NEW YORK INSTITUTE OF TECHNOLOGY
Vancouver

INCS 615: Advanced Network and Internet Security

Project: Network Intrusion Detection Using Machine Learning

Instructor: Dr. Zhida Li
Summer 2023 SYLLABUS (INCS 615-VA1)
Email: zli74@nyit.edu

Term Project (group)

Welcome to the group-based project (in groups of up to 3). In this project, we will learn four types of intrusions and use the machine learning model(s) to detect them: **coding (Python) + presentation**.

You need to give a presentation on Week 7.2 (July 6). Each group will have 10 minutes (presentation + Q&A)

Term Project carries 20% of your final grade: project code – group (10%) + presentation – group (10%)

Tasks:

1. **Download the shared folder from Google Drive:**
   https://drive.google.com/drive/folders/1Iwd2LqYEJcXfv6mcsCeJ1Sb-C80Q5_z7?usp=sharing
   Folder name:
   NIDS_using_MachineLearning
   Unzip the folder.

   Folder content:
   - Python code (Jupyter file): *INCS615_project_NIDS_ML.ipynb*
   - NSL-KDD training data (42 columns): *KDDTrain+_20Percent_651.csv*
   - NSL-KDD testing data (42 columns): *KDDTest+_651.csv*
   - Webpage (NSL-KDD) offline: *nsl-kdd_index.html*
   - Paper for NSL-KDD: *A_detailed_analysis_of_the_KDD_CUP_99_data_set.pdf*
   - Paper for additional details of NSL-KDD:
   A_detailed_analysis_of_the_KDD_CUP_99_data_set.pdf

   Content of the CSV files:
   Row 1: feature names
   Columns 1-41: features
   Column 42: labels for regular = 0, DOS = 1, R2L = 2, U2R = 3, Probe = 4.

2. **Learn and understand your NSL-KDD data:**
Cyber attacks are becoming more sophisticated and, hence, more difficult to detect. Using efficient and effective machine learning techniques to detect network anomalies and intrusions is an important aspect of cyber security.

Machine learning algorithms have been evaluated for robustness, high accuracy, and training time when classifying various datasets collected from communication networks. Reliable testing and validation of anomaly and intrusion detection algorithms depend on the quality of datasets such as traffic collected from deployed networks or experimental testbeds. The most widely used benchmark datasets in the literature are Knowledge Discovery in Databases (KDD) Cup 1999 (KDD'99) and NSL-KDD. KDD'99 intrusion dataset is based on the Defense Advanced Research Projects Agency (DARPA) 1998 dataset.

The NSL-KDD dataset is an improved version of the KDD'99 intrusion dataset. Data were captured from an evaluation testbed and included large numbers of virtual hosts and user automata. The KDD'99 dataset was used in various IDSs. NSL-KDD is a randomly selected subset of KDD'99 after redundant data were removed and is a widely used benchmark for evaluating anomaly detection techniques. NSL-KDD dataset captures TCP, UDP, and Internet Control Message Protocol (ICMP) traffic collected using the tcpdump utility. It contains four types of intrusion attacks: DoS, R2L, U2R, and Probe described in Table 1.

Each network connection is represented by 41 features: 38 numerical and 3 categorical ("protocol_type", "service", and "flag") features. Categorical features will be removed in our experiment.

Table 1: NSL-KDD dataset: Four types of intrusion attacks are listed: DoS, R2L, U2R, and Probe.

Type	**Intrusion attacks**
DoS	back, land, neptune, pod, smurf, teardrop, mailbomb, processtable, udpstorm, apache2, worm
U2R	buffer-overflow, loadmodule, perl, rootkit, sqlattack, xterm, ps
R2L	fpt-write, guess-passwd, imap, multihop, phf, spy, warezmaster, xlock, xsnoop, snmpguess, snmpgetattack, httptunnel, sendmail, named
Probe	ipsweep, nmap, portsweep, satan, mscan, saint

The features with continuous and discrete types are described in Table 2. We consider five-way classification, the targets are regular, DoS, R2L, U2R, and Probe, labeled as 0, 1, 2, 3, and 4, respectively. We employ the KDDTrain+_20Percent_651 dataset for training and KDDTest+_651 dataset for testing.

Table 2: NSL-KDD features: Definitions, types, and descriptions. Each network connection is represented by 41 features: 38 numerical and 3 categorical ("protocol_type", "service", and "flag")

features.

Feature	Definition	Type	Description
1	duration	continuous	length (seconds) of the connection
2	protocol_type	discrete	type of the protocol (TCP, UDP)
3	service	discrete	network service on the destination, (HTTP, telnet)
4	flag	discrete	normal or error status of the connection
5	src_bytes	continuous	no. of data bytes from source to destination
6	dst_bytes	continuous	no. of data bytes from destination to source
7	land	discrete	1 if connection is from/to the same host/port; 0 otherwise
8	wrong_fragment	continuous	no. of "wrong" fragments
9	urgent	continuous	no. of urgent packets
10	hot	continuous	no. of "hot" indicators
11	num_failed_logins	continuous	no. of failed login attempts
12	logged_in	discrete	1 if successfully logged in; 0 otherwise
13	num_compromised	continuous	no. of "compromised" conditions
14	root_shell	discrete	1 if root shell is obtained; 0 otherwise
15	su_attempted	discrete	1 if "su root" command attempted; 0 otherwise
16	num_root	continuous	no. of "root" accesses
17	num_file_creations	continuous	number of file creation operations
18	num_shells	continuous	no. of shell prompts
19	num_access_files	continuous	no. of operations on access control files
20	num_outbound_cmds	continuous	no. of outbound commands in an ftp session
21	is_host_login	discrete	1 if the login belongs to the "hot" list; 0 otherwise
22	is_guest_login	discrete	1 if the login is a "guest"login; 0 otherwise
23	count	continuous	no. of connections to the same host as the current connection in the past 2 s
24	srv_count	continuous	no. of connections to the same service as the current connection in the past 2 s
25-26	serror_rate and srv_serror_rate	continuous	no. of connections that have "SYN" errors
27-28	rerror_rate and srv_rerror_rate	continuous	no. of connections that have "REJ" errors
29	same_srv_rate	continuous	no. of connections to the same service
30	diff_srv_rate	continuous	no. of connections to different services
31	srv_diff_host_rate	continuous	no. of connections to different hosts
32	dst_host_count	continuous	no. of connections to the same service as the current connection in the past 2 s
33	dst_host_srv_count	continuous	no. of connections to the same service as the current connection in the past 2 s
34	dst_host_same_srv_rate	continuous	no. of connections to the same service
35	dst_host_diff_srv_rate	continuous	no. of connections to different services
36	dst_host_same_src_port_rate	continuous	no. of connections to the same source port
37	dst_host_srv_diff_host_rate	continuous	no. of connections to different hosts
38-39	dst_host_serror_rate and dst_host_srv_serror_rate	continuous	no. of connections that have "SYN" errors no. of connections that have "SYN" errors
40-41	dst_host_rerror_rate and dst_host_srv_rerror_rate	continuous	no. of connections that have "REJ" errors

References:

[1] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Ottawa, ON, Canada, July 2009, pp. 1–6.

[2] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory," *ACM Trans. Inform. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, Nov. 2000.

3. **Go back to Google Drive:**
   Create your own copy of the folder:
   NIDS_using_MachineLearning_teamNo

* Note: This is very important so that you do not alter the original code.

While in your own folder on Google Drive, upload your data, run (double click) on:
INCS615_project_NIDS_ML.ipynb

Note:
We assume that you do not already have the Colab web application.
To use Colab, select "Open with", search for Application named Colaboratory, and install it. Finally, connect to the Application.

For advanced users only:

In case you wish to run the exercise locally, you need to install Jupyter development environment.
Instructions are posted at:
https://jupyter-notebook.readthedocs.io/en/stable/

To install Jupyter, you will also need the "pip" package management system (installer) for Python:
python get-pip.py
Instruction is available at:
https://jupyter.org/install
---

4. **You are now ready to follow the steps:**
   INCS615_project_NIDS_ML.ipynb

Machine learning libraries may be used:

- https://scikit-learn.org/stable/index.html
- https://pandas.pydata.org/
- https://pytorch.org/
- https://www.tensorflow.org/

Resources

Journals

- IEEE Communications Surveys & Tutorials
- IEEE Network - The Magazine of Global Internetworking
- IEEE Communications Magazine
- ACM Computer Communication Review
- IEEE Journal on Selected Areas in Communications
- IEEE Transactions on Cybernetics
- IEEE Transactions on Systems, Man, and Cybernetics
- IEEE/ACM Transactions on Networking

Note:
Papers from ACM: https://libguides.nyit.edu/az.php?s=16453&a=a&p=1, select "ACM Digital Library", then log in.
Papers from IEEE Explore: https://libguides.nyit.edu/az.php?s=16453&a=i&p=1, select "IEEE Xplore", then log in.

Citation style

Please follow the IEEE citation style when writing references:

- IEEE Editorial Style Manual: https://journals.ieeeauthorcenter.ieee.org/create-your-ieee-article/createthe-text-of-yourarticle/ieee-editorial-style-manual/

- IEEE Reference Guide: http://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf

How to Present Your Project

10 minutes of presentation + 1 minute of Q&A
Make sure all team members are present and that everyone gets properly introduced.

First slide outline:

INCS 615: Advanced Network and Internet Security
Summer 2023
Term Project Presentation
PROJECT TITLE

YOUR NAME(s)
YOUR NYIT STUDENT IDs
YOUR E-MAIL ADDRESS(es)
YOUR TEAM #

Consecutive slides:
1. INTRODUCTION TO THE PROJECT
Introduce the project. Include motivation and overview. Describe the project idea, its scope, and what you are trying to accomplish. State the most important aspect of the project. Present an overall, high-level description of your project.

2. OVERVIEW OF RELATED WORK
Include a short overview and description of related work and state-of-the-art that you have cited in references. The related work can include the application of IDS and the use of machine learning for classifying NSL-KDD data.

3. PROBLEM DESCRIPTION
Provide technical details of the problem that you plan to address/discuss.

4. IMPLEMENTATION
Include data analysis, machine learning algorithm (s), results, and analysis (even if not completed). Include descriptions of the overall design, use flowcharts for sections of the design as needed, and high-level architecture (if appropriate).

5. DISCUSSION
Describe what made it a non-trivial task and what the difficulties were. Suggest alternative approaches to the solution. Suggest improvements and future work. Describe briefly what you have learned from this project.

6. REFERENCES
List the references used and give credits to people whose work you used.

7. CONTRIBUTIONS
List the contributions of the individual team members.

Grading Scheme

CLASS PRESENTATION (total 100 points)

1. Introduction to the project: motivation and overview (5 points)
2. Overview (20 points)
3. Problem description: technical details (20 points)
4. Implementation: (even if not completed) (50 points)
5. Organization and time management (5 points)

Please Submit:

- **Code** (py or *ipynb*) and **Presentation slides** (PowerPoint and PDF files) of your term project.

  Submit the file(s) for your project submission to the Canvas site.

- Include the cover page.

- Note:
  More credit will be given to completed projects (even if they are less difficult) than to the ambitious unfinished work.
  In the case of team work, we will make every attempt to evaluate the individual performance of each team member.

Good luck!