Flow Anomaly Based Intrusion Detection System for Android Mobile Devices

Panagiotis I. Radoglou-Grammatikis

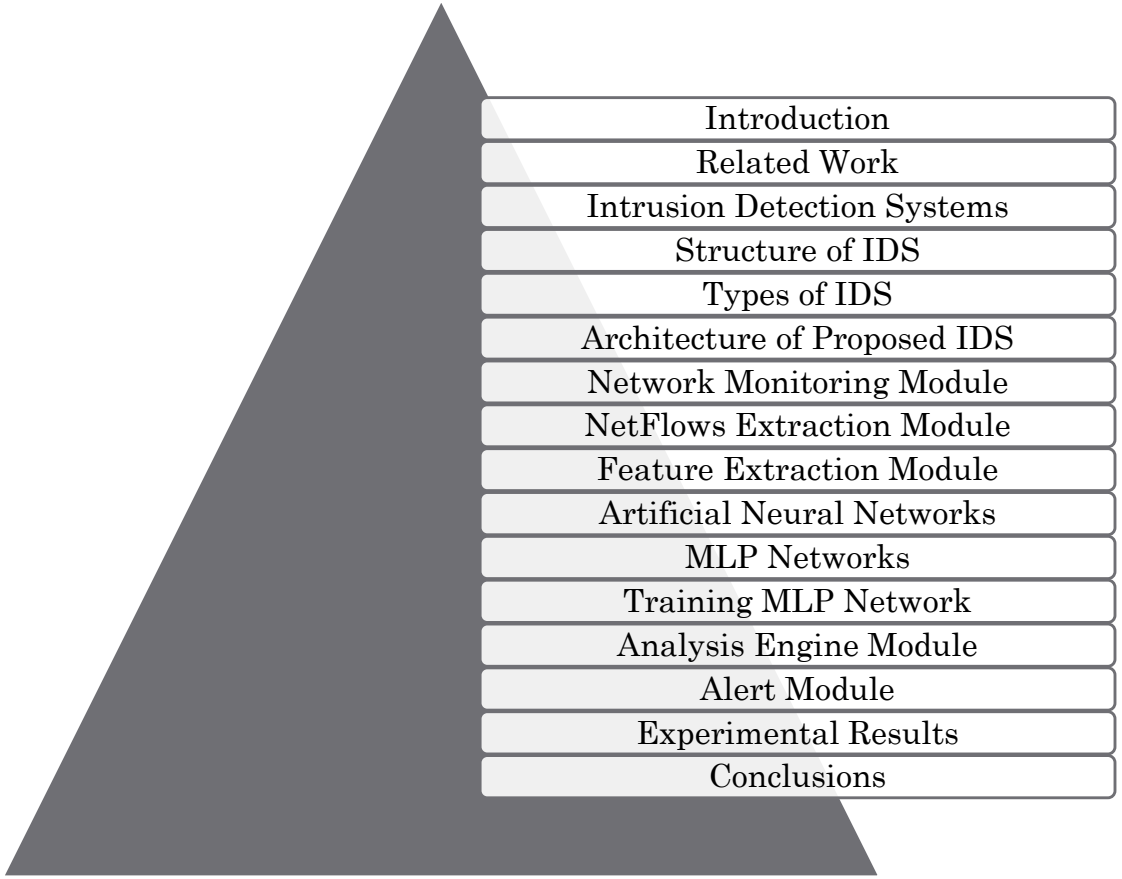
Panagiotis G. Sarigiannidis

University of Western Macedonia, Kozani, Greece

MOCAST

International Conference on Modern Circuits and Systems Technologies

Outline


Introduction
Related Work
Intrusion Detection Systems
Structure of IDS
Types of IDS
Architecture of Proposed IDS
Network Monitoring Module
NetFlows Extraction Module
Feature Extraction Module
Artificial Neural Networks
MLP Networks
Training MLP Network
Analysis Engine Module
Alert Module
Experimental Results
Conclusions

Introduction

- Over the last few years, mobile devices have gained increasing popularity due to the variety of the data services they offer (e.g. Internet, e-mailing, gaming).
- However, the expansion of the mobile devices comes with the increase of size of information that is processed, resulting in increasing the potential targets stemming from an everincreasing number of various security threats.
- We will present a light-weight Intrusion Detection System (IDS) for Android mobile devices, which can detect anomaly NetFlows.

Related Work

- A wide range of security components appears in the market in response to security issue of mobile computing systems.
- However, most of the mobile security software focus on the traditional PC security-based approaches and don't take into consideration the unique characteristics of the mobile devices, such as limitations in terms of CPU, memory and battery power.
- A new kind of IDS for mobile devices is proposed which takes into consideration the CPU, the memory, and the power consumption of the mobile devices.

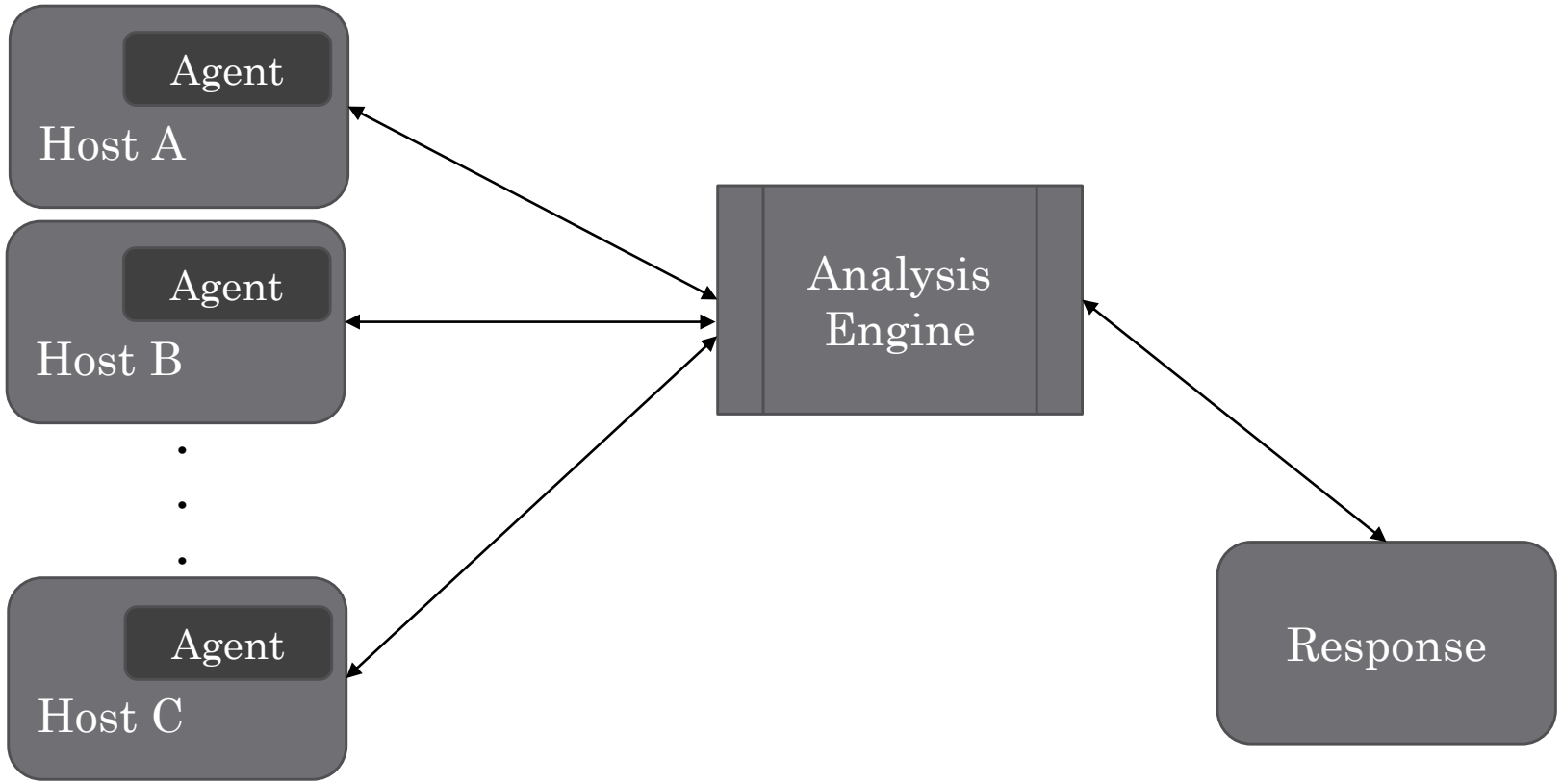
Intrusion Detection Systems

The main goal of an effective IDS is to provide high rates of attack detection with very small rates of false alarms.

Main characteristics:

- Identifying a wide range of intrusions
- Timely intrusion detection
- High accuracy rate
- Friendly user interface

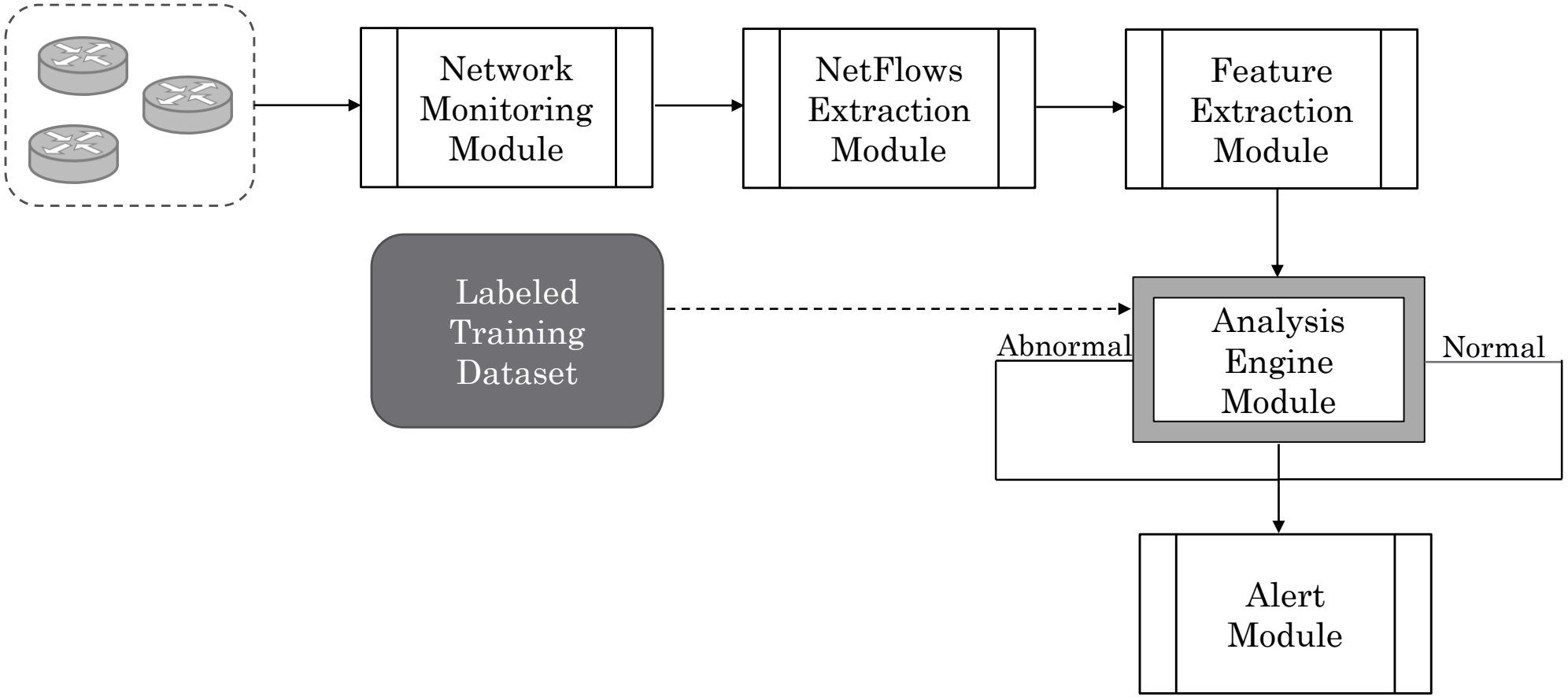
Structure of IDS



Types of IDS

- **Signature-based IDS** - recognize known attacks
  - Define a set of attack signatures
  - Detect actions that match a signature
  - Add new signatures often
  - Examples: ARMD, ASIM, Bro, CSM, CyberCop, GRIDS, Stalker, Tripwire
- **Anomaly-based IDS** - recognize anomaly behavior
  - Define a set of metrics for the system
  - Build a statistical model for those metrics during “normal” operation
  - Detect when metrics differ significantly from normal
  - Examples: AAFID, MIDAS, NADIR, UNICORN

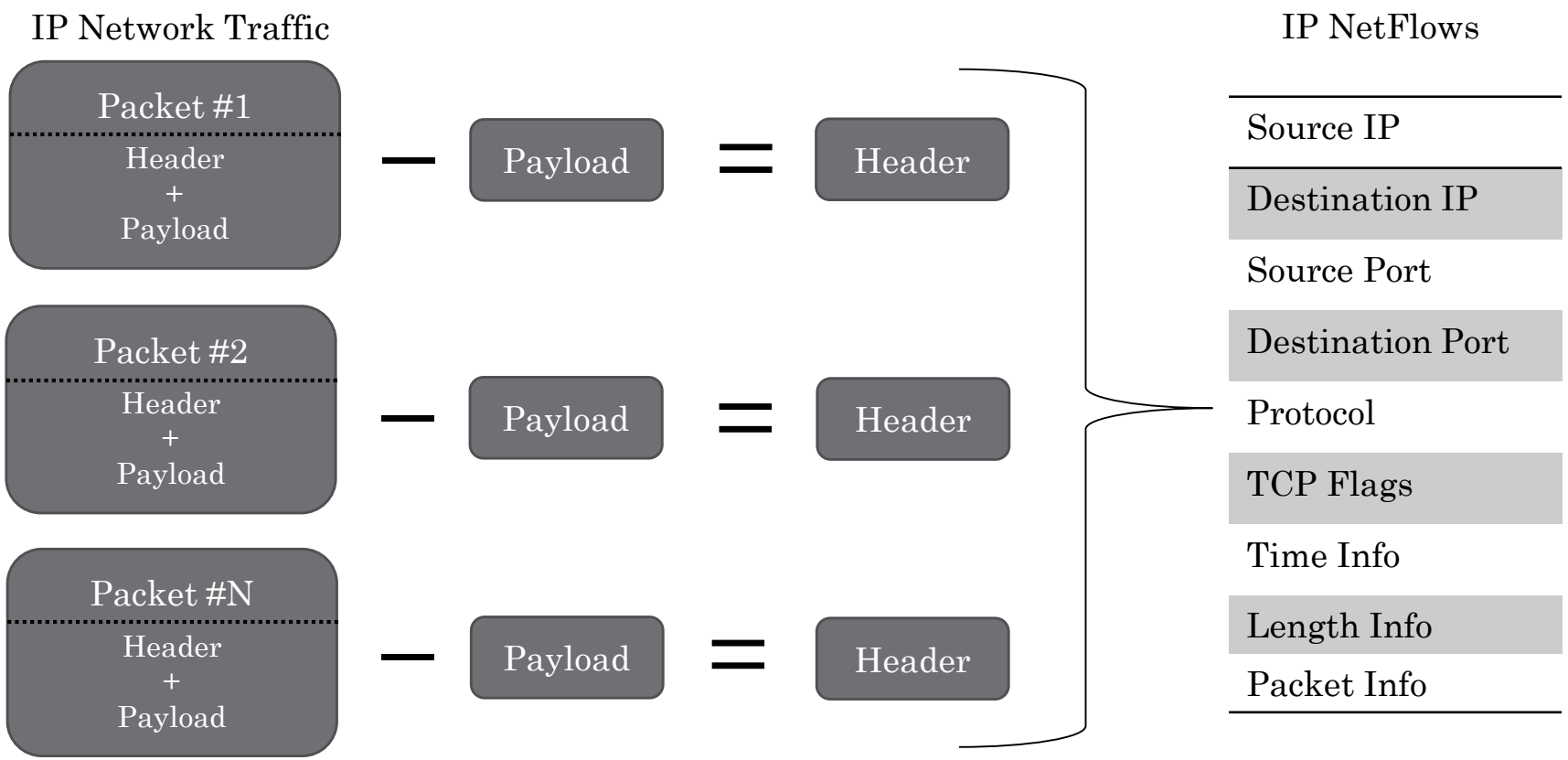
Architecture of Proposed IDS



Network Monitoring Module

The main operation of this module is to monitor the network information. When the user enables the operation of this module is created an Android service, which continually captures and analyses the network traffic. To this end, we utilized the Scapy library.

NetFlows Extraction Module



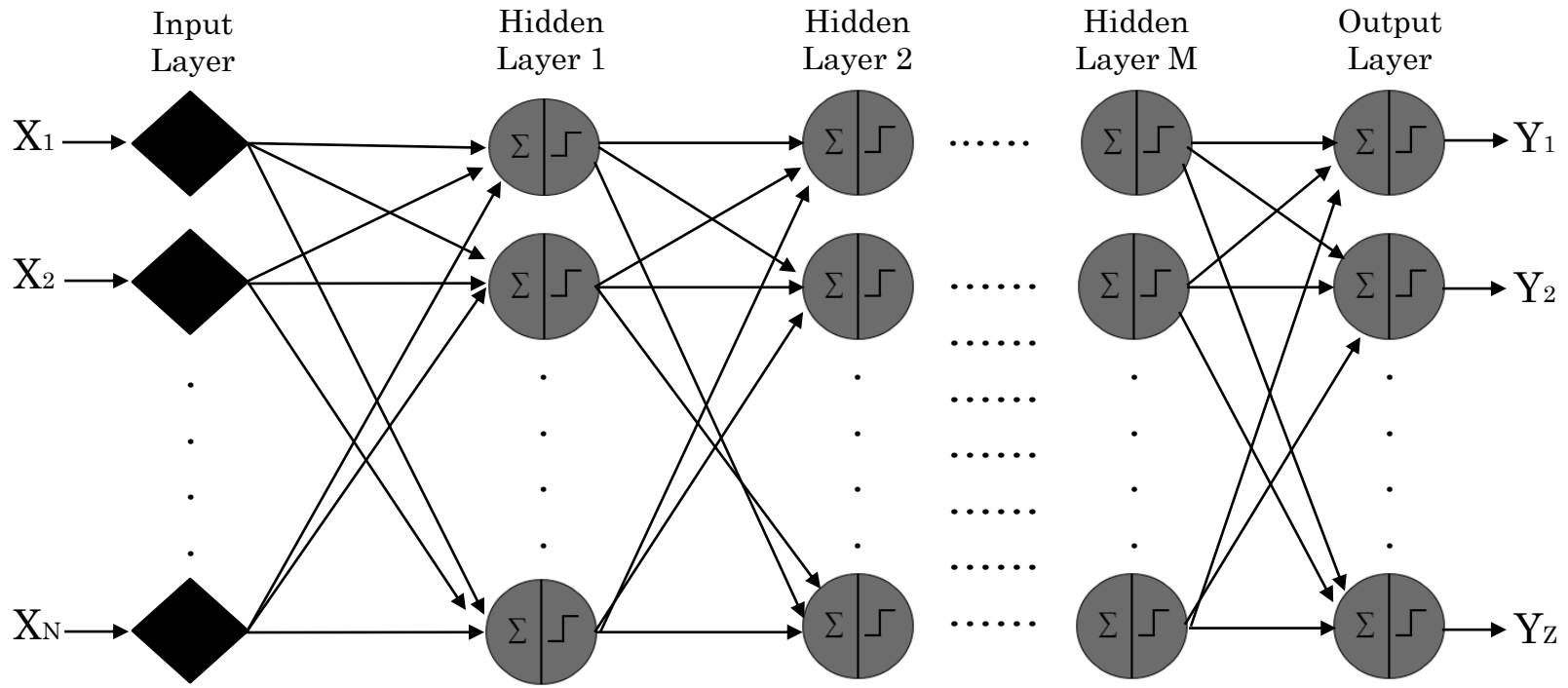
Feature Extraction Module

- **Average NetFlow Size**: It provides a useful hint for anomalous events, such as port scan.
- **Average Packet Number**: One of the main features of DoS attacks is the source IP spoofing, which makes the task of tracing attacker's true source very difficult. A side effect is the generation of flows with a small number of packets, i.e., about 3 packets per flow. This differs from normal traffic that usually involves a higher number of packets per flow.
- **NetFlow Duration**: The duration of a NetFlow can indicate many kinds of intrusions, such as worms, Botnets, and DoS attacks.
- **Average Packet Size**: Low average size can be a sign of anomaly. For example, in TCP flooding attacks packets of 120 bytes are typically sent.

Artificial Neural Networks

- An artificial neural network (ANN) is an information-processing model which consists of individual neurons.
- Each neuron is fundamentally a summing element followed by an activation function.
- The output of each neuron is fed as an input to all of the neurons in the next layer.
- During training, the neural network parameters are optimized to associate outputs with corresponding input patterns.
- When an ANN is applied, it processes the input patterns and tries to output the corresponding class.
- In our approach, we utilized a multi-layer Perceptron (MLP). MLP is a layered feed forward ANN which typically trained with back propagation algorithms.

MLP Networks

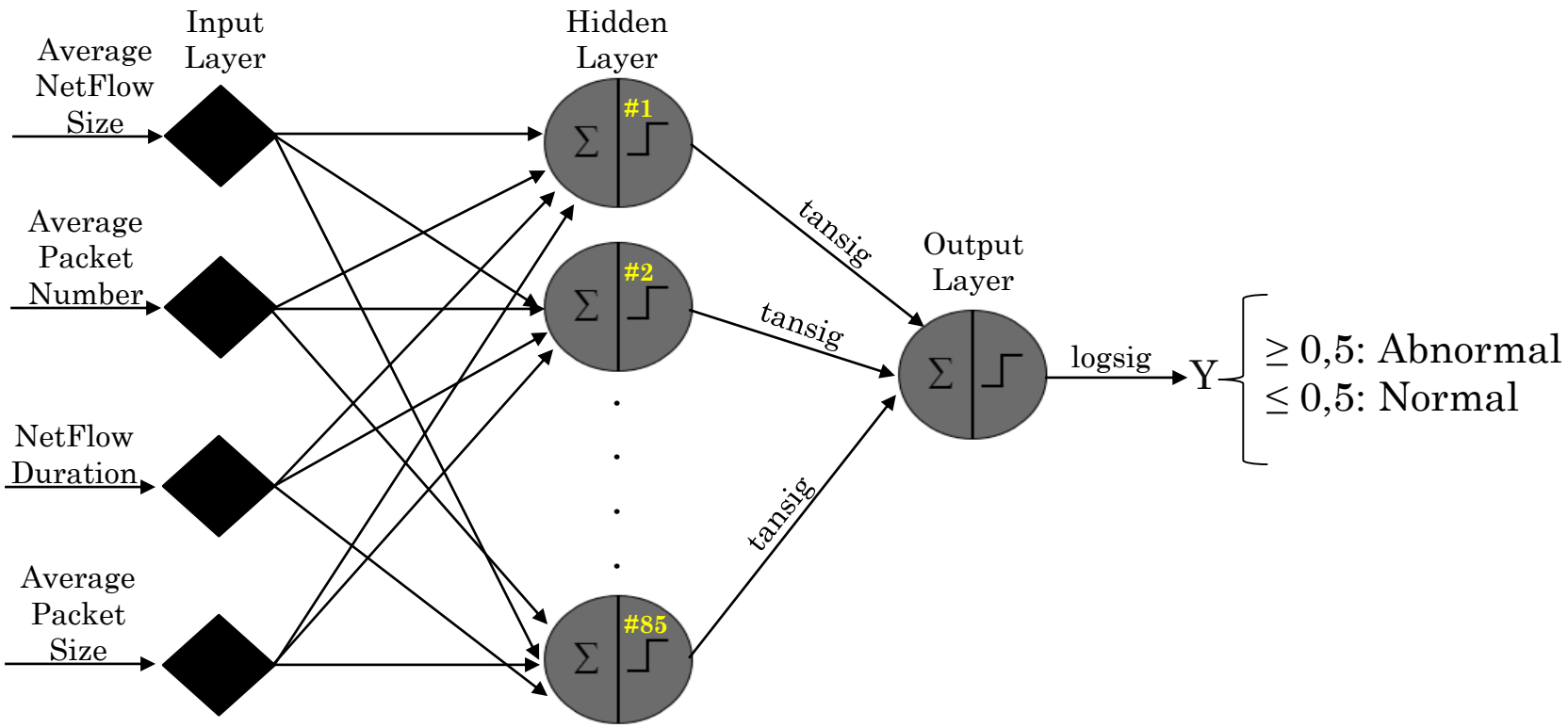


Training MLP Network

Total NetFlows	Normal NetFlows	Abnormal NetFlows
806132	361433	444699

- Levenberg-Marquardt
- Mean Squared Error $\left(MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - y_i)^2\right)$
- CTU-13 Dataset
- MATLAB Neural Network Toolbox

Analysis Engine Module



Alert Module

This module runs at the final stage of the proposed system. It demonstrates the characteristics of the NetFlows which exhibited abnormal behavior.

Experimental Results – 1/2

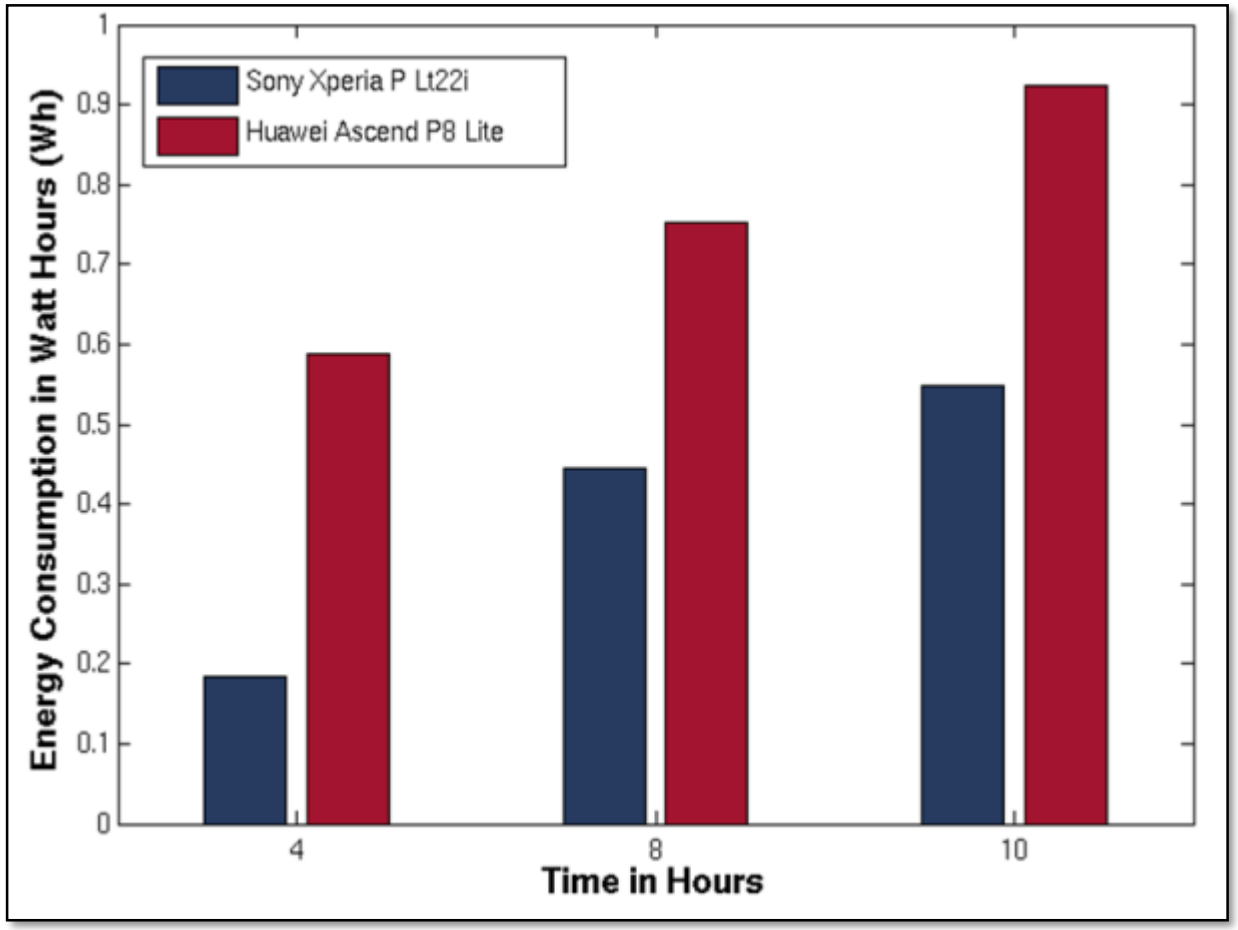
Mobile devices that we utilized:

- Sony XPERIA P LT22i (2x ARM-Cortex-A9 processor at 1.00 GHz, Android version 4.1.2, Li-Ion 1305 mAh battery)
- Huawei Ascend p8 Lite (8x ARM-Cortex-A53 at 1.2 GHz, Android version 5.0.1, Li-Ion 2200 mAh battery)

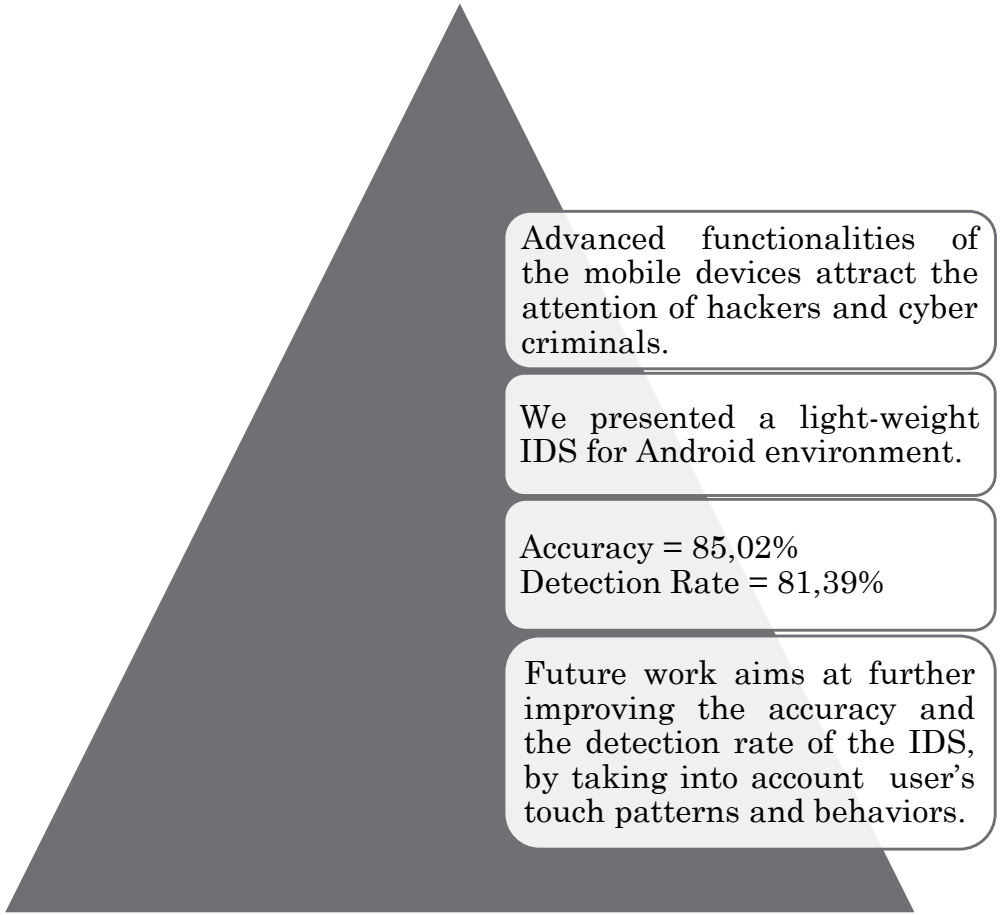
$$\textbf{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} = 85{,}02\%$$

$$\textbf{Detection Rate} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 81{,}39\%$$

Experimental Results – 2/2



Conclusions



Thank you!
Questions?