Experiential Learning: Case Study-based Portable Hands-on Regression Labware for Cyber Fraud Prediction

Hossain Shahriar[1], Michael Whitman[2], Dan Lo[3], Fan Wu[4], Cassandra Thomas[4], Alfredo Cuzzocrea[5]
[1]Department of Information Technology, Kennesaw State University, USA
[2]Institute for Cybersecurity Workforce Development, Kennesaw State University, USA
[3]Department of Computer Science, Kennesaw State University, USA
[4]Department of Computer Science, Tuskegee University, USA
[5]DISPES Department, University of Calabria, Italy

Abstract - **Machine Learning (ML) analyzes, and processes data and develop patterns. In the case of cybersecurity, it helps to better analyze previous cyber attacks and develop proactive strategy to detect and prevent current and future security attacks. Both ML and cybersecurity are important subjects in computing curriculum, but using ML for cybersecurity is commonly explored. This paper designs and presents a case study-based portable labware experience built on Google's CoLaboratory (CoLab) for a ML cybersecurity application to provide students with access to hands-on labs anywhere and anytime, reducing or eliminating tedious installation and configuration requirements. This will help students better focus on learning concepts and gaining valuable experience through hands-on problem solving skills. This paper provides an overview of case-based hands-on regression labware for cyber fraud prediction using credit card fraud as an example.**

Keywords* – *Case Study-Based Learning, Cybersecurity, Machine Learning, Credit Fraud, Spam Detection.

I. INTRODUCTION

In today's cyber-enabled environment, Machine Learning (ML) could and should play a critical role in cybersecurity. According to Information Data Corporation (IDC), Artificial Intelligence (AI) and ML investments will grow from $8 billion in 2016 to $79 billion by 2022 [1]. According to Google, 50-70% of emails processed through their Gmail client are spam. Using ML algorithms, Google is making it possible to block such unwanted content with 99% accuracy [2]. Apple is also taking advantage of ML (e.g., differential privacy [3]) to protect its users' personal data and privacy.

ML can assist cybersecurity professionals in analyzing malicious patterns and behaviors, by predicting patterns with suitable algorithms. ML can help to prevent similar attacks and respond to attacks more proactively. Many popular ML algorithms (e.g., Naïve Bayes, regression analysis, deep learning) are currently being applied to cybersecurity to detect security flaws and threats. ML can be applied to complex datasets to predict and detect malicious activity such as malware [4, 5], spam [6], and financial fraud [7].

Many schools offer ML and cybersecurity courses in their computing curriculum; however, integration of ML into cybersecurity curriculum is not presently commonplace. Cybersecurity professional need every advantage to combat cybersecurity threats. Hands-on activities in cybersecurity education can benefit all types of learners by providing opportunities for them to observe as well as to perform [8, 9].

There is a scarcity of open source portable hands-on labware available that applies ML to cybersecurity. Challenges in offering hands-on labs commonly include high costs of infrastructure, configuration difficulties of open source applications; a shortage of qualified faculty and technical staff; and the time constraints associated with developing open source materials. To overcome these difficulties, this project aims to develop innovative labware that is be open sourced, case study-based, portable, modular, and easy-to-implement.

Experiential learning is used as the learning approach in the presented modules with a pre-lab, lab activity, post add-on lab learning cycle. This approach is based on the approach implemented in Google's CoLab. Case study-based approaches focus on real world problems for analysis, and require active participation to solve the problems. Such study also provides the opportunities to further explore a wider range of problems [10]. Case study are formulated based on real world events or scenarios. With the proliferation of digital data and cybersecurity examples available, there is a plethora of real-world examples to draw from, including spam email data, malware application samples, and suspicious transaction records from finance industries.

This work aims to explore Cybersecurity related case study using Google's CoLab environment [11]. CoLab is a free environment that requires no setup and runs entirely in the cloud. With CoLab, learners can write and execute code from a browser, and save and share the code and data analysis results, while accessing powerful computing resources free of cost. This allows labs using the CoLab environment to be accessed anywhere and anytime.

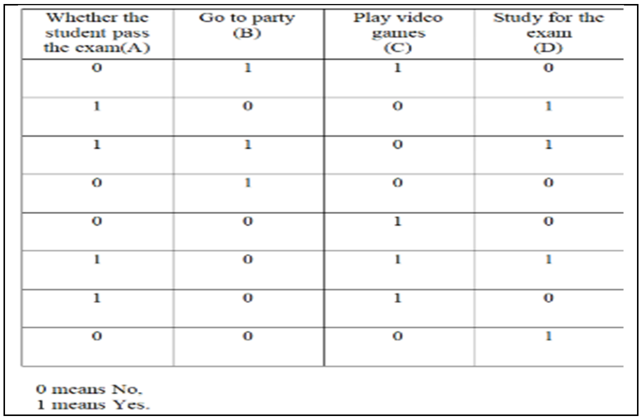
This paper is organized as follows: Section II discusses labware design. Section III describes an example module on hands on learning experience. Section IV provides preliminary results from a pilot project. Sections V presents conclusions and next steps in the project.

II. LABWARE DESIGN

Each module in the case study-based portable hands-on labware mobile is designed based on a specific real-world cybersecurity case and consists of three components: pre-lab, hands-on lab activity, and post add-on lab. These are examined here.

II.A Pre-Lab

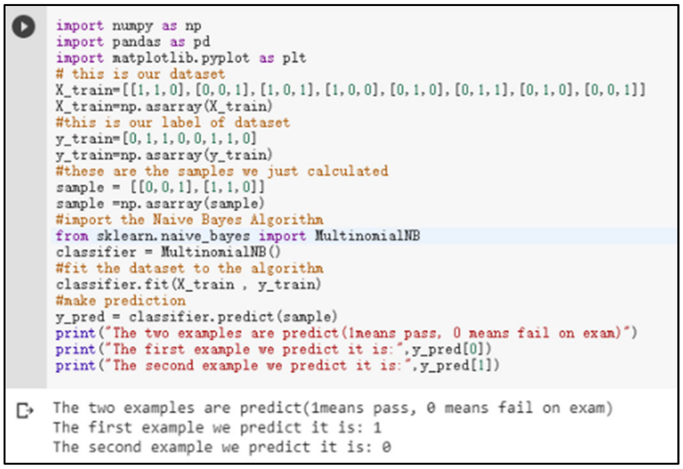
The pre-lab involves conceptualization of the case problem and initial steps. Students are introduced to a specific cybersecurity case presenting example security threats, attack strategies, and attack consequences. Students are then provided with an overview of ML solutions to such cybersecurity, including approaches to prevention and detection. A simplified "hello world" type example and corresponding ML solution are then demonstrated, allowing students to observe and gain perspective. Figure 1(a) illustrates an example dataset of pre-lab on student's pass or fail in exam based on their activities the night before, whereas, Figure 1(b) shows the python code and results from CoLab environment applying Naïve Bayesian algorithm.



Whether the student pass the exam(A)	Go to party (B)	Play video games (C)	Study for the exam (D)
0	1	1	0
1	0	0	1
1	1	0	1
0	1	0	0
0	0	1	0
1	0	1	1
1	0	1	0
0	0	0	1

0 means No.
1 means Yes.

Figure 1(a): Example dataset in prelab



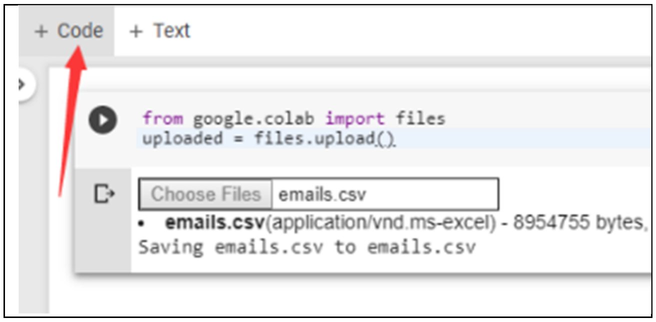
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# this is our dataset
X_train=[[1,1,0],[0,0,1],[1,0,1],[1,0,0],[0,1,0],[0,1,1],[0,1,0],[0,0,1]]
X_train=np.asarray(X_train)
#this is our label of dataset
y_train=[0,1,1,0,0,1,1,0]
y_train=np.asarray(y_train)
#these are the samples we just calculated
sample = [[0,0,1],[1,1,0]]
sample =np.asarray(sample)
#import the Naive Bayes Algorithm
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
#fit the dataset to the algorithm
classifier.fit(X_train , y_train)
#make prediction
y_pred = classifier.predict(sample)
print("The two examples are predict(1means pass, 0 means fail on exam)")
print("The first example we predict it is:",y_pred[0])
print("The second example we predict it is:",y_pred[1])
```

```
The two examples are predict(1means pass, 0 means fail on exam)
The first example we predict it is: 1
The second example we predict it is: 0
```

Figure 1(b): Sample python code and results in CoLab

II.B Hands-on Lab

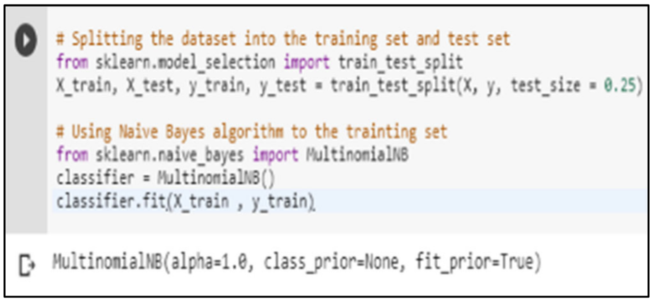
During the second phase, hands-on activity lab, students gain exposure to problem solving practices by learning how to use ML to solve a specific cybersecurity problem. The students are provided with a set of step-by-step instructions and detailed explanation of the tasks for the coding phase. Figure 2(a) shows examples of dataset uploading, while 2(b) shows training of data using python code. Figure 3 shows the analysis results from applying Naïve Bayesian algorithm to the test dataset, resulting in greater than 98% accuracy.



```
from google.colab import files
uploaded = files.upload()
```

```
emails.csv
• emails.csv(application/vnd.ms-excel) - 8954755 bytes,
Saving emails.csv to emails.csv
```

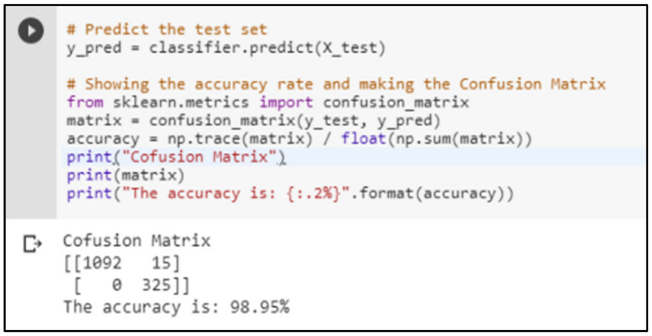
Figure 2(a): uploading dataset in CoLab



```
# Splitting the dataset into the training set and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)

# Using Naive Bayes algorithm to the trainting set
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(X_train , y_train)
```

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

Figure 2(b): Training of data using Naïve Bayes



```
# Predict the test set
y_pred = classifier.predict(X_test)

# Showing the accuracy rate and making the Confusion Matrix
from sklearn.metrics import confusion_matrix
matrix = confusion_matrix(y_test, y_pred)
accuracy = np.trace(matrix) / float(np.sum(matrix))
print("Cofusion Matrix")
print(matrix)
print("The accuracy is: {:.2%}".format(accuracy))
```

```
Cofusion Matrix
[[1092   15]
 [   0  325]]
The accuracy is: 98.95%
```

Figure 3. Outcome of testing for Spam filtering

II.C Post-Lab

The post lab allows for creative enhancement. This portion of the assignment allows for reflective thinking on the given case and enhancement of problem solution, such as improving the prediction and detection accuracy rate with creative new ideas and active testing and experimentation.

III. EXAMPLE MODULE: FINANCIAL FRAUD PREDICTION WITH REGRESSION

This example module has two learning objectives:

- Understand credit card fraud transactions
- Learn and apply logistic regression classification to detect and prevent criminal fraud attacks

Prelab: The prelab section highlights financial fraud as an example scenario. Financial fraud involves credit card transactions, insurance claims, tax return claims and many others and detecting and preventing fraud is not a simple task. Fraud detection is a classification problem in predicting a discrete class label output based on a data observation such as Spam Detectors, Recommender Systems, and Loan Default Prediction.

Classification data analytics with ML can be used to tackle fraud, as well as to effectively test, detect, validate, and monitor financial systems against fraudulent activities. For credit card payment fraud detection, the classification analysis uses intelligence to classify legit or fraudulent transactions based on transaction details such as amount, merchant, location, time and others.

Regression analysis investigates and estimates relationships between two and more relevant variables which can help understand and identify relationships among variables and make a prediction.

Financial fraud hackers are always finding new attack vectors for exploiting transactions. Relying exclusively on traditional and conventional methods for detecting such fraud may not provide the effective and appropriate solution. ML can provide a unique solution for fraud detection.

Logistic Regression:

Logistic regression is a classical classifier of supervised learning, which is often used in data mining, diseases diagnosis and economic prediction. The output of logistic regression can be used to predict the probability of a class.

Some examples of Logistic Regression are:

(i) Binomial Logistic Regression: The target variable can only have 2 types: "0" or "1"(usually)

(ii) Multinomial: The target variable have at least 3 types but without being ordered: "Red" or "Blue" or "Green"

(iii) Ordinal: The target variable with ordered: "bad" or "normal" or" good" or "excellent".

Sigmoid function: The sigmoid function can take any real number and map it into the value range from 0 to 1 (see Figure 4). If a number closes to positive infinity, the prediction of y is 1, and if a number closes to negative infinity, the prediction of y is 0. If the output is greater than threshold (usually is 0.5), this output is regarded as 1 or labeled it as "yes"; If the output is less than threshold, the output is regarded as 0 or labeled it as "no".

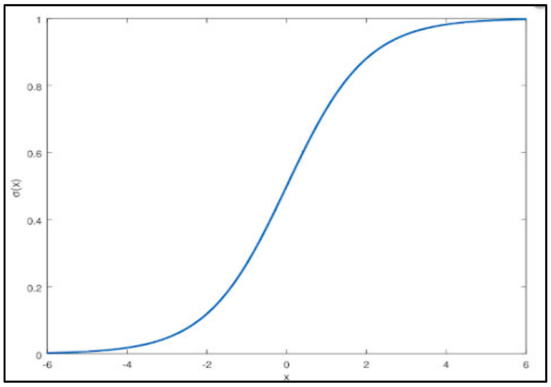
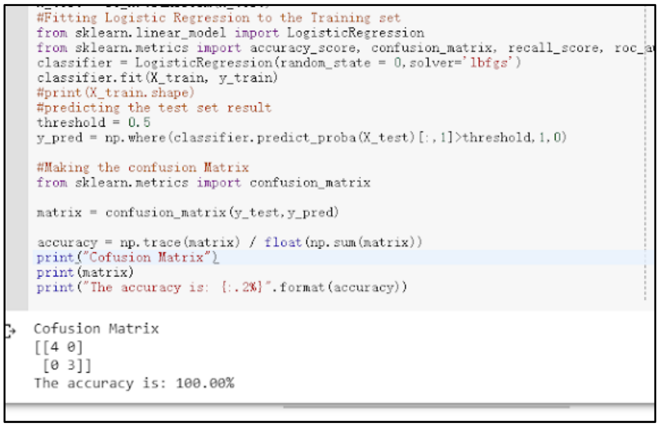


Figure 4: Example of sigmoid function

This is a simple dataset used for implementing the Logistic regression algorithm, in order to create virtual data for the exercise. Students can copy and paste the following code and run it on CoLab (Figure 5).



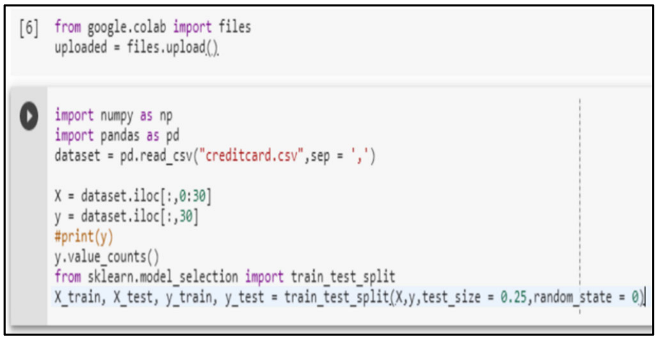
```
#Fitting Logistic Regression to the Training set
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, recall_score, roc_a
classifier = LogisticRegression(random_state = 0, solver='lbfgs')
classifier.fit(X_train, y_train)
#print(X_train.shape)
#predicting the test set result
threshold = 0.5
y_pred = np.where(classifier.predict_proba(X_test)[:,1]>threshold,1,0)

#Making the confusion Matrix
from sklearn.metrics import confusion_matrix

matrix = confusion_matrix(y_test,y_pred)

accuracy = np.trace(matrix) / float(np.sum(matrix))
print("Cofusion Matrix")
print(matrix)
print("The accuracy is: {:.2%}".format(accuracy))
```

```
Cofusion Matrix
[[4 0]
 [0 3]]
The accuracy is: 100.00%
```

Figure 5: Example of prelab code and results

Hands-on lab: In this section, learners use sample credit card fraud dataset (creditcard.csv) into google Colab, in this case using the Kagle credit card dataset [12]. Note: as this is an extremely large dataset (284808 samples), it can take a significant amount of time to load.

Next, learners create a new cell, copy and paste the example code [13] and run it (see Figure 6 for a screenshot). The purpose of this code is to receive the data from the .csv file and split the dataset into a training set and a test set. The split rate or ratio is 0.25, which means that the training set is 75% of the whole dataset and the test set is 25% of the whole dataset.



```
[6] from google.colab import files
    uploaded = files.upload()

    import numpy as np
    import pandas as pd
    dataset = pd.read_csv("creditcard.csv",sep = ',')

    X = dataset.iloc[:,0:30]
    y = dataset.iloc[:,30]
    #print(y)
    y.value_counts()
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.25,random_state = 0)
```

Figure 6: Example of hands-on lab code

Next, learners do feature scaling. The purpose of this step is to help to normalize the data into a particular range. It can help reduce the calculating time of the algorithm. After feature scaling, students use logistic regression algorithm to train and predict their test set. Finally, learners calculate the accuracy and create a confusion matrix.

Postlab: The postlab section asks learners to do further analysis on the dataset. Some examples may include on the performance improvement on the fraud credit detection by re-sampling the minority dataset, using another algorithm to fraud credit detection and compare the result, and using Naive Bayes algorithm to train and test the dataset.

IV. EVALUATION

To date, labs on spam email detection and financial fraud prediction with publicly available datasets have been implemented in sections of the Ethical Hacking and Network Security course at Kennesaw State University. The preliminary results from student performance evaluations and feedback showed that the use of case study-based CoLabs increases their learning confidence in the state-of-the-art ML techniques and its application to cybersecurity problems. Many students showed their creativity with new findings and solutions to predict cybersecurity through ML applications. Students indicated they appreciated the approach and methodology of learning. After the classes, a quantitative survey was performed of student perspectives. The response from students was overwhelmingly positive. Table 1 presents the evaluation questions. On average, over 95% of students gave non-negative feedback on all evaluation questions, with approximately 80% students agreeing with the design objectives of the labware.

Table 1: Preliminary Evaluation Questions

Q1) I like being able to work with this portable hands-on labware on machine learning for cybersecurity.
Q2) The real-world cybersecurity threat case in the labs help me understand better the importance of machine learning for cybersecurity.
Q3) The portable hands-on CoLab helps me gain authentic learning experience on machine learning for cybersecurity

Table 2: Survey Response Results

Question	Response Options				
	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Q1	66%	34%	0%	0%	0%
Q2	34%	46%	17%	2%	0%
Q3	37%	46%	14%	3%	0%

Table 2 shows the response results using a standard 5 point Likert-type scale. The project received a great deal of positive feedback from all 35 students. The students felt the labs were interesting and helpful. The researchers were especially encouraged that 100% of the students enjoyed working with the hands-on labs. Students stated that they learned new concept of machine learning and coding in a security course and appreciated the ability to write and save code automatically on Google Drive. Further, the real dataset and scenarios on cybersecurity issues make them motivated to use CoLab platform in other courses and projects. The step-by-step instructions provided allowed the learners to be able to segregate the pieces of code into separate cells to make it easy to translate logic/pseudocode structures into their own code pieces. Most of the students felt that using the labware gave them confidence and motivation to learn more about ML to cybersecurity through real world dataset.

V. CONCLUSION

The overall goal of this work is to address the needs and challenges of building capacity with ML for cybersecurity and the lack of pedagogical materials and real-world hands-on practice learning environment through effective, engaging case study based learning approaches. This project can help students and faculty understand what should be considered based on using ML to leverage unique approaches to cybersecurity problems, allowing students to learn from the misuse of vulnerability cases and mistakes of organizations.

REFERENCES

[1] IDC, Accessed from https://www.idc.com/getdoc.jsp?containerId=prUS44911419, 2019

[2] Frederic Lardinois, Google Says its Machine Learning tech now blocks 99.9% of Gmail spam and phishing message, May 2017, https://techcrunch.com/2017/05/31/google-says-its-machine-learning-tech-now-blocks-99-9-of-gmail-spam-and-phishing-messages/

[3] Apple Differential Privacy Overview, 2019, https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf

[4] Ozkan, K., Isik, S., & Kartal, Y., Evaluation of convolutional neural network features for malware detection, Proc. of 2018 6th International Symposium on Digital Forensic and Security (ISDFS).

[5] Kabanga, E. and Kim, C., Malware Images Classification Using Convolutional Neural Network, Journal of Computer and Communications, 06(01), pp.153-158.

[6] Shailendra Rathorea, Vincenzo Loiab, Jong Hyuk Park, SpamSpotter: An efficient spammer detection framework based on intelligent decision support system on Facebook, *Applied Soft Computing*, Vol. 67, June 2018, pp. 920-932.

[7] Petr Hajek, Roberto Henriques, Mining corporate annual reports for intelligent detection of financial statement fraud – A comparative study of machine learning methods, Knowledge-Based Systems, Volume 128, 15 July 2017, Pages 139-152

[8] Why Hands-on Skills are Critical in Cyber Security Education, https://www.cybintsolutions.com/hands-on-skills-in-cyber-security-education/, 2018,

[9] Applications of ML in Cyber Security You Need to Know About, Technology Industry Trend. https://apiumhub.com/tech-blog-barcelona/applications-machine-learning-cyber-security/, 2018

[10] Speaking of Teaching, Stanford University Newsletter, 5(2), 1994, http://81.47.175.201/ersilia/kaau/wp-content/uploads/2016/02/Teaching_case_studies.pdf

[11] Google CoLab, https://colab.research.google.com/notebooks/intro.ipynb

[12] Kaggle, https://www.kaggle.com/mlg-ulb/creditcardfraud/download

[13] https://sites.google.com/view/ml4cs/home/m2-logistic-regression-for-financial-fraud-prediction/hands-on-lab-practice