



# Comparative Analysis of Multiple Machine Learning Models for Cloud-Based Intrusion Detection Systems



A CSE484 Group Project Demonstration







# What is Intrusion Detection?

Intrusion detection is like a **burglar alarm** for computer networks. Just as you'd protect your house from thieves, intrusion detection systems monitor network activity for unauthorized access or malicious behavior. When detected, they raise alerts, allowing swift response to potential threats.

An intrusion detection system is like a **security guard** for computers, watching for any bad guys trying to break in.

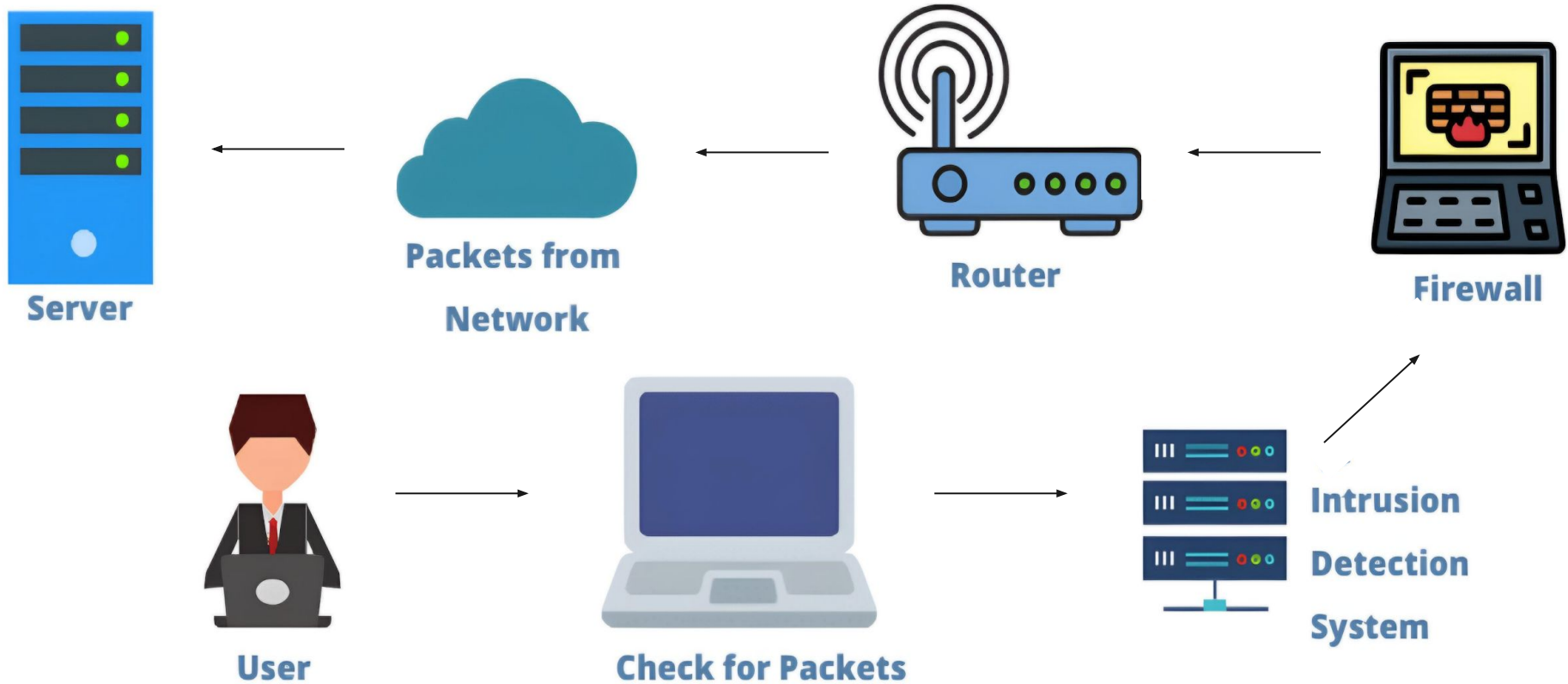


# What are we trying to Achieve?

1. Test all the popular Machine Learning algorithms to know who is more suitable for intrusion detection.
2. Deploy the best trained model on a cloud server.
3. Implement the server as a middleman to protect the real website.

We're finding the **smartest computer friend** to protect our clubhouse online.

# INTRUSION DETECTION SYSTEM





# The Dataset for training the models is


A real world Network  
Intrusion Detection  
Dataset from kaggle. It has **all  
the basic features** that a server  
receives when a computer tries to  
connect to it.

Kaggle's cool dataset mimics **computer chats** with servers!

# Features Used to Train The Models

	src_bytes	dst_bytes	flag	diff_srv_rate	count	logged_in	dst_host_diff_srv_rate	error_rate	service	protocol_type	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	srv_count	dst_host_count	hot
0	491	0	9	0.00	2	0	0.03	0.00	19	1	0.17	0.00	2	150	0
1	146	0	9	0.15	13	0	0.60	0.00	41	2	0.88	0.00	1	255	0
2	0	0	5	0.07	123	0	0.05	1.00	46	1	0.00	0.00	6	255	0
3	232	8153	9	0.00	5	1	0.00	0.20	22	1	0.03	0.04	5	30	0
4	199	420	9	0.00	30	1	0.00	0.00	22	1	0.00	0.00	32	255	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
26835	0	0	1	0.07	244	0	0.08	0.00	55	1	0.00	0.00	18	255	0
26836	0	0	1	0.07	137	0	0.07	0.00	64	1	0.00	0.00	7	255	0
26837	0	0	1	1.00	509	0	1.00	0.21	41	1	0.00	0.00	1	255	0
26838	0	0	5	0.06	252	0	0.08	1.00	33	1	0.00	0.00	2	255	0
26839	0	0	5	0.05	253	0	0.05	1.00	34	1	0.00	0.00	22	255	0

26840 rows x 15 columns




# The Machine Learning Models

It Is a classification based task. So we trained the dataset on multiple classification models like **Random Forest, KNN, Logistic Regression, AdaBoostClassifier, Naive Bayes**. After training and testing the model we choose Random Forest as the our trained model.

**Random Forest**, The Champion!





# Deployment of the trained model

1. **Deployed** the model on streamlit.
2. This website will work as the **middleman** between the actual website and the client.
3. Because of the short time we could not build the website, but here is a **short demonstration**.

Model's on Streamlit, our website superhero! Quick demo time!





# Deployment links

**Detection System** → [System](#)

**Model Training Code** → [Colab](#)

**Dataset** → [Dataset](#)

Quick demo time!

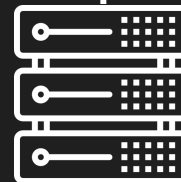
Incoming  
Traffic

```
1 df = df.loc[:, ['service',  
2 'flag',  
3 'src_bytes',  
4 'dst_bytes',  
5 'count',  
6 'same_srv_rate',  
7 'diff_srv_rate',  
8 'dst_host_srv_count',  
9 'dst_host_same_srv_rate',  
10 'dst_host_same_src_port_rate',  
11 'class']]
```



Streamlit



Access  
Denied



# Visualization of the Whole Process



# Conclusion And Summary

1. As the accuracy of the model is **not** 100%, we will get some error.
2. These are basic ML models implementation, in future we want to work with more complex models to detect the underlying relationship.

Every end is a new beginning.

