks by using IDS (Intrusion Detection Systems) & IPS (Intrusion Prevention System). However, our systems are falling behind when it comes to detecting and preventing these attacks from affecting their network. Attackers are constantly upgrading and inventing new ways to obfuscate or manipulate data to be able to trick the IDS/IPS software into allowing them through.





PROJECT OVERVIEW

- Our primary objective was to develop a merged IDS solution by integrating multiple open source models. The specific goals include:
 1. Investigate the architecture and compatibility of the open source models to identify areas for integration and determine the viability of merging these systems.
 2. Evaluate the performance of each IDS model individually using standard metrics such as accuracy, recall, precision, and F1 score. This evaluation will provide insights into the strengths and weaknesses of each model in detecting network intrusions.
 3. Combine the two pre-trained IDS models using ensemble techniques such as Voting, Stacking, or Weighted Averaging. The aim is to leverage the diverse predictions of both models and create an ensemble IDS that potentially outperforms the individual models in accuracy and precision.





Tools Used

Python

- For Data Processing, to preprocess and clean the raw data before training the IDS models. Using implementations like Numpy.
- Model Training, allows for straightforward model training using specific algorithms.
- Visualization, libraries such as Matplotlib and Seaborn, allow for visualization of data, model performance and training history.

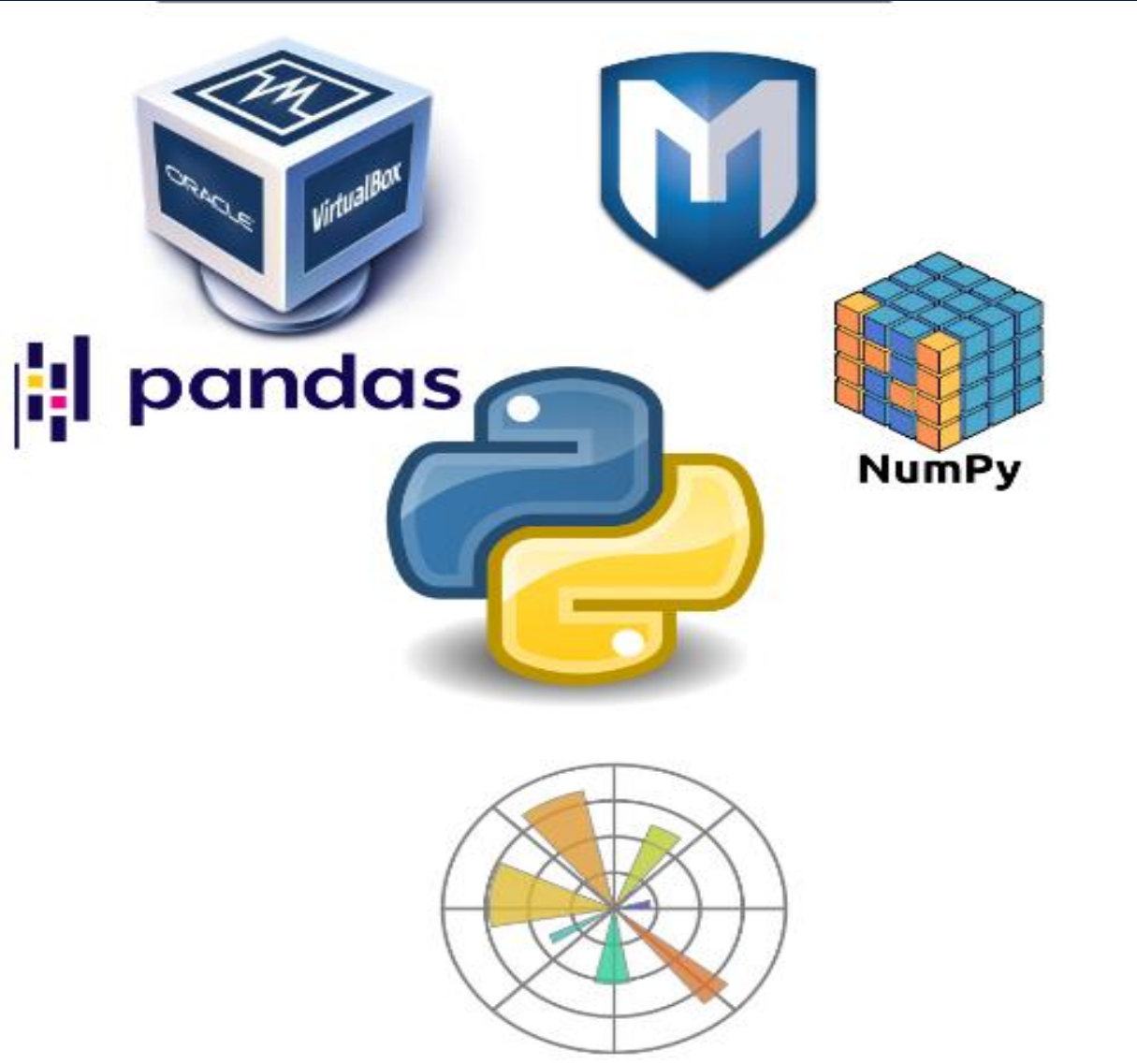
VirtualBox VM

- Kali Linux Virtual Machines & Windows 10 Virtual Machines for running tests and analyzing the models during live experiments on a network.
- Metasploit was used to test the accuracy using various exploits.

Open Source IDS models

- Intrusion Detection Systems using KDD99-Dataset





FIU

Engineering
& Computing

FLORIDA INTERNATIONAL UNIVERSITY



Obstacles before testing

1. Data Availability and Quality: Obtaining a suitable and reliable dataset for training and testing the IDS models.
2. Model Selection and Training: Selecting an appropriate machine learning algorithm for the individual IDS models is crucial.
3. Performance Metrics: Selecting appropriate evaluation metrics to assess the merged model performance is critical.



Obstacles during testing

1. Model Performance Discrepancies: The individual IDS models exhibited some variations in performance when combined into a merged model.
2. Error Analysis and Validation: Analyzing errors and misclassifications made by the merged IDS is crucial to identify weaknesses and improve its performance.

Model 1

```
7701/7720 [=====>.] - ETA: 0s - loss: 0.5105 - accuracy: 0.8030
Epoch 1: loss improved from inf to 0.51048, saving model to kddresults/dnn3layer/checkpoint-01.hdf5
7720/7720 [=====] - 7s 823us/step - loss: 0.5105 - accuracy: 0.8030
9720/9720 [=====] - 5s 480us/step
-----
accuracy
0.805
recall
1.000
precision
0.805
f1score
0.892
```



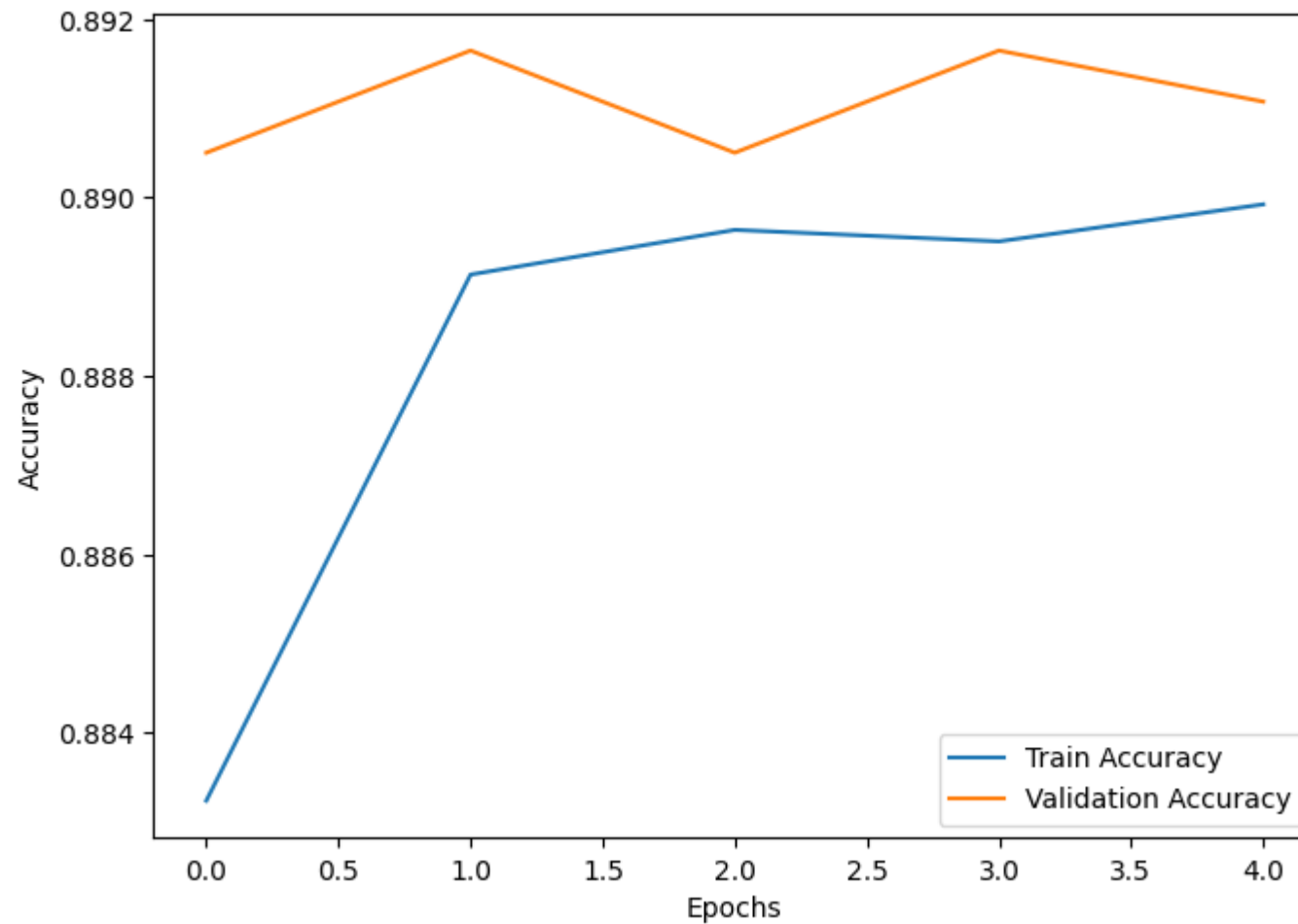
Model 2



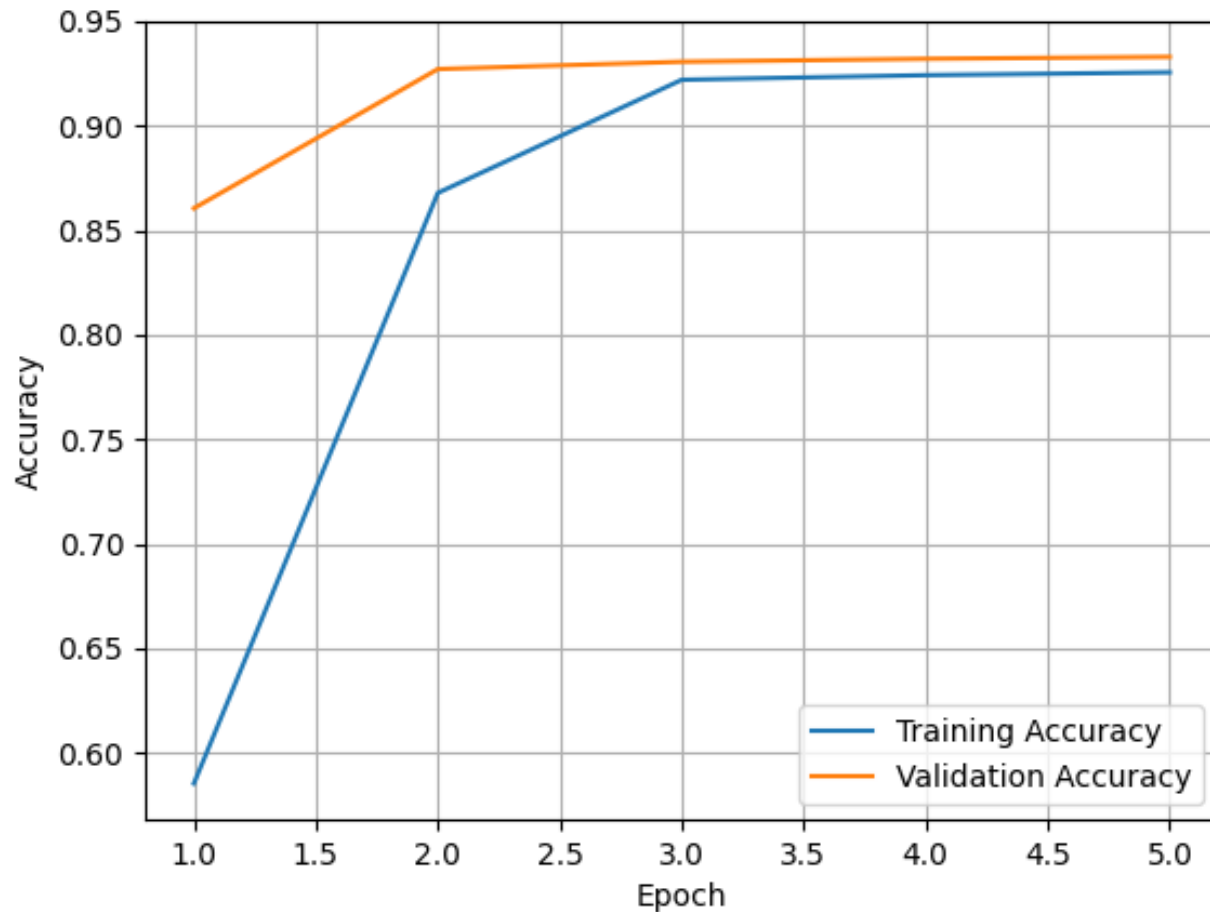
	precision	recall	f1-score	support
0	1.00	0.00	0.00	377
1	1.00	0.00	0.00	9
2	1.00	0.00	0.00	3
3	1.00	0.00	0.00	23
4	1.00	0.00	0.00	4
5	0.61	0.88	0.72	271
6	1.00	0.00	0.00	7
7	1.00	0.00	0.00	2
9	0.82	0.99	0.89	20660
10	1.00	0.00	0.00	74
11	0.95	0.88	0.92	35261
12	1.00	0.00	0.00	1
13	1.00	0.00	0.00	1
14	1.00	0.00	0.00	68
15	1.00	0.00	0.00	165
16	1.00	0.00	0.00	2
17	1.00	0.00	0.00	343
18	0.63	1.00	0.78	245
19	1.00	0.00	0.00	1
20	1.00	0.00	0.00	376
21	1.00	0.00	0.00	336
22	1.00	0.00	0.00	6
accuracy			0.89	58235
macro avg	0.96	0.17	0.15	58235
weighted avg	0.90	0.89	0.88	58235



Model 2



Model3 (Merged Model)



Model3 Training Algorithm

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, classification_report
from keras.models import load_model
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
import keras # Import keras explicitly for the callback

# Load the trained models
model1 = load_model('kddresults/dnn3layer/dnn3layer_model.hdf5')
model2 = load_model('kddresults/testing/test_model.hdf5')

# Load the kddcup99 dataset
testdata = pd.read_csv('https://raw.githubusercontent.com/raahulvigneswaran/Intrusion-Detection-Systems/master/dnn/kdd/binary/Testing.csv',

# Prepare the dataset for evaluation
X_test = testdata.iloc[:, :-1].values # Features
y_test = testdata.iloc[:, -1].values # Labels

# Convert labels to integer type
y_test = y_test.astype(int)

# Standardize the input features
scaler = StandardScaler()
X_test = scaler.fit_transform(X_test)

# Create a validation set to tune the ensemble weights
X_train, X_val, y_train, y_val = train_test_split(X_test, y_test, test_size=0.3, random_state=42)

# Convert labels to integer type
y_val = y_val.astype(int)
```





Findings

- **Enhanced Accuracy:** The merged IDS demonstrated a remarkable increase in accuracy compared to both individual models. The combination of diverse predictions from the individual models led to a more robust and reliable detection of network intrusions, resulting in significantly higher accuracy.
- **Improved Precision:** Precision, which measures the proportion of true positive predictions among all positive predictions, was notably enhanced in the merged IDS. This improvement indicated that the merge was highly effective in correctly identifying intrusions while minimizing false positives, leading to a more precise classification of network threats.
- **Scalability and Efficiency:** Despite the complexity introduced by the merged approach, the merged model remained scalable and efficient. It demonstrated manageable computational overhead, making it feasible for implementation in larger network infrastructures.

Conclusion

In conclusion, the Merged Intrusion Detection System (IDS) project successfully demonstrated the power of merging two individual IDS models using ensemble techniques to enhance network security. Through rigorous implementation and testing in Python, the project revealed that the combined model surpassed the accuracy and precision of the individual models.



Questions



FIU

Engineering
& Computing

FLORIDA INTERNATIONAL UNIVERSITY