Arhuaco: Deep Learning and Isolation Based Intrusion Detection in High Energy Physics

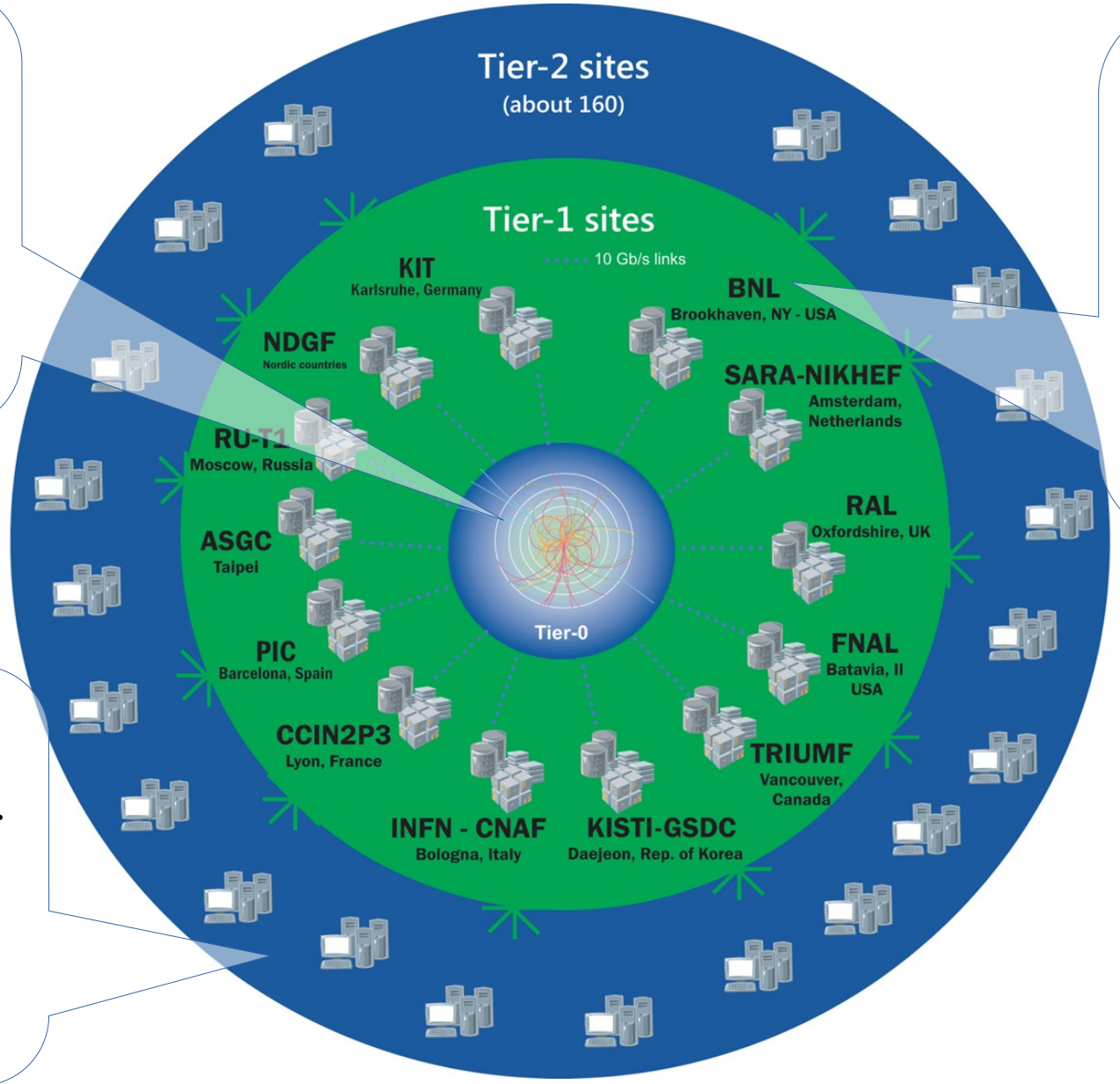
Dr. Andrés Gómez Ramírez, Prof. Dr. Udo Kebschull
IRI - Goethe-Universität Frankfurt am Main

Symposium on Artificial Intelligence for Science, Industry and Society 2019, Mexico City, Mexico.

Grid computing, LHC and the ALICE grid

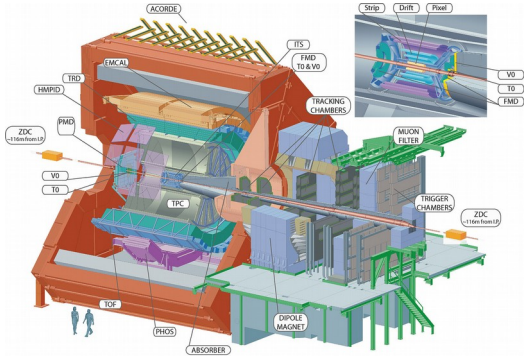
The Large Hadron Collider (LHC) produces particle collision data that is distributed around the world for scientific community analysis.

Grid users can send applications or jobs to process distributed data. A user's application can be executed in any available server in a cluster.



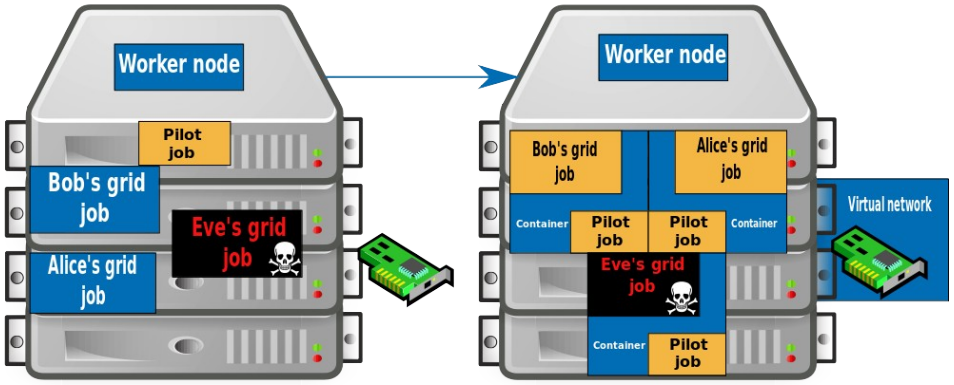
Grid – high performance computing clusters working together in a network, commonly for science. ALICE grid: centralized control, configuration and monitoring.

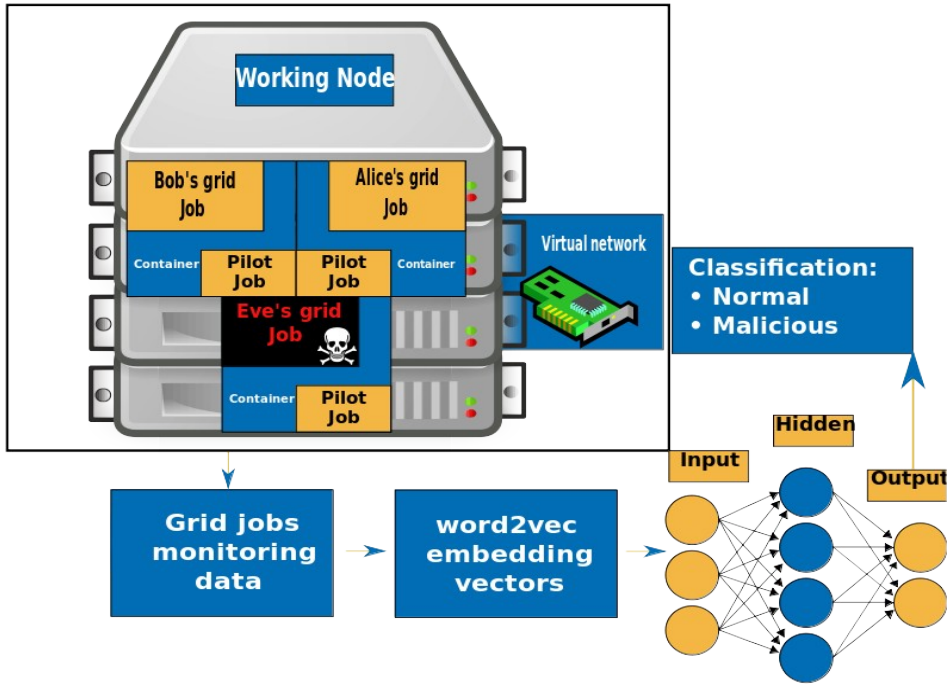
ALICE: A Large Ion Collider Experiment.



WLCG image: http://wlcg.web.cern.ch/documents-reference

Goals of the project





Problems:

- Users can execute any application: **arbitrary code execution by design**.
- Payloads are frequently executed directly on host Operating System.
- Network sections are **shared**.
- **Hundreds of thousands** of jobs running simultaneously.
- Expensive to have many security experts **monitoring the Grid**.

Solutions:

- Security by **isolation**.
- Isolation for **extracting** of monitoring data.
- Automated intrusion **detection** and **prevention**.

Traditional Solutions: Intrusion detection and prevention systems (IDPS), and virtual machines (VM)

IDPS

- Based on static rules.
- Previously known attacks.
- Need to be manually update by human experts.
- Cannot be automatically adapted to new environments.

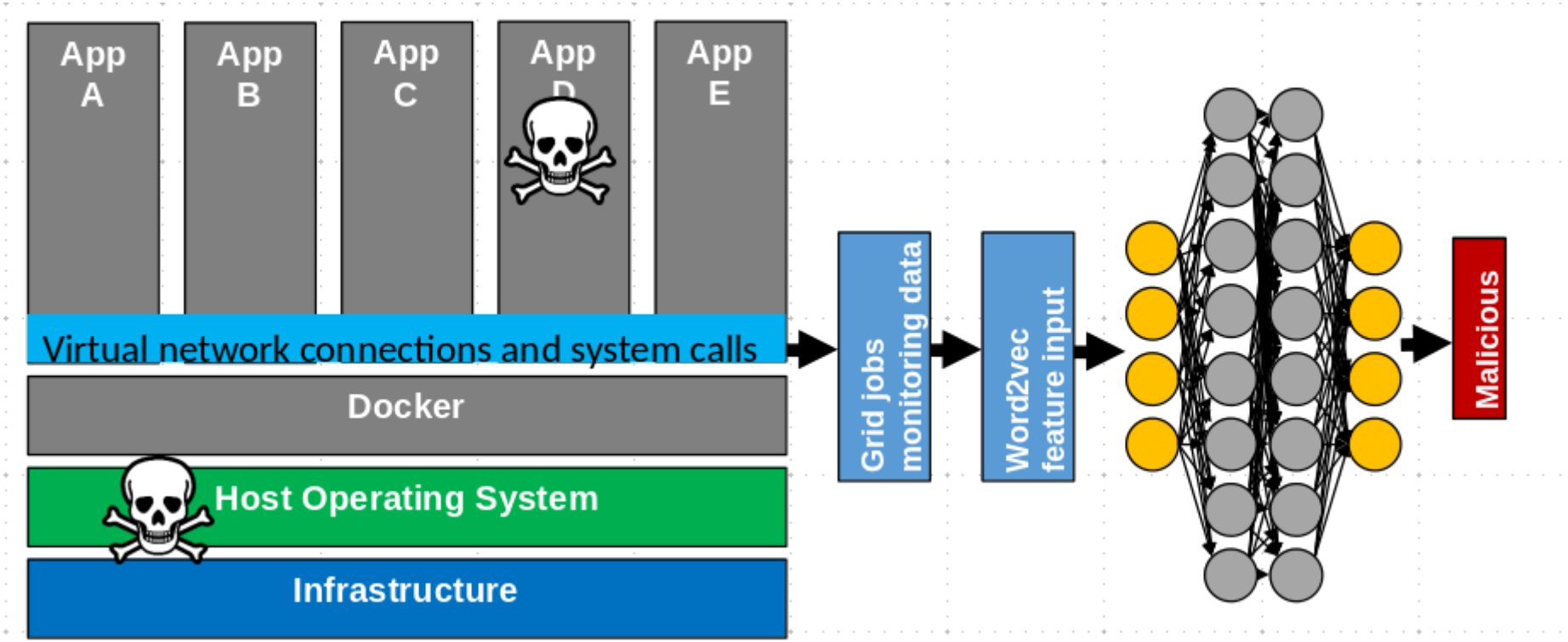
The Zeek Network Security Monitor

Virtual Machines

- Full or partial hardware emulation.
- They consume many resources.
- Not practical to run a job per VM.
- Jobs are not isolated from each other.

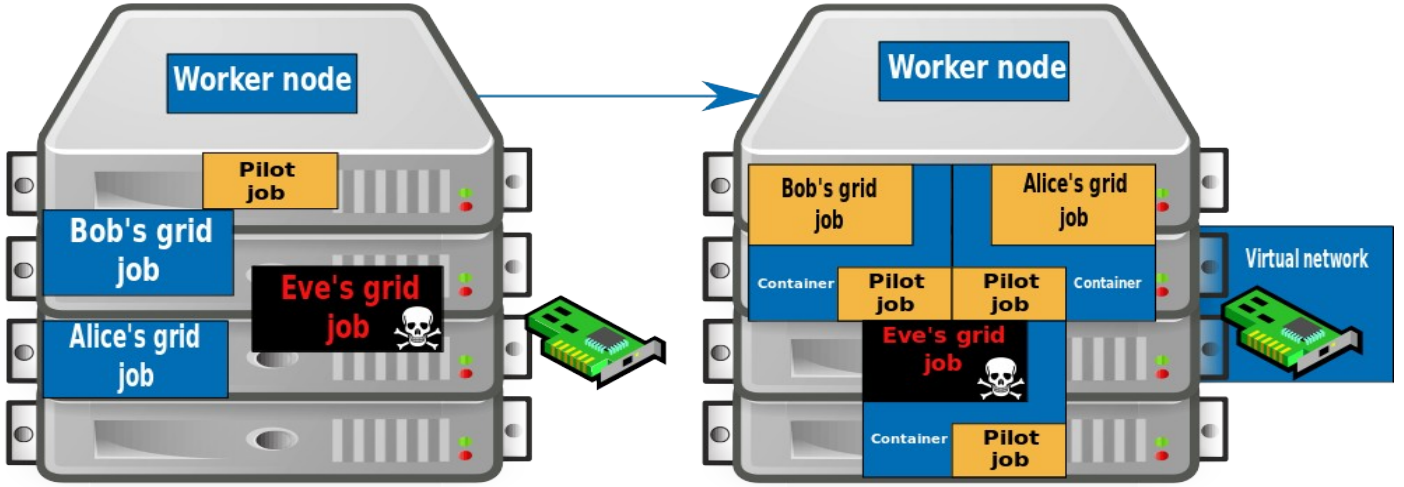
Arhuaco: A Grid computing Security Monitoring and Isolation framework

- Linux Containers for Isolating jobs
- **Deep Learning** for intrusion detection → Grid Jobs – **normal vs malware**
- **Convolutional Neural Networks**
- **Recurrent Neural Networks**
- **Generative** models for improving training

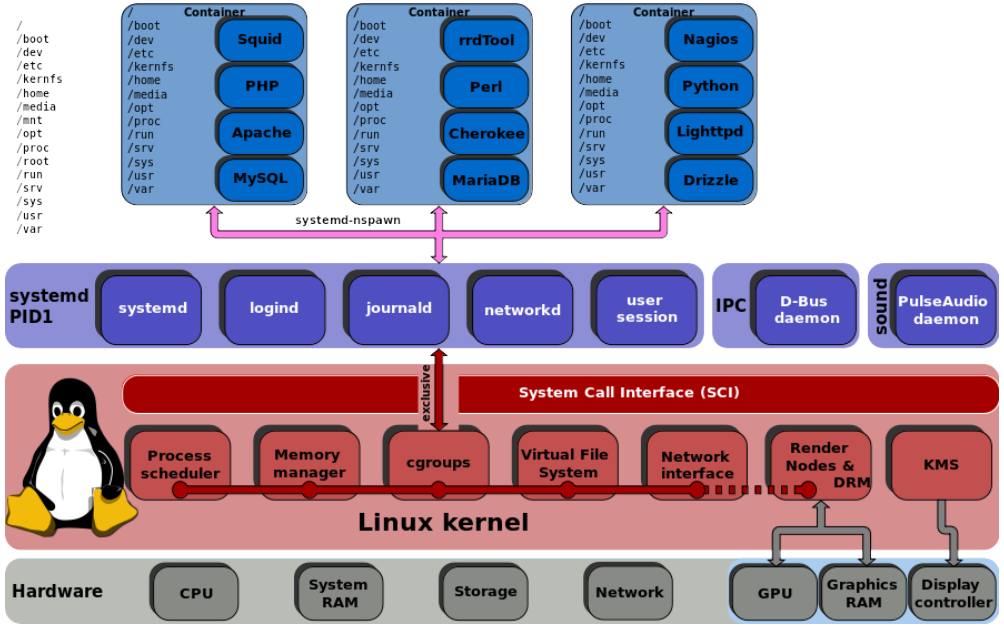


Proposed solution: 1. Grid Job execution and network isolation

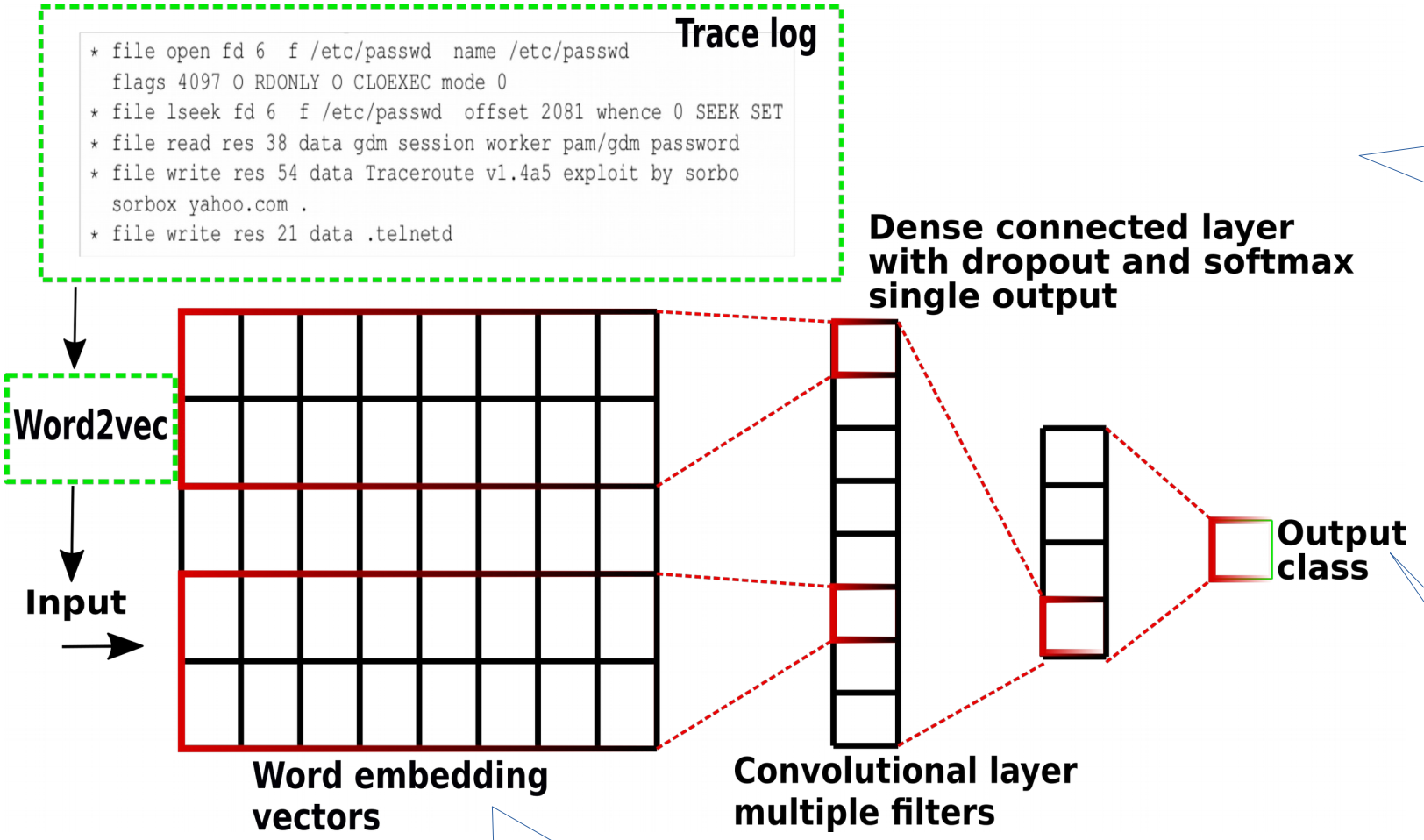
Grid jobs are executed inside containers for isolation among the underlying system and other jobs.



Linux containers are a lightweight alternative to virtual machines. Processes are executed over the same kernel.



Proposed Solutions: 2. Word2vec and convolutional neural networks for grid job classification



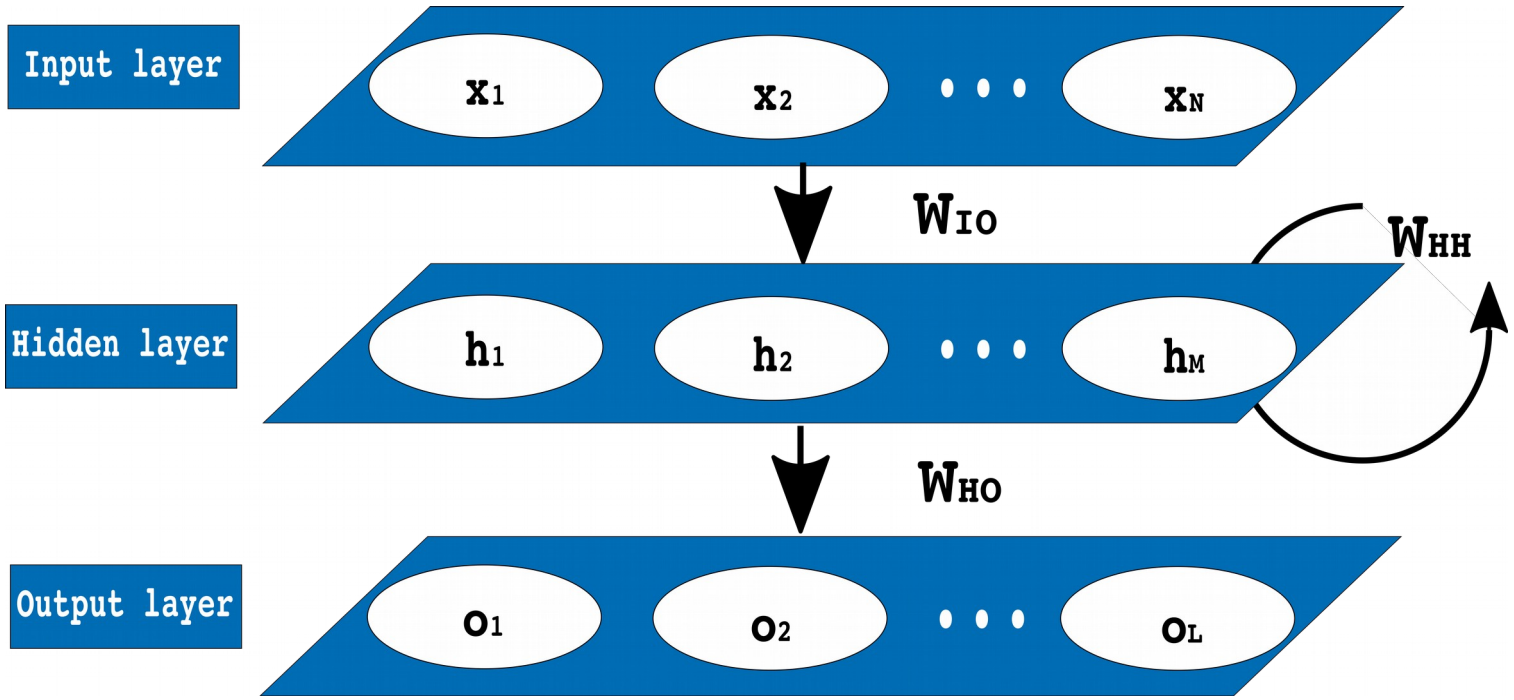
The input grid monitoring data is used as text. It is preprocessed with word2vec to transform words into vectors of numbers.

The created vectors are the input for the convolutional neural network.

Finally the neural network output is the class of the gird job trace: normal or malicious.

Proposed Solutions 3: Long short-term memory (LSTM) for synthetic data generation:

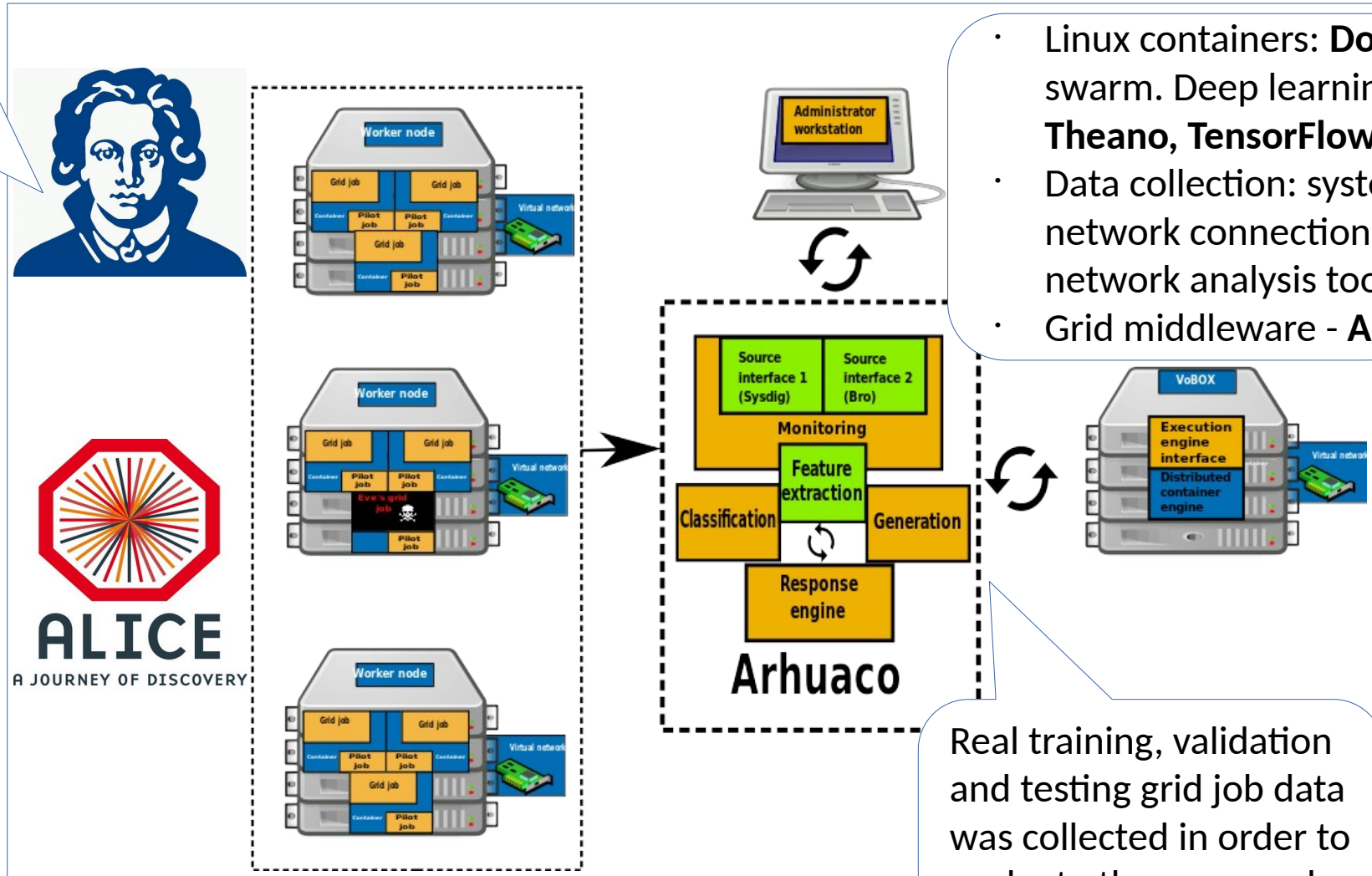
LSTMs learn a model from the input data and can generate new data similar to the input one.



They provide a feedback loop in the hidden layers that allows them to remember long term relationship between the input data.

Implementation: Arhuaco, a proof-of-concept framework

UF (Goethe-Universität Frankfurt am Main) is a testing grid site, connected to the ALICE grid, utilized for testing the PhD project.



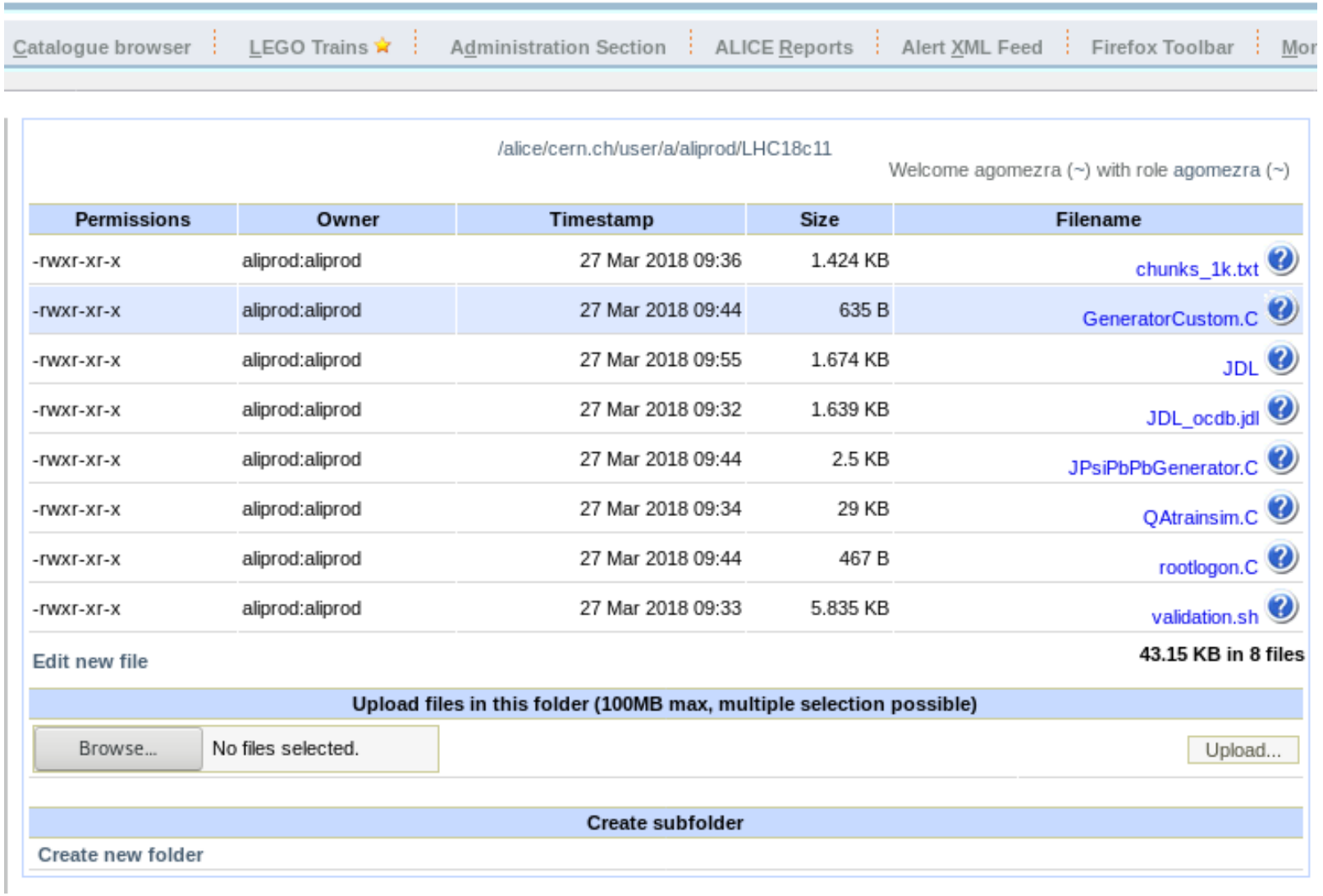

```
# Build the model
# Graph subnet with one input and one output,
# convolutional layers concateneted in parallel
graph_in = Input(shape=(sequence_length,embedding_dim))
convs = []
for fsz in filter_sizes:
    # Conv1D: keras convolutional layer
    # Embedding: it allows to use word vectors as inputs
    conv = Conv1D(activity_regularizer=l2(
                                regularizer_param),
                  padding="valid",
                  strides=1,
                  kernel_regularizer=l2(
                              regularizer_param),
                  filters=num_filters,
                  activation="relu",
                  kernel_size=fsz)(graph_in)
    pool = MaxPooling1D(pool_size=pool_size)(conv)
    flatten = Flatten()(pool)
    convs.append(flatten)
out = None
if len(filter_sizes) > 1:
    out = Concatenate()(convs)
else:
    out = convs[0]
graph = Model(outputs=out, inputs=graph_in)
self.model = Sequential()
self.model.add(Embedding(len(self.vocabulary)+1,
                embedding_dim,
                input_length=sequence_length,
                weights=self.embedding_weights))
```

- Linux containers: **Docker**, Docker swarm. Deep learning: **Keras, Theano, TensorFlow**, python 3.
- Data collection: system calls - sysdig, network connection - The **zeek** network analysis tool.
- Grid middleware - **ALICE AliEn**.

Real training, validation and testing grid job data was collected in order to evaluate the proposed solutions.

Evaluation of Arhuaco

MonALISA Repository for ALICE



Evaluation of Arhuaco

virustotal

SHA256: 5ff86d434be5a4011ddcd63b1dcf1ebb0b72ad9e27bfccf640f38dc117cf330d

File name: 341dcb650048862fe07cb53fba4a76fffe9bcd7e_86.tgz

Detection ratio: 21 / 53

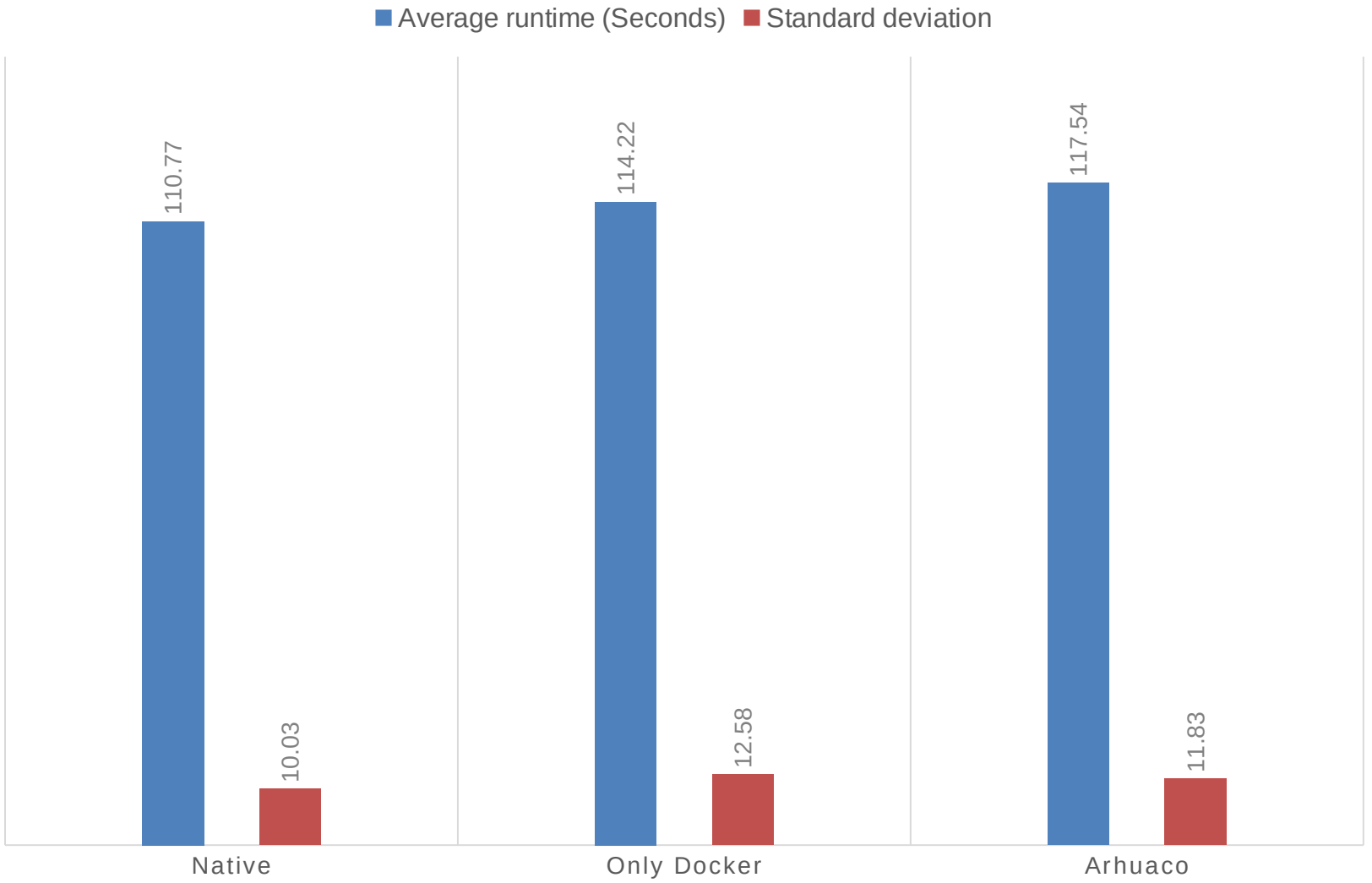
Analysis date: 2014-07-22 17:47:44 UTC (3 years, 9 months ago)

Analysis | Additional information | Comments 0 | Votes

Antivirus	Result	Update
Ad-Aware	Application.Linux.BitCoinMiner.A	20140722
AntiVir	LINUX/Procfake	20140722
Avast	ELF:BitCoinMiner-G [Tool]	20140722
BitDefender	Application.Linux.BitCoinMiner.A	20140722
CAT-QuickHeal	Linux.RiskTool.BitCoinMiner.a	20140722
Comodo	UnclassifiedMalware	20140722
DrWeb	Linux.CpuMiner.1	20140722
ESET-NOD32	Linux/BitCoinMiner.D	20140722
F-Secure	Application.Linux.BitCoinMiner	20140722

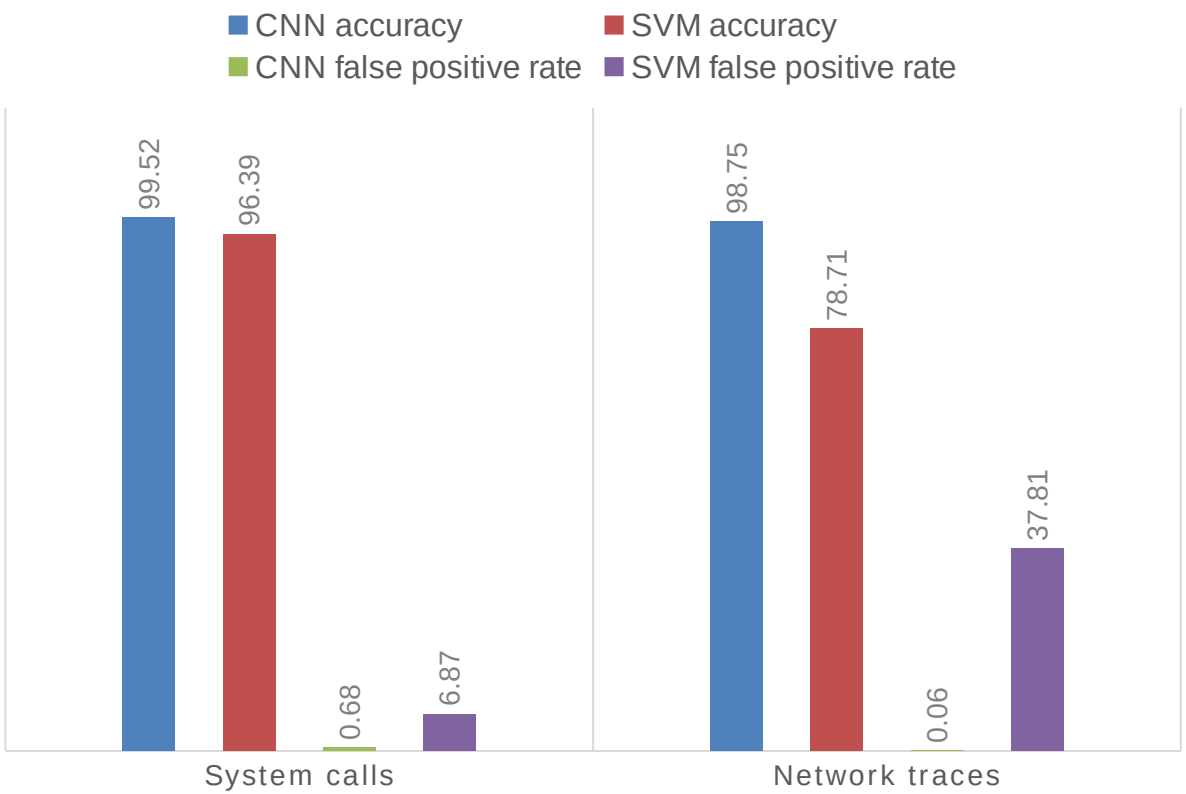
Results of the Arhuaco evaluation: Performance impact

- Same **ALICE grid job** executed **1600** times.
- Runtime measured in **seconds**.
- **Native:** jobs running over the host operation system.
- **Docker:** jobs running inside containers.
- **Arhuaco:** jobs running isolated and monitored.



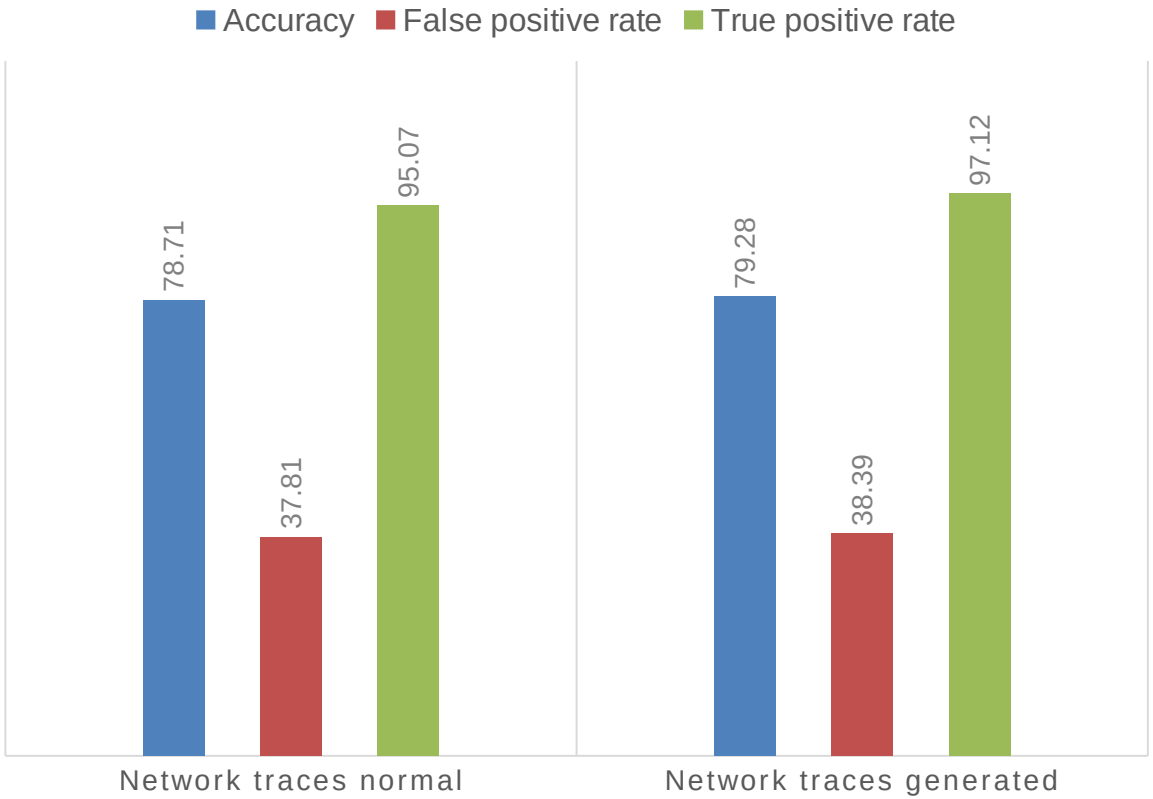
Results of the Arhuaco evaluation: deep learning algorithms

- **Metrics measured as percentage.**
- False positive Rate (Best result is close to 0%).
- Accuracy (Best result is close to 100%).
- Sensitivity or True Positive Rate (Same as accuracy).



CNN vs. SVM

- **CNN:** word2vec + convolutional neural network.
- **SVM:** Bag-of-words + support vector machine.



SVM vs. SVM with generated data

Summary

- Docker **containers** can be used to isolate and extract behavior information from grid jobs without big performance impact.
- **Deep learning** is highly effective to identify “malicious” grid jobs.
- **CNNs with word2vec** preprocessing provides improved accuracy than traditional SVM.
- Synthetic **generated data** can enhance the training dataset coverage for intrusion detection in grid computing.
- **Arhuaco** increases the security of the grid by a combination of isolation and security monitoring.
- **Source code** available here: https://gitlab.com/kuronosec/arhuaco. Licensed under **Apache 2.0.**

Thank you!

ALICE
A JOURNEY OF DISCOVERY

FAIR

IRI

GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN