****

**Work Integrated Learning Programmes Division**

**M.Tech (AIML/ DSE)**

**Machine Learning**

**Assignment**

**Problem Statement 5**

**ML based network intrusion detection**

Develop a machine learning-based classification model to identify and categorize network traffic as normal or malicious using the NSL-KDD dataset. Malicious activities include different attack types such as DoS, Probe, R2L, and U2R attacks. The model should accurately classify the traffic type, helping to detect and prevent cyber intrusions effectively.

Dataset:

[KDDTrain+.txt](https://github.com/jmnwong/NSL-KDD-Dataset)

**Metadata:**

1. duration: Length of the connection in seconds.
2. protocol\_type: Type of protocol (e.g., TCP, UDP, ICMP).
3. service: Network service on the destination (e.g., HTTP, FTP, SMTP).
4. flag: Status flag of the connection (e.g., SF for a normal connection, RSTO for reset).
5. src\_bytes: Number of data bytes sent from source to destination.
6. dst\_bytes: Number of data bytes sent from destination to source.
7. land: Boolean flag indicating whether the connection is to/from the same host and port.
8. wrong\_fragment: Number of wrong fragments in the packet.
9. urgent: Number of urgent packets in the connection.
10. hot: Number of “hot” indicators (e.g., access to critical files).
11. num\_failed\_logins: Number of failed login attempts.
12. logged\_in: Boolean flag indicating if the login was successful.
13. num\_compromised: Number of compromised conditions.
14. root\_shell: Boolean flag indicating if root access is obtained.
15. su\_attempted: Number of su root command attempts.
16. num\_root: Number of root accesses.
17. num\_file\_creations: Number of file creation operations.
18. num\_shells: Number of shell prompts invoked.
19. num\_access\_files: Number of operations accessing control files.
20. num\_outbound\_cmds: Number of outbound commands in an FTP session.
21. is\_host\_login: Boolean flag indicating if the login belongs to the host list.
22. is\_guest\_login: Boolean flag indicating if the login is for a guest account.
23. count: Number of connections to the same host as the current connection in the past 2 seconds.
24. srv\_count: Number of connections to the same service as the current connection in the past 2 seconds.
25. serror\_rate: Percentage of connections that have SYN errors.
26. srv\_serror\_rate: Percentage of connections to the same service that have SYN errors.
27. rerror\_rate: Percentage of connections that have REJ errors.
28. srv\_rerror\_rate: Percentage of connections to the same service that have REJ errors.
29. same\_srv\_rate: Percentage of connections to the same service.
30. diff\_srv\_rate: Percentage of connections to different services.
31. srv\_diff\_host\_rate: Percentage of connections to different hosts.
32. dst\_host\_count: Number of connections to the same destination host.
33. dst\_host\_srv\_count: Number of connections to the same service as the destination host.
34. dst\_host\_same\_srv\_rate: Percentage of connections to the same service at the destination host.
35. dst\_host\_diff\_srv\_rate: Percentage of connections to different services at the destination host.
36. dst\_host\_same\_src\_port\_rate: Percentage of connections to the same source port at the destination host.
37. dst\_host\_srv\_diff\_host\_rate: Percentage of connections to different destination hosts for the same service.
38. dst\_host\_serror\_rate: Percentage of connections to the destination host with SYN errors.
39. dst\_host\_srv\_serror\_rate: Percentage of connections to the destination host and same service with SYN errors.
40. dst\_host\_rerror\_rate: Percentage of connections to the destination host with REJ errors.
41. dst\_host\_srv\_rerror\_rate: Percentage of connections to the destination host and same service with REJ errors.

**Class Labels**

* Normal: Indicates legitimate traffic.
* Attack: Includes various intrusions such as Smurf, Back, Land, Neptune, Pod, etc.

As we are working on a binary classification problem, label the classes other than ‘Normal’ as ‘Attack’.

**Important points to remember**

* Use only classical machine learning models. Strictly no deep learning models.
* Structure and organize the code into sections. Provide your observations at the end of each section.

**Assignment workflow**

1. **Import Libraries/Dataset** 
   1. Download the dataset.
   2. Import the required libraries and packages.
2. **Data Visualization and Exploration [1M]**
   1. Print 5 rows for sanity check to identify all the features present in the dataset and if the target matches with them.
   2. Provide appropriate data visualizations to get an insight about the dataset.
   3. Do the correlational analysis on the dataset. Provide a visualization for the same. Will this correlational analysis have effect on feature selection that you will perform in the next step? Justify your answer. **Answer without justification will not be awarded marks.**
3. **Data Pre-processing and cleaning [2M]**
   1. Do the appropriate pre-processing of the data like identifying NULL or Missing Values if any, handling of outliers if present in the dataset, skewed data etc. Perform feature encoding. Mention the pre-processing steps performed in the markdown cell.
   2. Apply appropriate feature engineering techniques (if applicable). Apply the feature transformation techniques like Standardization, Normalization, etc. You are free to apply the appropriate transformations depending upon the structure and the complexity of your dataset. Provide proper justification. **Techniques used without justification will not be awarded marks**. Explore a few techniques for identifying feature importance for your feature engineering task.

**4. Model Building [11M]**

* 1. Split the dataset into training and test sets. **Answer without justification will not be awarded marks.** [1]
     1. Train = 80 % Test = 20%
     2. Also, try to split the dataset with different ratios of your choice.
  2. 1) Implement predictive models/classifiers using the following classification approaches: [8]
     1. Logistic Regression
     2. Decision tree
     3. K-Nearest Neighbour
     4. Ensemble Methods (any one of your choice)

1. Tune hyperparameters (e.g., number of trees, maximum depth) using cross-validation. For each of the above models. Justify your answer. [2]

**5. Performance Evaluation [6M]**

* 1. Compare the performances of each model/classifier considering the given dataset using different evaluation measures such as Precision, Recall, F1-Score, AUC-ROC. Show the comparison chart in Python notebook [4]
  2. Identify the model, which you think is the best amongst all the models that have been trained. Also, explain why you think this is the best model. Answer this question in the notebook itself. [2]

**Instructions for Assignment Evaluation**

1. Organise your code in separate sections for each task. Add comments to make the code readable.
2. Deep Learning Models are strictly not allowed. You are encouraged to learn classical Machine learning techniques and experience their behaviour.
3. Notebooks without output shall not be considered for evaluation.

For clarification, contact **Nivethitha Somu (Dr.)** at **nivethitha.somu@wilp.bits-pilani.ac.in**