**🔐 Intrusion Detection System (IDS)**

**👨‍💻 Submitted By**

* **Name:** Vishal Prajapati
* **Department:** CSE (AI & ML), KIET Group of Institutions
* **Email:** vishalprajapatisp321@gmail.com

**📌 Problem Statement**

In the modern digital world, networks face constant cyber threats. Traditional IDS tools that use signature-based detection often fail to detect novel or zero-day attacks. This project addresses the challenge by implementing a **machine learning-based IDS** that detects network anomalies using **supervised learning** on the **UNSW-NB15 dataset**.

**💡 Proposed Solution**

We developed a system that:

* Processes network traffic features
* Trains a machine learning model (XGBoost)
* Deploys the model via a Flask API
* Accepts real-time traffic feature input (via JSON)
* Predicts:  
  0 → Normal Traffic  
  1 → Intrusion/Attack

**🛠️ Technologies Used**

| **Component** | **Description** |
| --- | --- |
| **Language** | Python |
| **ML Algorithm** | XGBoost (Gradient Boosting Classifier) |
| **Libraries** | scikit-learn, numpy, pandas, joblib |
| **Framework** | Flask (REST API server) |
| **Platform** | Google Colab (training), VS Code (API) |
| **Testing Tool** | Python requests library |

**🔄 System Architecture**

1. **Data Preprocessing**
   * Used UNSW-NB15 dataset
   * Selected key features like sbytes, tcprtt, Dpkts, etc.
   * Encoded categorical features (like proto, state)
   * Scaled features using StandardScaler
2. **Model Training**
   * Trained an **XGBoost Classifier**
   * Achieved ~95% accuracy
   * Saved the model as logistic\_model.pkl (you later used Logistic Regression for deployment)
3. **Model Deployment**
   * Used Flask to deploy the API
   * / (GET): Confirm API running
   * /predict (POST): Accepts 12 features → Returns 0 or 1

**🧪 API Testing**

**✅ GET Request**

📄 **Script: getrequest.py**

python

CopyEdit

url = 'http://127.0.0.1:5000/'

response = requests.get(url)

🟢 Response: "Intrusion Detection API is running 🚀"

**✅ POST Request**

📄 **Script: postrequest.py**

python

CopyEdit

url = 'http://127.0.0.1:5000/predict'

data = { ... } # 12 features

response = requests.post(url, json=data)

🟢 Response: {'prediction': 0} or {'prediction': 1}

**📊 Result**

* **Model Accuracy:** 95%
* **Evaluation Metrics:** Precision, Recall, F1-Score
* **Output Screenshot:** Attached (output.png, Intrusion\_Detection\_unsw\_nb15 - Output.png)
* **Model Deployment:** Successfully integrated with Flask and responds to live API calls

**✅ Key Features**

* Real-time attack prediction
* Lightweight Flask deployment
* Easy integration for further security tools
* Can be hosted locally or on a cloud server

**🔮 Future Scope**

* Integrate user behavior analytics
* Use deep learning models for better accuracy
* Real-time alert system (SMS/email)
* Dashboard with visualization (e.g., via Streamlit or Dash)
* Extend to IoT networks or edge devices

**🔚 Conclusion**

This project demonstrates how machine learning can strengthen cybersecurity by building an **automated, scalable, and real-time Intrusion Detection System**. By leveraging XGBoost and deploying it with Flask, this system provides a practical, lightweight prototype ready for extension in industrial networks.

**🔗 References**

* [UNSW-NB15 Dataset](https://www.unsw.edu.au/about-us/our-story/our-story)
* [XGBoost Documentation](https://xgboost.readthedocs.io/en/latest/)
* [Scikit-learn Documentation](https://scikit-learn.org/stable/)
* Flask Web Development – Miguel Grinberg