Project Title: Intrusion Detection System Using Machine Learning (Windows Environment)

Overview: This project aims to build a Machine Learning-based Intrusion Detection System (IDS) that can classify network traffic into normal or attack categories. The implementation is done entirely on a Windows machine using Python and publicly available network traffic datasets.

Key Components:

1. **Dataset Selection**:

2. Chosen a dataset (e.g., CICIDS2017, NSL-KDD, etc.) that contains labeled network traffic samples.

3. Downloaded the dataset in `.csv` format.

4. **Data Loading**:

5. Used `pandas` to load the dataset.

6. Automatically extracted all available column names to detect the label/target column.

```
import pandas as pd

df = pd.read_csv('path_to_dataset.csv')
print(df.columns)  # To identify the label column
```

1. **Data Preprocessing**:

2. Handled missing values, if any.

3. Encoded categorical values (e.g., attack types).
4. Scaled numerical features using `StandardScaler`.

```
from sklearn.preprocessing import LabelEncoder, StandardScaler

df.dropna(inplace=True)
label_encoder = LabelEncoder()
df['Label'] = label_encoder.fit_transform(df['Label'])

scaler = StandardScaler()
```

```
X = scaler.fit_transform(df.drop('Label', axis=1))
y = df['Label']
```

1. **Model Training**:

2. Split the data into training and testing sets.

3. Trained a classifier (e.g., Random Forest).

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

1. **Model Evaluation**:

2. Evaluated using accuracy, precision, recall, F1-score.

3. Plotted confusion matrix for visual inspection.

```
from sklearn.metrics import classification_report, confusion_matrix

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

1. **Prediction on New Data**:

2. Load new network data and classify it using the trained model.

```
new_data = pd.read_csv('new_data.csv')
new_data_processed = scaler.transform(new_data.drop('Label', axis=1))
predictions = model.predict(new_data_processed)
```

1. **Environment**:

2. OS: Windows 10

3. Language: Python 3.11
4. Libraries: pandas, sklearn, matplotlib, seaborn

Future Scope:

- Integrate with real-time traffic using `scapy` or `pyshark`.
- Add a web-based dashboard for live alerts.
- Extend to deep learning models (e.g., LSTM for sequence-based traffic).

Result: Achieved successful training and classification of attack vs normal traffic. The model is ready to be deployed on a lightweight Windows setup without the need for Kali Linux or any penetration testing framework.