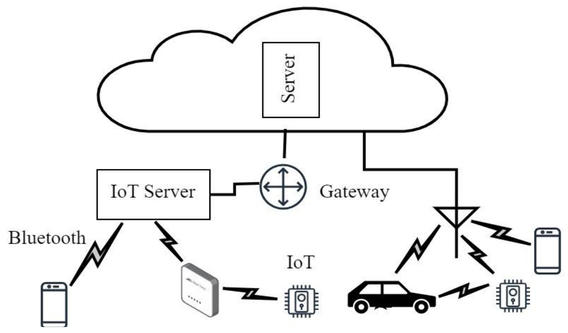
# Securing IoT Servers: Shallow vs. Deep Neural Network Architectures

Niranjan W. Meegammana, Shilpa Sayura Foundation, Sri Lanka
Harinda Fernando, Sri Lanka Institute of Information Technology, Sri Lanka

## INTRODUCTION




An IoT Environment

- Application servers in IoT environments face critical challenges from network attacks.
- Conventional IDS and firewalls struggle to mitigate these attacks due to the unique network topologies, heterogeneity, and resource limitations of IoT networks.
- We investigate the effectiveness of shallow and deep neural network (NN) architectures in detecting attacks on IoT application servers.

## OBJECTIVES

- Develop four shallow and a deep NN models.
- Create two datasets with 20 and 40 features.
- Train and test four models using two datasets.
- Compare and evaluate the performance of four models.

## SOLUTION

- Creating varied complex NN models.
- Deploy them on IoT environments with varying resource levels.

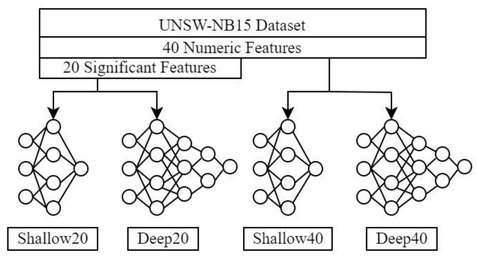
**Presented at**
**AINTEC 2024 Conference**
**ACM SIGCOMM 2024**
**UNSW, Sydney**
**Australia**

## METHODOLOGY

- Follow Machine Learning (ML) pipeline .
- Balance the UNSW-NB15 dataset using undersampling
- Create two datasets with 20 significant features obtained from Pearson Correlation and Information and 40 numeric features.
- Splitting: training (90%), validation (5%), and testing (5%).
- Shallow model : 1 layer (512 neurons)
- Deep model : 7 layers (256, 128, 64, 32, 16, 8, 4 neurons).
- Hyperparameter tuning using keras random search.

Best hyperparameters of four models

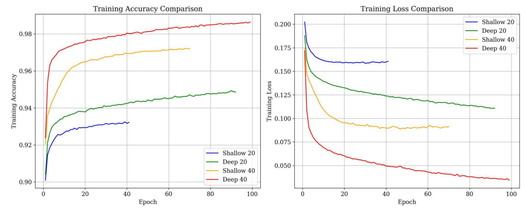
| Configuration | Shallow20 | Shallow40 | Deep20 | Deep40 |
|---|---|---|---|---|
| Model Parameters | 11265 | 49313 | 21008 | 54433 |
| Data Inputs | 20 | 40 | 20 | 40 |
| Hidden layers | 1 | 1 | 7 | 7 |
| Neurons per layer | 512 | 512 | 256, 128, 64, 32, 16, 8, 4 | 256, 128, 64, 32, 16, 8, 4 |
| activation_layer1 | Relu | Tanh | tanh | ReLU |
| activation_layer2 | - | - | tanh | ReLU |
| activation_layer3 | - | - | ReLU | leaky_relu |
| activation_layer4 | - | - | ReLU | ReLU |
| activation_layer4 | - | - | ReLU | ReLU |
| activation_layer6 | - | - | ReLU | ReLU |
| activation_layer7 | - | - | leaky_relu | leaky_relu |
| optimizer | Rmsprop | Adam | Adam | Adam |
| learning_rate | 0.001 | 0.001 | 0.001 | 0.001 |
| weight_initializer | he_normal | he_normal | he_normal | glorot_uniform |
| batch_size | 256 | 64 | 8 | 16 |
| Output function | Sigmoid | Sigmoid | Sigmoid | Sigmoid |




Four Models Trained

- Train four models using supervised learning.
- Test model performance and resource utilization.
- Compare and evaluate the model performance.

## RESULTS




Validation loss and accuracy change during training

Model performance evaluation

| Performance Data | Shallow20 | Deep20 | Shallow40 | Deep40 |
|---|---|---|---|---|
| Training Time (S) | 289 | 902 | 512 | 962 |
| Max. Validation Accuracy | 0.93 | 0.94 | 0.97 | 0.98 |
| End Epoc | 10 | 61 | 39 | 68 |
| Minimum Validation Loss | 0.166 | 0.130 | 0.088 | 0.057 |
| Prediction Time | 0.55 | 0.56 | 0.57 | 0.59 |
| Test Accuracy | 0.93 | 0.94 | 0.97 | 0.98 |
| Precision | 0.94 | 0.96 | 0.97 | 0.98 |
| Recall | 0.92 | 0.92 | 0.97 | 0.98 |
| F1 Score | 0.93 | 0.94 | 0.97 | 0.98 |
| ROC AUC Score | 0.986 | 0.990 | 0.996 | 0.998 |
| Average Precision | 0.987 | 0.991 | 0.996 | 0.998 |

Resource Utilization

| Performance Data | Shallow20 | Deep20 | Shallow40 | Deep40 |
|---|---|---|---|---|
| Model Size (Kb) | 110 | 190 | 648 | 708 |
| Prediction Time (s) | 0.55 | 0.56 | 0.57 | 0.59 |
| CPU Usage % | 0.25 | 0.26 | 0.30 | 0.32 |
| Memory Usage (GB) | 0.016 | 0.017 | 0.019 | 0.021 |

## CONCLUSION

- Deep40 model demonstrates superior accuracy (98%) and generalization, with high resource usage (0.32%).
- Shallow20 models resource-efficient with reduced accuracy (95% of Deep40), low resource usage (75% of Deep40) suitable for resource-constrained environments.

## REFERENCES

Aggarwal, Charu C. 2023. *Neural Networks and Deep Learning*. Springer Nature.

Bianchini, Monica, and Franco Scarselli. 2014. "On the Complexity of Neural Network Classifiers: A Comparison between Shallow and Deep Architectures." *IEEE Transactions on Neural Networks and Learning Systems* 25 (8): 1553–65. https://doi.org/10.1109/tnnls.2013.2293637.

Chollet, François. 2018. *Deep Learning with Python*. Shelter Island (New York, Estados Unidos): Manning.

Hapke, Hannes, and Catherine Nelson. 2020. *Building Machine Learning Pipelines : Automating Model Life Cycles with Tensorflow*. Beijing ; Boston ; Farnham ; Sebastopol ; Tokyo O'reilly.

Moustafa, Nour, Mohamed Abdel-Basset, and Reda Mohamed. 2022. *Deep Learning Approaches for Security Threats in IoT Environments*. Wiley-IEEE Press.