

Research / Project Plan for the Next Term

1. Research Objective for the Next Term

The objective of the next phase of this research is to design, implement, and evaluate a hybrid deep learning model that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for network intrusion detection. This phase focuses on transforming the proposed framework into a fully functional system and validating its effectiveness using a benchmark dataset.

2. Methodology Plan

The project will follow a structured methodology consisting of several key stages.

The first stage involves dataset preparation. The NSL-KDD dataset will be collected and analyzed to understand its structure and features. Data preprocessing will include cleaning the dataset, handling any inconsistencies, encoding categorical features, and normalizing numerical values. The dataset will then be divided into training and testing sets to support model development and evaluation.

The second stage focuses on model development. A Convolutional Neural Network (CNN) will be designed to extract meaningful spatial features from the dataset, while a Long Short-Term Memory (LSTM) network will be used to capture temporal patterns in the data. These two models will be integrated into a hybrid CNN-LSTM architecture to enhance the overall performance of intrusion detection.

The third stage is model training. The proposed model will be trained using the prepared dataset. Various hyperparameters such as learning rate, batch size, and number of epochs will be tuned to optimize model performance. Techniques such as regularization and dropout may be applied to reduce the risk of overfitting.

The fourth stage involves model evaluation. The trained model will be tested using the testing dataset, and its performance will be measured using standard evaluation metrics including Accuracy, Precision, Recall, and F1-score. Additional evaluation tools such as confusion matrix and ROC curve will be used to analyze classification performance in detail.

The final stage includes comparison with traditional machine learning models. Algorithms such as Decision Tree, Random Forest, and Support Vector Machine (SVM) will be implemented and compared with the proposed CNN-LSTM model to demonstrate the effectiveness of the hybrid approach.

3. Tools and Technologies

The implementation of this project will require several tools and technologies. Python will be used as the primary programming language, along with deep learning frameworks such as TensorFlow or PyTorch for model development. Data processing will be performed using libraries such as Pandas and NumPy. Scikit-learn will be used for implementing traditional machine learning models and evaluation metrics. Visualization tools such as Matplotlib and Seaborn will be used to present results. The development environment will be based on Jupyter Notebook or Visual Studio Code.

4. Timeline Plan

The project will be completed in a sequence of phases over the next term. Initially, the dataset will be prepared and preprocessed. This will be followed by the design of the model architecture.

After that, the model will be implemented and trained using the dataset. Once the training phase is complete, the model will be evaluated and tested to assess its performance. Finally, the results will be analyzed, and the findings will be documented as part of the thesis.

5. Expected Outcomes

The expected outcome of this research is the development of a robust hybrid CNN-LSTM model capable of accurately detecting network intrusions. The model is expected to achieve improved detection accuracy compared to traditional machine learning approaches. Additionally, the research aims to reduce false positive rates and provide a detailed analysis of the model's performance across different evaluation metrics.

6. Risk and Challenges

Several challenges may arise during the implementation of this project. These include data imbalance within the dataset, which may affect model performance, as well as the risk of overfitting in deep learning models. Computational limitations may also pose challenges when training complex models. Furthermore, selecting optimal hyperparameters may require extensive experimentation. These challenges will be addressed through appropriate data preprocessing techniques, model optimization strategies, and careful evaluation.