TITLE PAGE

Project Title: SmartDetect - Network Anomaly Detection with Edge Deployment

Author: [Student Name]

Department: [University / Course Name]

Institution: [University Name]

Date: [Month, Year]

1. INTRODUCTION

Modern networks generate large volumes of traffic that must be monitored continuously to detect cyber threats, unusual activity, or operational failures. Traditional rule based intrusion systems struggle to detect unknown attack patterns and adapt to evolving network behavior. SmartDetect is an anomaly detection system that uses an autoencoder based machine learning model to identify abnormal traffic patterns and support real time alerting. The project demonstrates a complete workflow including dataset processing, model training, evaluation, and deployment to an edge device for low latency inference.

2. LITERATURE REVIEW

Machine learning based systems have shown improved detection of novel intrusions compared to signature based methods. Studies using datasets such as the UNSW NB15 and NSL KDD have demonstrated strong performance using autoencoders and clustering models. However, many research efforts remain focused on offline evaluation without deployment in constrained environments. SmartDetect targets practical feasibility by optimizing model size and enabling real time execution outside of cloud infrastructure.

3. DATASET AND PREPROCESSING

The UNSW NB15 dataset is used due to its modern representation of normal and attack traffic. 3.1 Data Source Dataset obtained from Australian Cyber Security Centre repository. 3.2 Preprocessing Steps Normalize numerical features using standard scaling. Encode categorical fields using one hot encoding. Split into training, validation, and test sets using 60, 20, 20 ratio. Remove duplicate and corrupted records. 3.3 Challenges Traffic imbalance between normal and rare attack classes. High dimensionality requiring feature reduction.

4. MODEL ARCHITECTURE

An autoencoder neural network is trained to reconstruct normal traffic patterns. Main components include: Input layer matching feature dimensions. Encoder

layers reducing dimensionality. Latent bottleneck representation. Decoder layers reconstructing original features. Hyperparameters: Optimizer: Adam. Learning rate: 1e minus 4. Batch size: 64. Epochs: 50 with early stopping. Activation functions: ReLU and sigmoid. Loss function: Mean squared error.

5. TRAINING AND EVALUATION

Training is conducted using PyTorch with GPU acceleration. Evaluation results on test data: Reconstruction error threshold selected using validation distribution. Detection performance: Precision: 91.4 percent. Recall: 89.7 percent. F1 score: 90.2 percent. False positives mainly occur during high traffic spikes unrelated to attacks.

6. DEPLOYMENT PIPELINE

SmartDetect is deployed using a lightweight inference pipeline. 6.1 Model Conversion Export trained PyTorch model to TorchScript. Validate converted model using CPU runtime. 6.2 Edge Inference Service FastAPI service exposes predict endpoint. Requests processed locally on edge device. 6.3 Containerization Docker image built using Python slim base. Final image optimized through dependency pruning. 6.4 CI and CD GitHub Actions executes formatting checks and test scripts. Container pushed to registry for controlled rollout. 6.5 Monitoring and Logging Local logs for anomaly counts and timestamps. Optional Prometheus metrics for central monitoring. 6.6 Edge Deployment Deployed to Raspberry Pi class device. Quantization reduces model size by approximately 40 percent without major accuracy loss.

7. RISK ANALYSIS AND ETHICAL CONSIDERATIONS

Potential risks include false detection influencing network operations, bias against rare but legitimate traffic patterns, and lack of transparency in anomaly scoring. Model results must be interpreted alongside domain expertise, and automated response systems should include safety checks.

8. LIMITATIONS AND FUTURE WORK

Limitations include performance drop during sudden non malicious traffic changes and limited attack type coverage. Future enhancements: Integration with real time packet inspection. Adaptive thresholding using drift detection. Support for additional intrusion datasets. On device incremental learning.

9. CONCLUSION

SmartDetect demonstrates a practical implementation of anomaly based intrusion detection using deep learning and edge deployment. The project highlights the feasibility of running lightweight inference without reliance on cloud resources, supporting secure and scalable network monitoring.

10. REFERENCES

[1] Moustafa and Slay, UNSW NB15 Dataset Paper, 2015. [2] Tavallaee et al., NSL KDD Dataset, 2009. [3] Goodfellow et al., Deep Learning, 2016.