ML Models for DDoS Attack Classification

Introduction

This document explains the different Machine Learning (ML) models used for detecting and classifying DDoS attacks in Autonomous Electric Vehicle (EV) networks. The models implemented include **Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Naïve Bayes (NB)**.

---

1. Support Vector Machine (SVM)

SVM was the best-performing model, achieving **95.50% accuracy**.
It works by finding an **optimal hyperplane** that maximizes the margin between classes.

Python Implementation:

```
from sklearn.svm import SVC import joblib  def train_svm(X_train, y_train):     svm_model = SVC(kernel='rbf', C=1.0, gamma='scale') svm_model.fit(X_train, y_train)     joblib.dump(svm_model, "models/svm_model.pkl")  # Save model     return svm_model`
```

---

2. K-Nearest Neighbors (KNN)

KNN achieved **94.33% accuracy**.
It works by classifying a data point based on the **majority class** of its k nearest neighbors.

Python Implementation:

```
from sklearn.neighbors import KNeighborsClassifier import joblib  def train_knn(X_train, y_train):     knn_model = KNeighborsClassifier(n_neighbors=5)     knn_model.fit(X_train, y_train) joblib.dump(knn_model, "models/knn_model.pkl")  # Save model     return knn_model
```

3. Logistic Regression (LR)

Logistic Regression achieved **90.75% accuracy**.
It is useful for **binary classification** tasks and predicts probabilities using a **logistic function**.

Python Implementation:

```
from sklearn.linear_model import LogisticRegression import joblib  def
train_lr(X_train, y_train):     lr_model = LogisticRegression()
lr_model.fit(X_train, y_train)     joblib.dump(lr_model,
"models/lr_model.pkl")  # Save model     return lr_model`
```

---

4. Naïve Bayes (NB)

Naïve Bayes achieved **82.06% accuracy**.
It is based on **Bayes' theorem** and assumes independence between features, making it efficient for **probabilistic classification**.

Python Implementation:

```
from sklearn.naive_bayes import GaussianNB import joblib  def
train_nb(X_train, y_train):     nb_model = GaussianNB()
nb_model.fit(X_train, y_train)     joblib.dump(nb_model,
"models/nb_model.pkl")  # Save model     return nb_model`
```

Conclusion

- **SVM** was the best-performing model with the highest accuracy.
- **KNN** performed well but can be computationally expensive for large datasets.
- **Logistic Regression** is effective for simple classification tasks.

- **Naïve Bayes** works well for probabilistic classification but assumes feature independence.